

Large Language Models are not Fair Evaluators

Peiyi Wang¹ Lei Li¹ Liang Chen¹ Zefan Cai¹ Dawei Zhu¹

Binghuai Lin³ Yunbo Cao³ Lingpeng Kong² Qi Liu² Tianyu Liu³ Zhifang Sui¹

¹ State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University

² The University of Hong Kong ³ Tencent Cloud AI

{wangpeiyi9979, nlp.lilei, zefncai}@gmail.com

leo.liang.chen@outlook.com; {dwzhu, szf}@pku.edu.cn

{lpk, liuqi}@cs.hku.hk;

{binghuailin, yunbocao, rogertyliu}@tencent.com

Abstract

In this paper, we uncover a positional bias in the evaluation paradigm of adopting large language models (LLMs), e.g., GPT-4, as a referee to score and compare the quality of responses generated by candidate models. We find that the quality ranking of candidate responses can be easily hacked by simply altering their order of appearance in the context. This manipulation allows us to skew the evaluation result, making one model appear considerably superior to the other, e.g., Vicuna-13B could beat ChatGPT on 66 over 80 tested queries with ChatGPT as an evaluator. We propose a simple yet effective calibration framework to address our discovered positional bias. To evaluate the effectiveness of our framework, we manually annotate the “win/tie/lose” outcomes of responses from ChatGPT and Vicuna-13B in the Vicuna Benchmark’s question prompt. Extensive experiments demonstrate that our approach successfully alleviates evaluation bias, resulting in closer alignment with human judgments. To facilitate future research on more robust large language model comparison, we integrate the techniques in the paper into an easy-to-use toolkit FairEval, along with the human annotations ¹.

1 Introduction

The rapid advancement of Large Language Models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022) has underscored the importance of evaluating their alignment with human intent in generated responses, making it an active field of research. Traditional n-gram metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), as well as more sophisticated model-based evaluations such as BERTScore (Zhang et al., 2020) and BARTScore (Yuan, Neubig, and Liu, 2021), are insufficient for thoroughly assessing this alignment (He et al., 2023). While human evaluation provides

¹<https://github.com/i-Eval/FairEval>

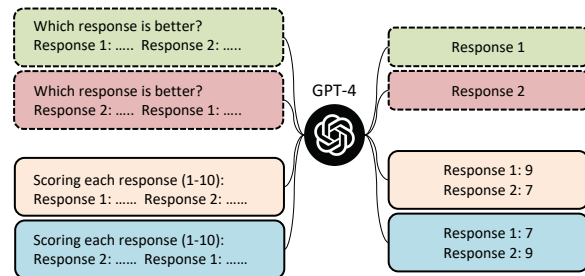


Figure 1: Simply changing the order of candidate responses leads to overturned comparison results, even though we add the command “ensuring that the order in which the responses were presented does not affect your judgment” into the prompt.

the most accurate measure of model performance and valuable insights, it can often be costly and time-consuming. As a result, there is a growing demand for automated assessment methods that can consistently align with human judgments while being more efficient and cost-effective.

ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) have recently demonstrated remarkable performance across various tasks, leading to their widespread use as both the annotators (Peng et al., 2023; Xu et al., 2023) and evaluators (Zheng et al., 2023; Peng et al., 2023; Sun et al., 2023; Zhou et al., 2023; Gao et al., 2023; Wang et al., 2023b; Dubois et al., 2023; Wang et al., 2023a). For example, The evaluation pipeline of Vicuna (Zheng et al., 2023) has gained significant interest and wide usage due to its simplicity and interpretability. It prompts GPT-4 to score and compare candidate responses and provide explanations, making it a valuable tool for evaluation. However, it is unclear how reliable LLMs are as evaluators, as they are known to be sensitive to textual instructions and inputs (Dong et al., 2022; Turpin et al., 2023; Bowman, 2023). This raises questions about the resilience of this paradigm against perturbations, such as the ordering of candidates during scoring,

potentially becoming the Achilles’ Heel that can be easily hacked for unreliable evaluations.

In this paper, we take a sober look at the LLMs-as-evaluator paradigm and uncover a significant positional bias. Specifically, we demonstrate that GPT-4 exhibits a preference for the first displayed candidate response by consistently assigning it higher scores, even when the order of candidates is subtly altered. As illustrated in Figure 1, merely swapping the presentation order can reverse evaluation outcomes. This bias is also present in ChatGPT, which typically favors the second response. These findings highlight previously overlooked limitations in the current evaluation paradigm.

