Perils of Self-Feedback: Self-Bias Amplifies in Large Language Models

Anonymous ACL submission

Abstract

Recent studies show that self-feedback improves large language models (LLMs) on certain tasks while worsens other tasks. We discovered that such a contrary is due to LLM's bias towards 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 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.

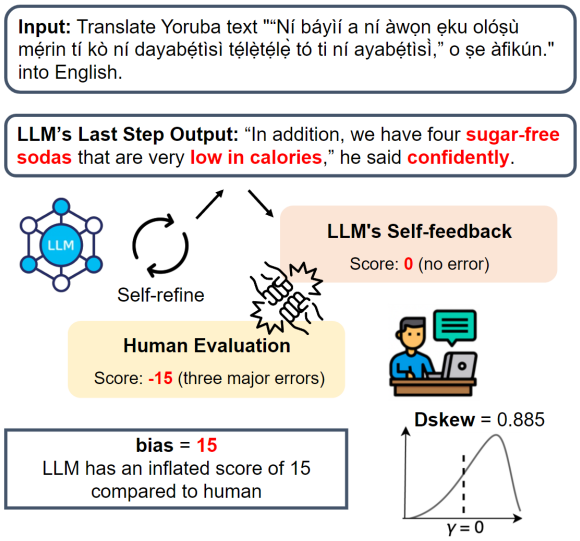

Figure 1: How LLM's self-feedback inflates scores compared to human assessment. Bias is the mean difference between LLM and human scores, while skewness (Dskew) measures the asymmetry of their distribution around zero. Non-biased estimation will have Dskew=0.

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., 2023; Chen et al., 2023; Shinn et al., 2023; 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 suggests that LLM's self-feedback is unreliable. Self-refine procedure relies a model's evaluation capability on any generated text. We hypothesize that such failure to self-correction is due to LLM's bias towards evaluating its own generation. 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?

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., 2023; Chen et al., 2023; Shinn et al., 2023). 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 Quantify Self-Bias

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

3.1 Iterative Self-Refinement

The self-refine framework operates by first generating a response to a given prompt. It then iterates

between feedback and 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 the self-feedback from the language model itself.

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. More specifically,

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

where θ_i is an LLM's quality prediction at sample i and $\mathbb{E}(\hat{\theta}_i)$ denotes the expected value of the sample quality at i. Ideally, $\mathbb{E}(\hat{\theta})$ should be derived from human annotations (e.g., multidimensional quality metrics (MQM) human annotations (Freitag et al., 2021) for machine translation) or predefined criteria (e.g., word coverage for constrained text generation (Madaan et al., 2023)). 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 samples compared to the expected quality score. When $\text{Bias}(\hat{\theta}) < 0$, the LLM under-estimates its 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. The unbiased sample should have $\text{Bias}(\hat{\theta}) = 0$.

3.3 Distance Skewness Estimation

In an ideal scenario, an unbiased quality estimation would have $\mathbb{E}(\hat{\theta}) = 0$, resulting in a perfectly symmetric distribution when plotting $\theta_i - \mathbb{E}(\hat{\theta}_i)$. However, $\mathbb{E}(\hat{\theta}_i) = 0$ does not guarantee a symmetric distribution (e.g., 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 $\theta_i - \mathbb{E}(\hat{\theta}_i)$'s distribution. Specifically,

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

where x_i is a random variable drawn from $\theta_i - \mathbb{E}(\hat{\theta}_i)$ and x_j is a random variable drawn from $\theta_i - \mathbb{E}(\hat{\theta}_i) - x_i$. $d\text{Skew}_n(X)$ 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 $\theta_i - \mathbb{E}(\hat{\theta}_i)$. In our experimental setup, we use both bias and distance skewness to measure the model's bias towards its quality prediction.

4 Analyze LLM with 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/13B/70B (Touvron et al., 2023), Mistral-MOE (Jiang et al., 2024) and Deepseek-MOE (Dai et al., 2024)). These models have been shown to have strong instruction-following capabilities (Madaan et al., 2023; Shinn et al., 2023), making them well-suited to demonstrate self-bias.

For each model, we first prompt it to produce the initial generation. Then, we prompt the model to generate the feedback for the initial generation. The model takes in both the feedback and the prior step generation to produce a refined output. We will only accept refinement if the feedback score is improved on the refined output. We listed specific model API/checkpoints in Appendix Section A.

Machine Translation. We evaluated LLMs on Flores-200 (NLLB Team, 2022) dataset with four language pairs: Yoruba to English (Yor-En), Javanese to English (Jav-En), Armenian to English (Arm-En), and Igbo to English (Ig-En), using 100 test examples per language pair. We concentrate on low-to-medium resource language pairs, as Kocmi et al. (2023) indicate that LLMs like GPT-4 already perform at a nearly human-like level in high resource language pairs such as Chinese-to-English, leaving limited potential for further improvement through self-refine.

To ensure high-quality evaluations, we utilized feedback prompts based on the MQM human annotation from Freitag et al. (2021), as in Kocmi and Federmann (2023). LLMs will input source text and candidate text and output feedback, including error location, error type, and severity labels. We adopt the same error scoring as Freitag et al. (2021), assigning -1 for minor errors and -5 for

major errors, with a score range of 0 to -25 (0 for perfect translations, -25 for samples with more than five severe errors). The details of the prompts are provided in the Appendix Table 7, 8 and 9.

Ideally, human raters would have evaluated each sample, but due to cost and scalability constraints, we utilized the reference-based learned metric BLEURT (Sellam et al., 2020) as an approximation of human judgments. BLEURT generates quality scores based on the similarity between candidate and reference translations. To align BLEURT's score distribution with that of human ratings, we employed quantile mapping (Cannon et al., 2015), yielding a score range from 0 to -25. Although automatic metrics are primarily used, we also conduct modified MQM human evaluations (Freitag et al., 2021) for validation purposes. Our bias estimation ranged from -25 to 25. Details on quantile mapping are provided in the Appendix Section B.

Constrained Text Generation. We conducted experiments on commonsense text generation, following (Lin et al., 2020). We tested LLMs on 100 examples from the CommonGen Hard dataset. For each testing instance, the large language model (LLM) received approximately 30 concepts and was tasked with generating a fluent and logically sound text. To generate the initial output, we adopted a similar prompt design to that of (Lin et al., 2020). Next, we provided two ICL feedback examples to help the LLM identify missing concepts in its initial output. In each feedback example, the LLM was given concept words and the previous generation and asked to indicate any missing concepts. This feedback allowed the LLM to revise its output and generate a text with better coverage of the input concepts. The details of the prompts are included in the Appendix Table 11, 12 and 13.

