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 (Costa-jussà et al., 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 8, 9 and 10.

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 12, 13 and 14.

To evaluate the coverage of the generated texts, we adopted the evaluation metric used in (Madaan et al., 2024). 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 use the self-refine pipeline by providing the feedback on the step-by-step solution. 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 11. In addition, we also conducted experiments by replacing the self-evaluation (LLM as evaluator) with self-consistency verification (self-consistency as an evaluator) (Huang et al., 2023a). We include those results in the Appendix D.

4.2 Self-Bias Amplification during Iterative Refinement

Machine Translation. In Figure 3, we illustrate that all large language models (LLMs) exhibit a

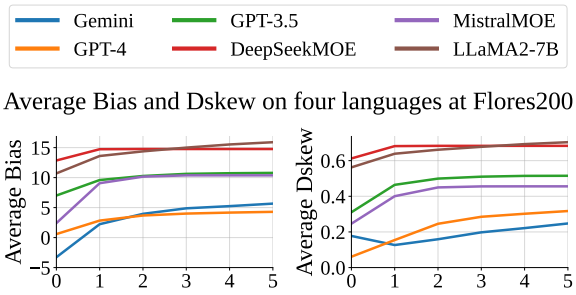

Figure 3: Average Bias and Dskew estimations for Yor-En, Jav-En, Arm-En, and Ig-En translations on FLores200, with the x-axis showing self-refine steps, reveal that all LLMs exhibit self-bias, where open-source LLMs exhibit higher levels than GPT-4 and Gemini.

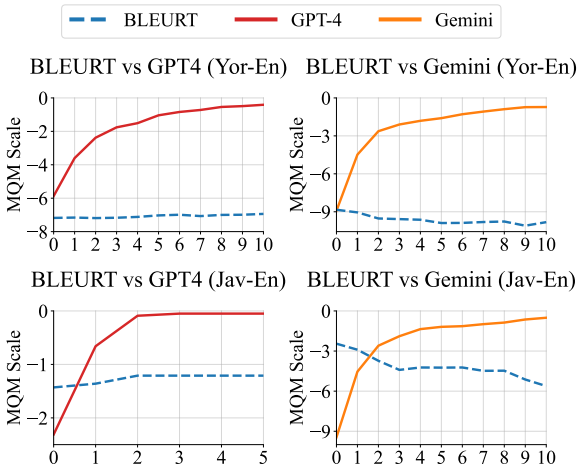

Figure 4: GPT-4 and Gemini overestimate improvements in self-refined outputs, leading to amplified bias over iterations compared to actual performance measured by BLEURT.

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 4, 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 measure grows larger with each iteration. 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 5 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.

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 6 illustrates that all large language models (LLMs) exhibit an increase in bias and skewness estimation in the iterative self-refine pipeline. This suggests that LLMs introduce self-biases towards some math solutions during self-refine.

Human Evaluation on Bias Estimation. 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 received measurable improvements via the self-refine pipeline (The raw human scores

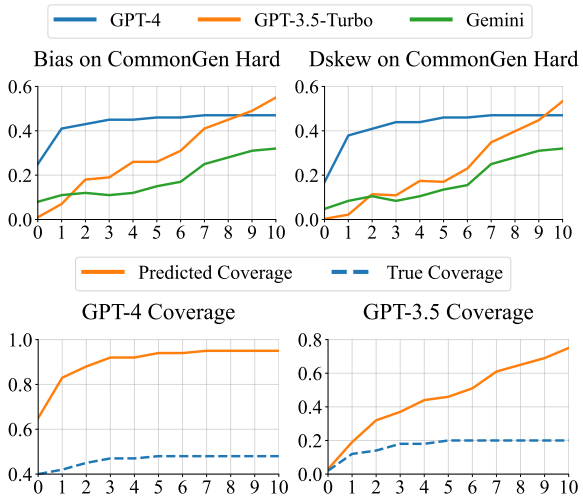

Figure 5: We evaluate the bias and distance skewness of generated texts produced by GPT-4, GPT-3.5-Turbo, and Gemini on the CommonGen dataset, across self-refinement steps. Additionally, we report the coverage of GPT-3.5-Turbo and GPT-4 compared to true concept coverage. We show that the rate of LLM's self-estimated improvements is much higher than the true coverage improvements, which leads to self-bias amplification.

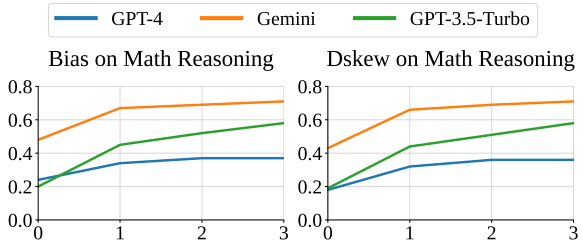

Figure 6: Bias and distance skewness in generated texts from GPT-4, GPT-3.5-Turbo, and Gemini are measured on MATH testing set throughout the self-refinement steps. Results show an increase in bias and skewness of some math solutions during iterative self-refine.

are included in the Appendix Table 5, 6 and 7), 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.

	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.

Human Evaluation on LLM's Output Quality. We conducted human evaluation on six LLM's self-feedback outputs at first and fifth iteration at Yoruba to English translation. For each LLM at each iteration, we annotate 100 samples. In total, we annotate 1200 samples. Specifically, human labor will check whether error annotation in the format of 'xxx' is a minor xxx error/'xxx' is a major xxx error/'xxx' is a critical xxx error (When LLM outputs an error-free annotations, it can have flexible forms, such 'None', 'No error', "Perfect translation").

In Table 2, we include format accuracy for all LLMs. We observed that all LLMs have either perfect or nearly perfect format at first and fifth iteration of self-feedback. This is expected as we explicitly provide three in-context examples to control the output format. We found that different LLMs make different format mistakes. For example, DeepSeekMOE produces one or two garbage outputs and GPT-3.5-Turbo produces two or three free form outputs, like "The machine translation