To address this issue, we propose three simple yet effective strategies to calibrate positional bias: **1) Multiple Evidence Calibration (MEC)**: We prompt the model to generate evaluation evidence before assigning scores, leveraging the inherent properties of causal language models for calibration. We also employ ensemble techniques to incorporate multiple evidence calibration results to further stabilize the evaluation. **2) Balanced Position Calibration (BPC)**: To further reduce positional bias, we evaluate each candidate in both positions across two runs and compute the final score as the average of the two runs. **3) Human In The Loop Calibration (HITLC)**: We also explore human-in-the-loop evaluation and consider a diversity-based method to get a cue to indicate biased candidates based on the evaluation results of MEC and BPC.

To assess the efficacy of our methods, we manually annotate the “win/tie/lose” outcomes of responses from ChatGPT and Vicuna-13B in the Vicuna benchmark (Zheng et al., 2023), encompassing 80 questions spanning 9 distinct question categories. Our MEC and BPC enhance the evaluation alignment of GPT-4 and ChatGPT by 9.8% and 14.3% accuracy, respectively. Moreover, based on MEC and BPC, our HITLC can further effectively integrate human assistance into the evaluation process. Specifically, with only a 20% human annotation cost, GPT-4 and ChatGPT can achieve comparable or even better annotation alignment with the average human performance, reducing the annotation cost by up to 39%.

In summary, our key contributions are: **1)** We reveal that LLMs exhibit severe positional bias, compromising their fairness as evaluators; **2)** We develop a calibration framework with three simple yet effective strategies to calibrate the positional bias of

| |
|---|
| [Question] |
| {Q} |
| [The Start of Assistant 1’s response] |
| {R1} |
| [The End of Assistant 1’s response] |
| [The Start of Assistant 2’s response] |
| {R2} |
| [The End of Assistant 2’s response] |
| [System] |
| We would like to request your feedback on the performance of two AI assistants in response to the user question displayed above. |
| Please rate the helpfulness, relevance, accuracy, level of details of their responses. Each assistant receives an overall score on a scale of 1 to 10, where a higher score indicates better overall performance. |
| Please first output a single line containing only two values indicating the scores for Assistant 1 and 2, respectively. |
| The two scores are separated by a space. In the subsequent line, please provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment. |

Table 1: The evaluation template with three slots ($\{Q\}$, $\{R1\}$ and $\{R2\}$) from Zheng et al. (2023). Even though the template emphasizes not letting the order affect the results (red text), large language models still have a large positional bias.

LLMs; **3)** We manually annotate the “win/tie/lose” outcomes of responses from ChatGPT and Vicuna-13B in the Vicuna benchmark and demonstrate the effectiveness of our proposed approach through experimental results, which show closer alignment with human judgments.

2 Positional Bias of the LLM Evaluator

2.1 LLMs as Evaluators

Recently, a lot of works (Zheng et al., 2023; Wang et al., 2023a; Sun et al., 2023; Zhou et al., 2023) have been utilizing LLMs such as GPT-4 as evaluators to compare the performance of two AI assistants. Table 1 illustrates a commonly used evaluation template from the Vicuna benchmark (Zheng et al., 2023). The template has three placeholders $T(Q, R1, R2)$, and is used to query the LLM for evaluation. For each testing question q , given two responses $r1$ and $r2$ from Assistant 1 and Assistant 2, respectively, the researchers populate these responses into the corresponding slots of the evaluation template to form a prompt: $T(Q = q, R1 = r1, R2 = r2)$. The prompt is then used to query the LLM in order to obtain the comparison result. In this paper, we found that LLM suffers from severe positional bias, i.e., by swapping the slots of the two responses and

| EVALUATORS | VICUNA-13B v.s. OTHER MODELS | VICUNA-13B WIN RATE | | CONFLICT RATE |
|------------|------------------------------|---------------------|---------------|-----------------|
| | | AS ASSISTANT1 | AS ASSISTANT2 | |
| GPT-4 | Vicuna-13B v.s. ChatGPT | 51.3% | 23.8% | 37 / 80 (46.3%) |
| GPT-4 | Vicuna-13B v.s. Alpaca-13B | 92.5% | 92.5% | 4 / 80 (5.0%) |
| ChatGPT | Vicuna-13B v.s. ChatGPT | 2.5% | 82.5% | 66 / 80 (82.5%) |
| ChatGPT | Vicuna-13B v.s. Alpaca-13B | 37.5% | 90.0% | 42 / 80 (52.5%) |