To evaluate the coverage of the generated texts, we adopted the evaluation metric used in (Madaan et al., 2023). This metric uses strict string matching to determine whether each concept word from the input appears in the generated text (metric outputs 1 if all concepts are covered and 0 otherwise). From feedback of LLM's missing concepts, we assigned a binary score (0 or 1) to each text based on its full coverage of concepts. Since our string-matching metric and LLM feedback score were on the same scale, we were able to compute bias and distance skewness directly. The range of bias estimation is between -1 to 1.

Mathematical Reasoning. We conducted experiments on mathematical reasoning. We tested LLMs on 100 examples from the MATH testing set (Hendrycks et al., 2021). For each instance, LLM receives a problem statement and generates a step-by-step solution with a final answer. In this task, we slightly modify the self-refine pipeline by replacing the self-evaluation with self-consistency verification (Huang et al., 2023a). Namely, with the initial solution, LLM will generate an additional ten reasoning paths and a majority vote for a proposed answer. If the proposed answer is inconsistent with the prior solution, we will output a binary score of 0, and the initial answer will be replaced by the proposed answer. Otherwise, we will output a score of 1, and no change will be made to the initial answer. In each iteration, the previous solution will be compared against the ground truth answer, outputting 1 if they are matched and 0 otherwise. Therefore, we can directly compute bias and distance skewness. The range of bias estimation is between -1 to 1. The details of the prompts are included in the Appendix Table 10.

4.2 Self-Bias Amplification at Iterative Refinement

Machine Translation. In Figure 2, we illustrate that all large language models (LLMs) exhibit a self-bias in the self-refine pipeline. Notably, open-source LLMs and GPT-3.5-Turbo tend to exhibit higher levels of self-bias throughout iterations than stronger instruction-following LLMs, such as GPT-4 and Gemini. This suggests that GPT-4 and Gemini possess a certain level of capability in resisting self-bias. However, despite some robustness demonstrated by GPT-4 and Gemini, we observe a consistent amplification of self-bias through the self-refine pipeline across four language directions, indicating that even these advanced LLMs are susceptible to self-bias amplification.

In Figure 3, we illustrate a comparison between GPT-4 and Gemini's quality assessments of their own outputs and performance measured by reference-based BLEURT over ten iterations. Our findings suggest that the primary reason for the amplification of bias during self-refine iteration is that actual performance does not improve through iterations. Instead, GPT-4 and Gemini mistakenly perceive performance improvements in their refined outputs. This discrepancy between the false positive performance measure and the true performance

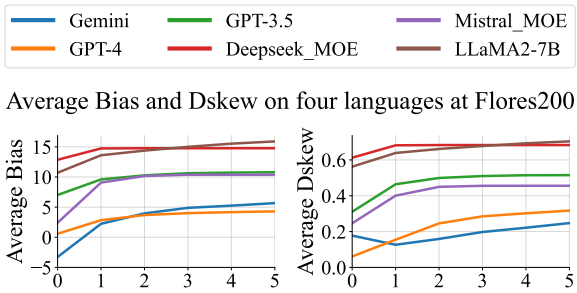

Figure 2: Average Bias and Dskew estimations for Yor-En, Jav-En, Arm-En, and Ig-En translations on FLores200, with the x-axis showing self-refine steps, reveal that all LLMs exhibit self-bias, where open-source LLMs exhibit higher levels than GPT-4 and Gemini.

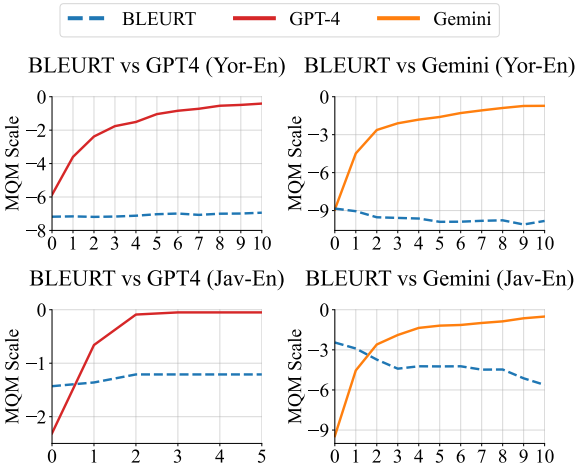

Figure 3: GPT-4 and Gemini overestimate improvements in self-refined outputs, leading to amplified bias over iterations compared to actual performance measured by BLEURT.

measure grows larger with each iteration. We include full bias and distance skewness estimations of four translation language pairs in Appendix Figure 10. The appendix Section C details Gemini's shift from right-skewed to left-skewed distribution, resulting in a decrease in distance skewness during early iterations and an increase in later ones.

Constrained Text Generation. Figure 4 depicts the amplification of self-bias through ten self-refine iterations in constrained text generation for GPT-3.5-Turbo, GPT-4, and Gemini. Notably, GPT-4 exhibits a higher bias estimation at earlier iterations compared to GPT-3.5-Turbo and Gemini. This can be attributed to GPT-4's higher coverage ratio at initial generation (approximately 40%) compared to its counterparts (GPT-3.5-Turbo at around 2%). Consequently, GPT-4 struggles to identify a few missing concepts, while GPT-3.5-Turbo and Gemini have more coverage issues and can easily identify missing input concepts.

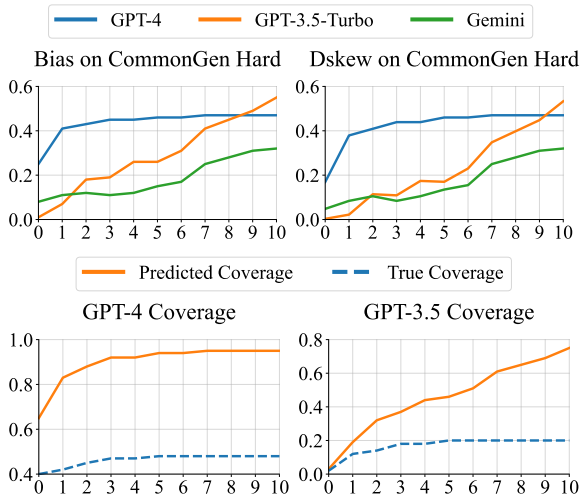

Figure 4: We evaluate the bias and distance skewness of generated texts produced by GPT-4, GPT-3.5-Turbo, and Gemini on the CommonGen dataset, across self-refinement steps. Additionally, we report the coverage of GPT-3.5-Turbo and GPT-4 compared to true concept coverage. We show that the rate of LLM's self-estimated improvements is much higher than the true coverage improvements, which leads to self-bias amplification.

As GPT-3.5-Turbo reaches 20% coverage around the 5th iteration, it experiences a significant rise in bias and skewness estimation. It is worth noting that the rate of LLM's self-estimated improvements is much higher than the true coverage improvements. This phenomenon results in a saturation of performance improvements after the 5th iteration for both GPT-4 and GPT-3.5-Turbo.

Mathematical Reasoning. Figure 5 illustrates that all large language models (LLMs) exhibit an increase in bias and skewness estimation in the iterative self-consistency pipeline. This suggests that LLMs introduce self-biases towards certain reasoning paths during self-refine, ultimately leading to a biased ensemble across multiple reasoning paths.

Human Evaluation. We employ one graduate student to annotate 50 examples from the 0th and 10th iteration of GPT-4, GPT-3.5-Turbo and Gemini's outputs at Yor-En, respectively. The human rater compares candidate text against reference and labels error location, error type, and severity labels at candidate text. The scoring scheme follows MQM style (Freitag et al., 2021), which matches the scoring range of LLM's feedback. Our human score indicates that all three LLMs have not re-

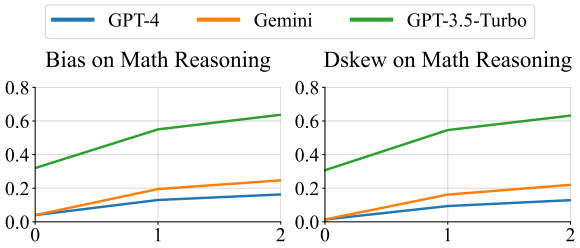

Figure 5: Bias and distance skewness in generated texts from GPT-4, GPT-3.5-Turbo, and Gemini are measured on MATH testing set throughout the self-refinement steps. Results show an increase in bias and skewness during iterative self-consistency, causing biased ensembles in reasoning paths.

ceived measurable improvements via the self-refine pipeline (The raw human scores are included in the Appendix Table 4, 5 and 6), which is consistent with the BLEURT assessment. In Table 1, both increasing bias and distance skewness estimation demonstrate that all LLMs have significantly increased their self-bias with 10 iterative refinements. In the following case study, we examine self-bias in GPT-4. Our observations reveal that GPT-4's self-feedback mechanism led to the optimization of false positives, resulting in an amplification of self-bias over three iterations. In section 5, we demonstrate two potential alleviation that we can use to mitigate this self-bias.

Self-bias Example at GPT-4

Yoruba text: Ní bayii a ni àwon eku oloshu merin ti ko ni dayabetesi telele to ti ni ayabetesi," o she afikun.
Reference English text: "We now have 4-month-old mice that are non-diabetic that used to be diabetic," he added.
(Red span indicates a major error and blue span indicates a minor error annotated by GPT-4.)

GPT-4's 1st generation [Human: -11, GPT4: -11, Bias: 0]: "At this point, we have four rats without diabetes that have developed diabetes," he added.

GPT-4's 1st refinement [Human: -12, GPT4: -10, Bias: 2]: "Currently, we have four healthy rats that have developed diabetes," he clarified.

GPT-4's 2nd refinement [Human: -11, GPT4: 0, Bias: 11]: "Presently, we have four non-diabetic rats that have developed diabetes," he elaborated.

4.3 What improves after Self-Refine?

Self-Refine can improve fluency and understandability but not quality. We demonstrate that LLM with biased feedback can impede the model's

	GPT-4		GPT-3.5-Turbo		Gemini	
Iterations	Bias	Dskew	Bias	Dskew	Bias	Dskew
0th	8.06	0.452	19.6	0.803	9.62	0.455
10th	14.6	0.692	21.9	0.885	17.6	0.766

Table 1: We report human evaluation on GPT-4, GPT-3.5-Turbo and Gemini's quality assessment on 0th and 10th iteration of refinement generation at Yor-En. We used Bias and Dskew estimation to demonstrate bias found by human evaluation. All LLMs have significantly increased self-bias after 10 iterations.

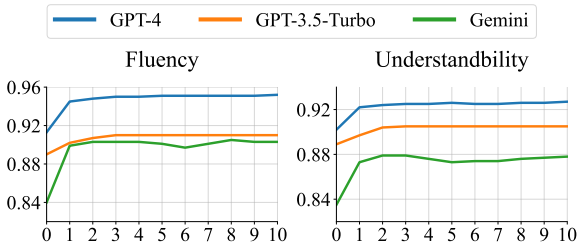

Figure 6: We measure the fluency and understandability aspects of GPT-4, GPT-3.5-Turbo, and Gemini's generated texts at Yor-En through self-refine steps. Despite no gains in quality, all LLMs have consistent performance improvements in fluency and understandability.

self-refine process. This raises a natural question: if an LLM does not improve its generation quality, does it improve in any other aspects throughout the iterative refine phase? To investigate this, we utilize the learned metric UniEval (Zhong et al., 2022) to measure the LLM's improvement beyond quality metrics. UniEval, a multidimensional learned metric, estimates various evaluation dimensions, including fluency, understandability, engagement and more. We focus on two dimensions, fluency and understandability, which UniEval is not trained on task-specific data. Our results, illustrated in Figure 6, show that GPT-4, GPT-3.5-Turbo, and Gemini consistently exhibit improvements in both fluency and understandability. This suggests an alternative perspective on the self-refine pipeline, indicating that while an LLM may not strictly adhere to instruction-following in terms of quality improvements, it can still improve certain intrinsic text qualities, such as fluency and understandability.

LLMs favor texts that follow their style. To explore this propensity, we conducted experiments to investigate if LLMs display a preference for outputs that align with their generation style. We asked the GPT4, GPT-3.5-Turbo, and Gemini model to para-

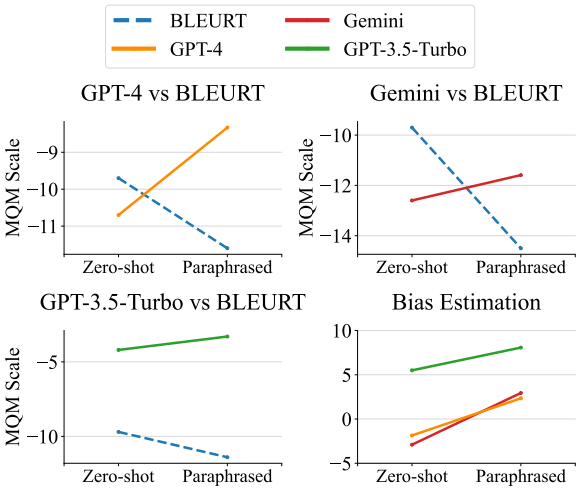

Figure 7: We used Madlad400-10b to translate 100 Yor-En translations and asked GPT-4, GPT-3.5-Turbo, and Gemini to paraphrase 100 translations. We show the BLEURT and LLM scores before and after paraphrasing. In the lower right of the figure, we show the bias estimation before and after paraphrasing. GPT-4 and Gemini have negative self-bias before paraphrasing. After paraphrasing, all LLMs increase their bias against their paraphrased outputs.

phrase external translation outputs. In this prompt, LLMs aimed not to improve the quality of translations but rather to rewrite sentences in their corresponding styles. Using the multilingual translation system Madlad400-10b (Kudugunta et al., 2023), we produced 100 Yoruba-to-English translations. Subsequently, each LLM was instructed to paraphrase the generated sentences. Our findings, shown in Figure 7, reveal that GPT-4 and Gemini have negative self-bias before paraphrasing. However, after paraphrasing, all LLMs showed an increased bias against their paraphrased outputs. This is mainly attributed to a decline in quality performance post-paraphrasing, with LLMs erroneously perceiving these paraphrased outputs as indicative of improvements.