Table 2: The Win Rate of Vicuna-13B significantly fluctuates when positioned as Assistant 1 and Assistant 2, with GPT-4 and ChatGPT serving as evaluators. CONFLICT RATE refers to the proportion of conflicting results given by the same evaluator when simply changing the position of two models.

querying LLM twice, the evaluator will most likely produce conflicting evaluation results, and the evaluator prefers the response at a certain position.

2.2 Revealing the Positional Bias

In this section, we adopt GPT-4 and ChatGPT as evaluators to analyze the characteristics of positional bias in LLM evaluators. We find that:

LLMs are sensitive to the position of responses.

As shown in Table 2, in the evaluation of “Vicuna-13B v.s. ChatGPT” and “Vicuna-13B v.s. Alpaca-13B”, when the order was changed, LLMs provide different evaluation results, e.g., the win rate of Vicuna-13B extremely differs when Vicuna-13B is evaluated as Assistant 1 and Assistant 2.

To empirically evaluate the sensitivity, we introduced a metric **Conflict Rate** to measure the sensitivity of the model to response positions quantitatively. Formally, given N examples $\{(q_i, r1_i, r2_i)\}_{i=1}^N$, for each example $(q_i, r1_i, r2_i)$, we query the LLM with two prompts $T(q_i, r1_i, r2_i)$ and $T(q_i, r2_i, r1_i)$, and obtain corresponding two evaluation results \mathbf{ER}_i^{r12} and \mathbf{ER}_i^{r21} . Then we calculate the Conflict Rate of the LLM evaluator as follows:

$$\text{Conflict Rate} = \frac{\sum_{i=1}^N \mathbb{I}(\mathbf{ER}_i^{r12} \neq \mathbf{ER}_i^{r21})}{N}, \quad (1)$$

where $\mathbb{I}(\cdot)$ is the indicator function. We found that GPT-4 exhibited conflict rates of 46.3% and 5.0%, respectively. In contrast, ChatGPT displayed considerably higher conflict rates, with figures of 82.5% and 52.5%, respectively. These findings indicate that LLMs can be self-conflicting due to the sensitivity of the response order in the template, with stronger models being less influenced by the placement of responses.

LLMs suffer from Positional Bias, i.e., they prefer the response in the specific position. Based

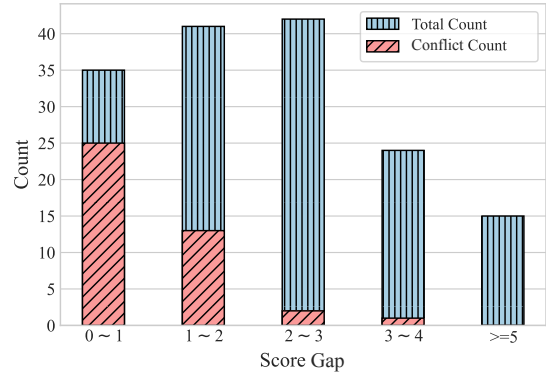


Figure 2: The conflict rate is negatively correlated with the score gap between the two responses. When swapping the order of two responses, the smaller the score gap between them, the more likely GPT-4 is to produce conflicting results.

on the same evaluation template T in Table 1, GPT-4 tends to favor the response in the first position, while ChatGPT shows a preference for the response in the second position. For example, as illustrated in Table 2, in the comparison “Vicuna-13B v.s. ChatGPT”, GPT-4 yields Win Rates of 51.3% and 23.8% for Vicuna-13B when it is positioned as Assistant 1 and Assistant 2, respectively. Conversely, ChatGPT indicates Win Rates of only 2.5% and up to 82.5% for Vicuna-13B when it is positioned as Assistant 1 and Assistant 2, respectively.

The degree of positional bias varies based on the difference in response quality.