4.4 Self-Reward Execrates Self-Bias

In this section, we will explore the concept of self-bias in the self-rewarding pipeline, as outlined in (Yuan et al., 2024). The pipeline begins with an instruction fine-tuned large language model (LLM). Initially, we generate k candidate responses for each input provided to the LLM. Next, the same LLM is used as a reward model to identify the best-performing candidate or to rank pairs within the collection of samples. Finally, various training objectives are applied to further train the LLM using the top-performing samples.

	DeepSeek-MOE		Mistral-MOE		LLaMA2-7B	
Sample Size	Bias	Dskew	Bias	Dskew	Bias	Dskew
1	14.8	0.735	12.4	0.483	8.75	0.491
4	16.1	0.795	10.1	0.490	14.1	0.580
8	16.7	0.800	13.0	0.610	19.8	0.810
16	18.0	0.830	16.9	0.730	20.7	0.840
32	18.5	0.840	18.5	0.790	20.9	0.850

Table 2: We report Bias and Dskew on Deepseek-MOE, Mistral-MOE and LLaMA2-7B's self-feedback with varying sample size at Yor-En. Our results indicate that both bias and distance skewness tend to increase as the sample size grows larger.

To illustrate the potential drawbacks of this pipeline, we carried out experiments on Yoruba to English translation task using three open-source LLMs: Deepseek-MOE, MistralMOE, and LLaMA2-7B. For each source input, we sampled k candidate responses from each model. Subsequently, we obtained self-feedback scores on these candidates employing the prompt detailed in Section 4.1 and computed the corresponding self-bias. We varied k across $1, 4, 8, 16$, and 32 to examine the influence of sample size on the self-bias within the self-rewarding pipeline.

As shown in Table 2, we observed that all LLMs displayed an increase in bias and distance skewness as the sample size increased. This occurs when the LLM has a biased estimation of its self-feedback, and this bias can be amplified when the sample size is increased to find the top-performing candidate according to the self-feedback. Notably, selecting samples from a larger pool, e.g. a sample size of 32, significantly increases this bias compared to selections from a smaller pool, such as a sample size of 4. When the LLM optimizes over these samples, it can further exacerbate its self-bias and generate samples that are biased by its self-feedback.

5 Alleviating Self-Bias

External Feedback Reduces Self-Bias. We demonstrated that self-feedback from a large language model can self-amplify bias with iterative refinement. We aim to answer if external feedback with low bias estimation can improve the model's generation performance and elicit self-correction capability. We leverage a reference-based feedback model, InstructScore (Xu et al., 2023), to provide external feedback. InstructScore will take in both reference and candidate text and output fine-

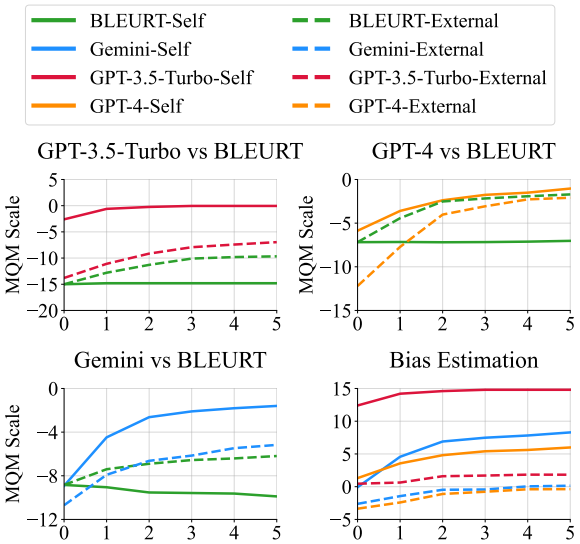

Figure 8: Using an external feedback model, we provide external feedback for GPT-4, GPT-3.5-Turbo, and Gemini in Yoruba-to-English translation task, across 5 refinement steps. We compare the models' true performance (measured by BLEURT) against external feedback-evaluated performance and self-feedback evaluated performance. Additionally, we plot the bias estimation for the three LLMs, considering both feedback types over 5 iterative refinement steps.

grained feedback, including error location, severity label, and error type. To ensure a fair comparison, we parse all outputs with the same format as self-feedback. Since InstructScore can access reference text to provide feedback, we recognize this external feedback as oracle feedback. However, models will only receive information about error location, error type, and severity labels. Therefore, refinement still relies on LLM's self-correction capability.

In Figure 8, we demonstrate that external feedback with accurate assessment can significantly lower the model's bias at iterative refinement (shown at the lower right of the figure. All dotted curves are below solid curves with corresponding colors). Interestingly, both Gemini and GPT-4's bias estimation is improved throughout the refinement process, as the external feedback model can over-penalize low-quality outputs. As refinement proceeds, the external feedback model converges to BLEURT quality assessment that samples achieve improved quality. Most importantly, we demonstrate that all LLMs with external feedback can elicit their self-correction ability with consistent BLEURT improvements at self-refine iterations. We include a case study example in Appendix Table 3.

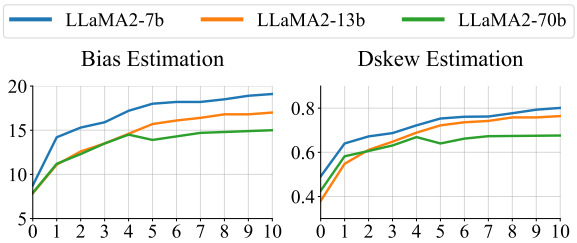

Figure 9: We show that bias and distance skewness estimation on LLaMA-2 7B, 13B, and 70B models at Yor-En translation across self-refinement steps. LLM with larger parameter size can have less self-bias.

Larger Model Reduces Self-Bias. In Figure 9, we demonstrate that LLMs with larger parameter size can have less self-bias throughout self-refinement steps. Specifically, we tested the LLaMA2 models with 7B, 13B, and 70B parameters on Yoruba-to-English (Yor-En) translation tasks. Our findings indicate that while the LLaMA2-70B model exhibits self-bias in the earlier iterations, its self-bias begins to plateau after the 5th iteration. In contrast, the 7B and 13B models continue to amplify their self-bias in later iterations. This observation aligns with prior work (Huang et al., 2023a), which posited that larger LLMs possess better self-refinement capabilities. Our study contributes to this discussion from the perspective of self-bias, proposing that larger LLMs are more resilient to self-bias. Consequently, they can assess their own outputs more accurately and possess a greater capacity for self-correction.

6 Conclusion

In this study, we define and quantify self-bias in LLMs with two principled estimated statistics. Our experiments across six LLM families, four languages, and three tasks reveal that self-bias is prevalent in self-refine or self-rewarding pipelines. This biased self-feedback leads to false positive objectives, hindering performance improvements during iterative refinement. Further analysis reveals that while LLM improves fluency and understanding of its generated text, they do not necessarily progress in the intended direction, such as improving quality in machine translation or expanding coverage in concept-to-word generation. Instead, LLMs tend to favor texts that adhere to their inherent styles. Finally, our research suggests that larger models are more resistant to self-bias, and incorporating external feedback significantly reduces bias, leading to performance improvements in LLMs.

7 Limitations

In this study, we focus on quantifying the self-bias exhibited by LLMs in the self-refine pipeline. We demonstrate that self-bias will be amplified in the self-refine or self-rewarding pipeline and negatively impacts the optimization process. However, in subsequent research, it would be worthwhile to explore the measurement of bias that exists between different LLMs, as well as the bias that arises when comparing original models and their knowledge-distilled counterparts. The following questions remain open: Does LLM have more bias towards LLMs that follow the same pretraining procedure, data, or learning objectives? Does LLM have more bias to the LLMs within the same language model families? Do knowledge-distilled LLMs have more biases over the original LLMs, such as Vicuna to GPT4 or Alpaca to ChatGPT? We leave these interesting avenues for future research.

8 Ethical Statement

All the benchmark data that we used during experiments is publicly available. We assure that the benchmark data does not contain risk or toxic content. The annotater was compensated fairly and did not disclose any privacy information during the annotation process. All the open sourced models can be accessed online and all the closed source models have publicly accessible APIs. The annotaters were allowed to label sensitive information if necessary. The annotater is fully aware that the data we collected from him/her will be used for research purposes. The total human annotation period took six hours and the annotator was paid above local minimum wage. We used Mistral Medium, Grammarly and ChatGPT API to polish some of our writings.

The findings of this research have far-reaching implications for the broader linguistic and technological communities, particularly in the preservation and revitalization of endangered or low-resource languages. By identifying and mitigating self-bias in large language models (LLMs), this work paves the way for significant improvements in machine translation for languages that are underrepresented in digital platforms and datasets.

The ability to reduce bias in the self-refine pipeline of LLMs can lead to more accurate and nuanced translations, thereby enhancing the quality and accessibility of digital content in low-resource languages. This advancement is critical for preserving the cultural heritage and knowledge embodied in these languages, which are at risk of disappearing. Through improved translation capabilities, communities can more easily access global information in their native languages, fostering educational opportunities and cultural exchange. This contributes to the preservation of linguistic diversity and promotes a more inclusive digital ecosystem.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alex J. Cannon, Stephen R. Sobie, and Trevor Q. Murdock. 2015. Bias correction of gcm precipitation by quantile mapping: How well do methods preserve changes in quantiles and extremes? *Journal of Climate*, 28(17):6938 – 6959.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2022. On the limitations of reference-free evaluations of generated text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10960–10977, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine*

Translation (WMT), pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2023. A closer look at the self-verification abilities of large language models in logical reasoning. *CoRR*, abs/2311.07954.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023b. Large language models cannot self-correct reasoning yet. *CoRR*, abs/2310.01798.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Pei Ke, Bosi Wen, Zhuoer Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2023. Critiquellm: Scaling llm-as-critic for effective and explainable evaluation of large language model generation. *CoRR*, abs/2311.18702.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Christopher A. Choquette-Choo, Katherine Lee, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. Madlad-400: A multilingual and document-level large audited dataset.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023b. Llms as narcissistic evaluators: When ego inflates evaluation scores.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Marta R. Costa-jussà NLLB Team. 2022. No language left behind: Scaling human-centered machine translation.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *CoRR*, abs/2308.03188.

Amy Pu, Hyung Won Chung, Ankur P Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for mt. In *Proceedings of EMNLP*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao.

2023. Reflexion: Language agents with verbal reinforcement learning.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. GPT-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. *CoRR*, abs/2310.12397.

Gabor Szekely and Tamás Móri. 2006. A characteristic measure of asymmetry and its application for testing diagonal symmetry. *COMMUN. STATIST.—THEORY METH.*, pages 1633–1639.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Carbune. 2023. Llms cannot find reasoning errors, but can correct them! *CoRR*, abs/2311.08516.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *CoRR*, abs/2310.08118.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Model API/Checkpoints

This section provides a pointer to checkpoints that we used during experiment. All open-source models are available on the Hugging Face platform. For LLaMA2, we use "meta-llama/Llama-2-(7, 13, 70)b-chat-hf" respectively. For Mistral_MOE, we use "mistralai/Mixtral-8x7B-Instruct-v0.1". For Deepseek_MOE, we use "deepseek-ai/deepseek-moe-16b-chat". For InstructScore, we use "xu1998hz/InstructScore". For the translation model Madlad400-10b, we use "google/madlad400-10b-mt". We used GPT-3.5-Turbo and GPT-4 from OpenAI platform (https://platform.openai.com). We use gemini-pro from Google Gemini API.

B Quantile Mapping

While BLEURT (Sellam et al., 2020) correlates highly with human judgments (Freitag et al., 2022), its scale of roughly 0 to 1 is incompatible with the MQM human annotations, which range from -25 to 0. A linear mapping is not feasible, as the BLEURT score is not calibrated to the human score, meaning a BLEURT score of 0.8 does not correspond to -5 in MQM annotations.

To address this issue, we employ quantile mapping (Cannon et al., 2015) to transform the BLEURT score into the distribution of human scores. This method involves learning a mapping function that maps the quantiles or percentiles of the predictive distribution to those of the observed distribution. In this case, our predictive distribution is derived from the BLEURT score distribution, while our observed distribution comes from the corresponding human score distribution.

We utilize the WMT22 shared metric task (Freitag et al., 2022) to obtain mapped BLEURT-human scoring pairs. In this shared metric task, each translation generated by different translation model is rated by humans using the MQM human rating scale. We also run BLEURT on the same set of translations to obtain BLEURT scores, resulting in 28125 mapped BLEURT-human scoring pairs.

We then perform the following steps: 1) Separately sort the data of the two distributions in ascending order. 2) Compute the cumulative distribution function (CDF) for each distribution. 3) Learn an interpolation function that maps the percentiles of the first distribution to the percentiles of the second distribution. 4) Apply the mapping function to the values drawn from the predictive distribution (BLEURT score distribution) to obtain the corresponding values in the observed distribution (human MQM score distribution).

This process maps the BLEURT score distribution to the human score distribution (from -25 to 0) while preserving the relative ordering of BLEURT scores. In our experiments, we used the latest BLEURT model, BLEURT-20 checkpoint (Pu et al., 2021), which demonstrates the highest correlation to the human judgments among its variants.

C Gemini's Skewness at Translation

Specifically, in the Java-English (Jav-En) language pair, Gemini initially assigns lower quality scores to its output compared to BLEURT assessments during early iterations, resulting in an underestimation of output performance. This phenomenon accounts for the decrease in distance skewness at the beginning, as the right-skewed distribution becomes more neutral. However, as bias accumulates in later iterations, the distribution shifts towards a left-skewed distribution, leading to an increase in distance skewness.

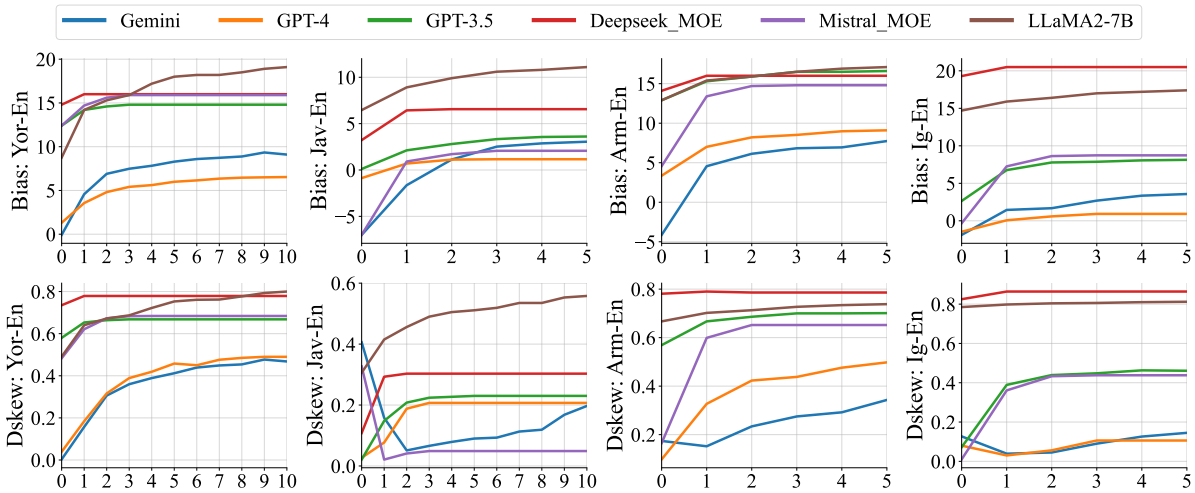

Figure 10: Full Bias and Dskew estimations for Yor-En, Jav-En, Arm-En, and Ig-En translations on FLores200, with the x-axis showing self-refine steps, reveal that all LLMs exhibit self-bias, where open-source LLMs exhibit higher levels than GPT-4 and Gemini.

External Feedback Example at GPT-4

Yoruba text: Ní bayii a ni àwon eku oloshu merin ti ko ni dayabetesi telele to ti ni ayabetesi," o she afikun.
Reference English text: "We now have 4-month-old mice that are non-diabetic that used to be diabetic," he added.
(Red span indicates a major error and blue span indicates a minor error annotated by GPT-4.)

GPT-4's 1st generation [Human: -11, InstructScore: -10, Bias: 1]: "At this point, we have four rats without diabetes that have developed diabetes," he added.

GPT-4's 1st refinement [Human: -2, InstructScore: -6, Bias: -4]: "At this point, we have four mice without diabetes that were diabetic," he added.

GPT-4's 2nd refinement [Human: -1, InstructScore: -1, Bias: 0]: "We now have 4-month-old mice that are non-diabetic that were diabetic," he added.

Table 3: This case study demonstrates that external feedback (oracle) from InstructScore (Xu et al., 2023) can remain low self-bias during iterative self-refine. By providing accurate error type, error location, and severity labels, InstructScore effectively elicits GPT-4's self-correction capability and improves its translation quality. Despite InstructScore's oracle-like role (which it can access reference text to make error annotations), it does not provide explicit corrections, requiring GPT-4 to rely on its internal knowledge for corrections.

Human Evaluation	Human	GPT-4	Bias	Dskew
0th Iteration	-15.0	-6.92	8.06	0.452
10th Iteration	-15.1	-0.52	14.6	0.692

Table 4: We report human evaluation and GPT-4's quality assessment on 0th and 10th iteration of refinement generation at Yor-En. We used Bias and Dskew estimation to demonstrate bias found by human evaluation.

Human Evaluation	Human	GPT-3.5	Bias	Dskew
0th Iteration	-22.2	-2.61	19.6	0.803
10th Iteration	-21.9	-0.03	21.9	0.885

Table 5: We report human evaluation and GPT-3.5-Turbo's quality assessment on the 0th and 10th iteration of refinement generation at Yor-En. We used Bias and Dskew estimation to demonstrate bias found by human evaluation.

Human Evaluation	Human	Gemini	Bias	Dskew
0th Iteration	-17.3	-8.92	9.62	0.355
10th Iteration	-18.3	-0.72	17.6	0.766

Table 6: We report human evaluation and Gemini's quality assessment on the 0th and 10th iterations of refinement generation at Yor-En. We used Bias and Dskew estimation to demonstrate bias found by human evaluation.

In-context-learning prompt for LLM's initial generation at translation:
Below is an instruction that describes a task.
Instruction:
Translate Chinese text into English.
Chinese: 新华时评：把优秀返乡农民工打造成乡村振兴生力军-新华网
English: Xinhua Commentary: Outstanding returning rural migrant workers can be a rural revitalization army - Xinhuanet

Below is an instruction that describes a task.
Instruction:
Translate English text into German.
English: You can come back any time as our chat service window is open 24/7
German: Sie können jederzeit wiederkommen, da unser Chat-Service-Fenster täglich rund um die Uhr geöffnet ist

Below is an instruction that describes a task.
Instruction:
Translate Yorba text into English.
Yorba: Won da Olori Skwodroni. Dilokrit Pattavee gege bi awako ofururu.
English: The pilot was identified as Squadron Leader Dilokrit Pattavee.

Below is an instruction that describes a task.
Instruction:
Translate Yoruba text into English.
Yorba: O ko ago ilekun WiFi, O wi.

Output for translation:
Can you please turn off the WiFi, I'm done.

Table 7: Those are the translation in context learning example we used to prompt all LLMs across four language directions at Flores200. In this example, the source translation is Yourba text "O ko ago ilekun WiFi, O wi.". The English output text is from LLaMA2-7B's generation "Can you please turn off the WiFi, I'm done.".

In-context-learning prompt for LLM's Self-feedback at translation:
You are an annotator for the quality of machine translation. Your task is to identify errors and assess the quality of the translation.
Based on the source segment and machine translation surrounded with triple backticks, identify error types in the translation and classify them. The categories of errors are: accuracy (addition, mistranslation, omission, untranslated text), fluency (character encoding, grammar, inconsistency, punctuation, register, spelling), locale convention (currency, date, name, telephone, or time format) style (awkward), terminology (inappropriate for context, inconsistent use), non-translation, other, or no-error.
Each error is classified as one of three categories: critical, major, and minor. Critical errors inhibit comprehension of the text. Major errors disrupt the flow, but what the text is trying to say is still understandable. Minor errors are technically errors, but do not disrupt the flow or hinder comprehension.

Source: "'大众点评乌鲁木齐家居商场频道为您提供高铁居然之家地址，电话，营业时间等最新商户信息，找装修公司，就上大众点评"' Translation: "'Urumqi Home Furnishing Store Channel provides you with the latest bussiness information such as the address, telephone number, bussiness hours, etc., of high-speed rail, and find a decoration company, and go to the reviews."' Annotate errors in the translation. MQM annotations:

"of high-speed rail" is a critical accuracy/addition error
"go to the reviews" is a major accuracy/mistranslation error
"etc.," is a minor style/awkwards error

Source: "'I do apologise about this, we must gain permission from the account holder to discuss an order with another person, I apologise if this was done previously, however, I would not be able to discuss this with yourself without the account holders permission."' Translation: "'Ich entschuldige mich dafür, wir müssen die Erlaubnis einholen, um eine Bestellung mit einer anderen Person zu besprechen. Ich entschuldige mich, falls dies zuvor geschehen wäre, aber ohne die Erlaubnis des Kontoinhabers wäre ich nicht in der Lage, dies mit dir involvement."' Annotate errors in the translation. MQM annotations:

'involvement' is a major accuracy/mistranslation error
'the account holder' is a major accuracy/omission error
'wäre' is a minor fluency/grammar error
'dir' is a minor fluency/register error

Source: "'Talks have resumed in Vienna to try to revive the nuclear pact, with both sides trying to gauge the prospects of success after the latest exchanges in the stop-start negotiations."' Translation: "'Ve Vídni se ve Vídni obnovily rozhovory o oživení jaderného paktu, přičemže obě partaje se snaží posoudit vyhlídky na úspěch po posledních výměnách v jednáních."' Annotate errors in the translation. MQM annotations:

've Vídni' is a major accuracy/addition error
'the stop-start' is a major accuracy/omission error
'partaje' is a minor terminology/inappropriate for context error

Source: "'Talks have resumed in Vienna to try to revive the nuclear pact, with both sides trying to gauge the prospects of success after the latest exchanges in the stop-start negotiations."' Translation: "'Ve Vídni se ve Vídni obnovily rozhovory o oživení jaderného paktu, přičemže obě partaje se snaží posoudit vyhlídky na úspěch po posledních výměnách v jednáních."' Annotate errors in the translation. MQM annotations:

Source: "'O ko ago ilekun WiFi, O wi."' Translation: "'He locked the WiFi door, he said."' Annotate errors in the translation. MQM annotations:

Output for Feedback:
'He locked the WiFi door' is a critical accuracy/mistranslation error

Table 8: Those are the feedback in context learning examples (three examples) that we used to prompt all LLMs across four language directions at Flores200. In this example, the source translation is Yourba text "O ko ago ilekun WiFi, O wi.". The English candidate text is "He locked the WiFi door" from GPT-4. GPT-4's self-feedback outputs "'He locked the WiFi door' is a critical accuracy/mistranslation error".

In-context-learning prompt for LLM's refinement at translation:
Please fix all errors. You can rewrite translation if translation is bad.

Source: "'大众点评乌鲁木齐家居商场频道为您提供高铁居然之家地址，电话，营业时间等最新商户信息，找装修公司，就上大众点评"'
Translation: "'Urumqi Home Furnishing Store Channel provides you with the latest bussiness information such as the address, telephone number, bussiness hours, etc., of high-speed rail, and find a decoration company, and go to the reviews."'

Feedback: 'of high-speed rail' is a critical accuracy/addition error
'go to the reviews' is a major accuracy/mistranslation error
'etc.,' is a minor style/awkward error

Improved Chinese-to-English translation:
Dianping Urumqi Renovation and Design Channel will provide you with the address, phone number, operation time and other information of HSR Easyhome, and please come to Dianping if you are looking for a renovation company.

Source: "'I do apologise about this, we must gain permission from the account holder to discuss an order with another person, I apologise if this was done previously, however, I would not be able to discuss this with yourself without the account holders permission."'
Translation: "'Ich entschuldige mich dafür, wir müssen die Erlaubnis einholen, um eine Bestellung mit einer anderen Person zu besprechen. Ich entschuldige mich, falls dies zuvor geschehen wäre, aber ohne die Erlaubnis des Kontoinhabers wäre ich nicht in der Lage, dies mit dir involvement."'

Feedback:
'involvement' is a major accuracy/mistranslation error
'the account holder' is a major accuracy/omission error
'wäre' is a minor fluency/grammar error
'dir' is a minor fluency/register error

Improved English-to-German translation:
Ich bitte um Entschuldigung, aber wir benötigen das Einverständnis des Kontoinhabers, um eine Bestellung mit einer anderen Person zu besprechen, falls es schon eingeholt wurde, entschuldige ich mich, aber ich kann dies ohne das Einverständnis des Kontoinhabers nicht mit Ihnen besprechen.
Source: "'Talks have resumed in Vienna to try to revive the nuclear pact, with both sides trying to gauge the prospects of success after the latest exchanges in the stop-start negotiations."'
Translation: "'Ve Vídni se ve Vídni obnovily rozhovory o oživení jaderného paktu, přičemže obě partaje se snaží posoudit vyhlídky na úspěch po posledních výměnách v jednáních."'

Feedback:
've Vídni' is a major accuracy/addition error
'the stop-start' is a major accuracy/omission error
'partaje' is a minor terminology/inappropriate for context error

Improved English-to-Cezch translation:
Ve Vídni byly obnoveny rozhovory o oživení jaderného paktu a obě strany se snaží odhadnout, jaké jsou vyhlídky na úspěch po posledních výměnách názorů v rámci přerušených jednání.

Source: "'O ko ago ilekun WiFi, O wi."'
Translation: 'He locked the WiFi door, he said."'

Feedback:
'He locked the WiFi door' is a critical accuracy/mistranslation error

Improved English-to-Yoruba translation:
Output for refinement:
"He secured the WiFi connection, he said."

Table 9: Those are the refinement in context learning examples (three examples) that we used to prompt all LLMs across four language directions at Flores200. In this example, the source translation is Yourba text "O ko ago ilekun WiFi, O wi.". The English text is GPT-4's output "He locked the WiFi door.". We gathered GPT-4's self-feedbak: "'He locked the WiFi door' is a critical accuracy/mistranslation error". GPT-4 outputs a refined output: "He secured the WiFi connection, he said.", according to the feedback and prior step generation.

Prompt for LLM's initial generation for reasoning:
You are a competitive math problem solver. Please generate a step-by-step solution. Your final answer must be enclosed in LaTeX's boxed tag.

How many vertical asymptotes does the graph of $y = \frac{2}{x^2+x-6}$ have?

Output for math answer:
A vertical asymptote is a vertical line that the graph of a function approaches but never crosses. A function has a vertical asymptote at $x = k$ if it has infinite limits as x approaches k from the left or right.For the rational function $y = \frac{2}{x^2+x-6}$, we know that it could have vertical asymptotes wherever the denominator is zero, because division by zero is undefined and yields infinity.Therefore, to find the vertical asymptotes, we solve the equation $x^2 + x - 6 = 0$. We apply factorization and get $(x - 2)(x + 3) = 0$. This gives us $x = 2$ and $x = -3$.In any case, we need to check if any of these could be holes instead of vertical asymptotes. This would be the case if $x = 2$ or $x = -3$ also made the numerator zero. However, the numerator is a constant and won't be zero. Therefore this function has two vertical asymptotes at $x = 2$ and $x = -3$. The answer is $\boxed{2}$.

Table 10: This is the prompt we used to generate mathematical reasoning. The output in this example comes from GPT-4. We first used this prompt to generate one reasoning path as initial generation. Then, we sample 10 different reasoning paths and use majority vote to find the most confident answer. To generate self-feedback, we compare majority voted answer and initial answer. If they are matched, feedback indicates answer is correct and no change should be made. If they are not matched, the feedback indicates answer is wrong and inital answer will be replaced by the majority voted answer. We alternate above two steps and iteratively refine the reasoning paths.

Concept-to-text prompt for LLM's initial generation:
You are generating text based on specified words. Do not provide any explanations or text apart from the text output.

Please generate a sentence that contains the exact string matches for the following concepts:

['vest', 'snow', 'remove', 'silhouette', 'dunk', 'step', 'item', 'toddler', 'plaster', 'wall', 'piece', 'tool', 'circle', 'pull', 'wrap', 'headphone', 'deck', 'suit', 'press', 'mother', 'pit', 'cupcake', 'priest', 'rink', 'counter', 'rest', 'piano', 'wine', 'serve', 'winter']

Output for Concept-to-Text generation:

In the winter, the toddler wearing a vest, pressed a piece of wet plaster against the wall in a circle, while his mother baking a cupcake was removing the snow from the deck with a tool; switching steps to the rhythm of a piano piece echoing from the headphones wrapped around her neck, she caught a glimpse of the priest serving wine, his silhouette dunking a cupcake into a pit at the rink counter, under the restful gaze of a suit-clad man.

Table 11: This is the concept-to-text prompt we used to test all LLMs at Commongen Hard. This example contains the output from GPT-4.

Concept-to-text in context learning prompt for LLM's self-feedback:
We want to create a sentence that contains all the specified concepts. Please provide feedback on the following sentences. The feedback should list all missing concepts. If all concepts are covered, output 'all covered'

Concepts: ['dog', 'frisbee', 'catch', 'throw']
Generated Sentence: A dog leaps to catch a thrown frisbee.
Feedback: all covered

Concepts: ['dog', 'frisbee', 'catch', 'throw']
Generated Sentence: Two dogs are throwing frisbees at each other .
Feedback: ['catch']

Concepts: ['vest', 'snow', 'remove', 'silhouette', 'dunk', 'step', 'item', 'toddler', 'plaster', 'wall', 'piece', 'tool', 'circle', 'pull', 'wrap', 'headphone', 'deck', 'suit', 'press', 'mother', 'pit', 'cupcake', 'priest', 'rink', 'counter', 'rest', 'piano', 'wine', 'serve', 'winter']
Generated Sentence: In the winter, the toddler wearing a vest, pressed a piece of wet plaster against the wall in a circle, while his mother baking a cupcake was removing the snow from the deck with a tool; switching steps to the rhythm of a piano piece echoing from the headphones wrapped around her neck, she caught a glimpse of the priest serving wine, his silhouette dunking a cupcake into a pit at the rink counter, under the restful gaze of a suit-clad man.
Feedback:

Output for Concept-to-Text feedback:
all covered

Table 12: This is the in-context learing (ICL) concept-to-text prompt (two ICL examples) we used to generate LLM's self-feedback at Commongen Hard. This example contains the self-feedback from GPT-4.

Concept-to-text prompt for LLM's refinement:
Concepts: ['fruit', 'motorcycle', 'perform', 'jacket', 'vehicle', 'place', 'mat', 'walk', 'world', 'area', 'kiss', 'mother', 'pass', 'report', 'club', 'axis', 'tricep', 'patient', 'listen', 'owner', 'uniform', 'floor', 'hamburger', 'use', 'wine', 'cross', 'bull', 'sell', 'lawn', 'friend']
Generated sentence: In a world where a fruit can perform like a motorcycle, I found my mother wearing a leather jacket, passing by her vehicle to walk towards the place where she runs a club. Not far from that area, my friend, a patient person, was ready to sell mats, used for gym workouts, especially for triceps. Being the owner of the shop, he always wears a uniform. While I was listening to this report, sitting cross-legged on the floor and munching my hamburger, I missed seeing a bull wandering on the lawn just outside. Just then, I received an unexpected kiss from my wife who brought me a glass of wine from our cellar which also acts as an axis for all the happenings of my life.
Missing Concepts: ['use', 'lawn']
Please revise generated sentence which covers all missing concepts and all given concepts, with the exact string matches.
New sentence:

Output for Concept-to-Text refinement:
In a world where a fruit can perform like a motorcycle, I found my mother wearing a leather jacket, passing by her vehicle to walk towards the place where she runs a club. Not far from that area, my friend, a patient person, was ready to sell mats, used for gym workouts, especially for triceps. Being the owner of the shop, he always wears a uniform. While I was listening to this report, sitting cross-legged on the floor and munching my hamburger, I missed seeing a bull wandering on the lawn just outside. Just then, I received an unexpected kiss from my wife who brought me a glass of wine from our cellar which also acts as an axis for all the happenings of my life.

Table 13: This is the concept-to-text refinement prompt we used to generate LLM's refinement at Commongen Hard. This example contains the refinement from GPT-4.