We notice that the conflict rate of “Vicuna-13B v.s. Alpaca-13B” is much lower than that of “Vicuna-13B v.s. ChatGPT”, suggesting that positional bias may not have the same impact on the assessment of different responses. One potential reason is that there is a significant difference in the quality of responses between Alpaca models and Vicuna models, and positional bias is not strong enough to change the judgment in such a situation. To further investigate this issue, we grouped all the examples based on

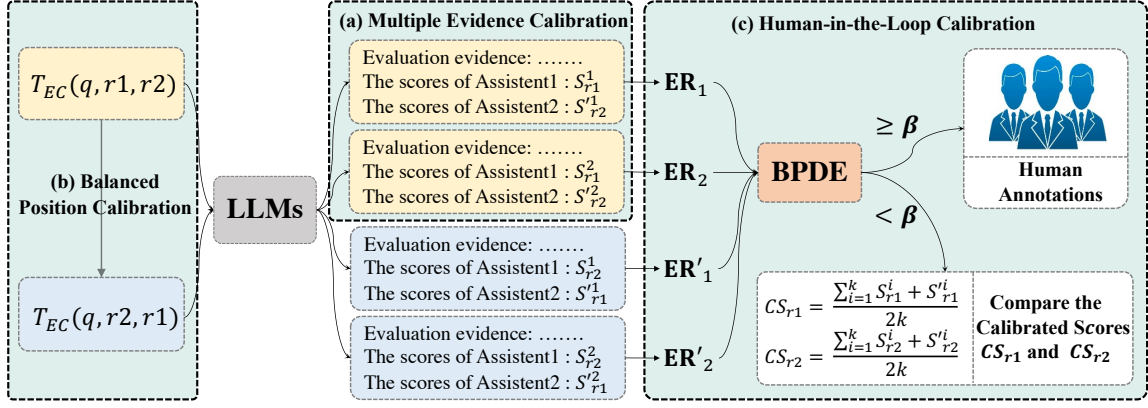


Figure 3: Demonstration of our calibration framework with three calibration methods. S_r and S'_r denotes scores of the response r in the first and second positions, respectively. BPDE is short for Balanced Position Diversity Entropy score, which is calculated based on the evaluation results (ER) of MEC and BPC.

| |
|---|
| [Question] |
| {Q} |
| [The Start of Assistant 1's response] |
| {R1} |
| [The End of Assistant 1's response] |
| [The Start of Assistant 2's response] |
| {R2} |
| [The End of Assistant 2's response] |
| [System] |
| |
| Please first provide a comprehensive explanation of your evaluation, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment. Then, output two lines indicating the scores for Assistant 1 and 2, respectively. |
| Output with the following format: |
| Evaluation evidence: <evaluation explanation here> |
| The score of Assistant 1: <score> |
| The score of Assistant 2: <score> |

Table 3: The evidence calibration evaluation template that prompts LLMs to generate the evaluation evidence first (red text), and then evaluate two responses.

the score difference between the two responses. As shown in Figure 2, we found that when the score difference between the two responses is small (e.g., score gap ≤ 1), the evaluation results of GPT-4 are significantly affected by the position of the responses. On the other hand, when the score difference between the two responses is large (e.g., score gap ≥ 3), GPT-4's evaluation results are relatively stable.

2.3 Calibrating the Positional Bias

We have identified that positional bias can significantly impact the evaluation results of LLMs, making them unfair evaluators. In this section, we propose a calibration framework with three simple yet

effective strategies to alleviate this bias to achieve a more reliable and fair evaluation result.

2.3.1 Multiple Evidence Calibration

Previous studies (Zheng et al., 2023; Wang et al., 2023b) utilize the evaluation template that draws the conclusion first and then makes an explanation, e.g., the template used in Table 1. However, due to the nature of the auto-regressive model, the conclusions generated by the model are not supported by the explanation generated afterward. To this end, as shown in Table 3, we design an evidence calibration (EC) evaluation template $T_{EC}(Q, R1, R2)$ that requires the model to generate the explanation (evaluation evidence) first and then give the score. In this way, the score can be calibrated with the evaluation evidence. To further improve the reliability of the evaluation, rather than generating only a single EC score for each response, we perform a multiple evidence calibration (MEC, Figure 3(a)) that samples k EC scores $\{S_{r1}^1, \dots, S_{r1}^k\}$ and $\{S_{r2}^1, \dots, S_{r2}^k\}$ for responses $r1$ and $r2$, where S_r and S'_r denotes scores of the response r at the first and second positions, respectively.

2.3.2 Balanced Position Calibration

We further employ a balanced position calibration (BPC) strategy to alleviate the previously identified positional bias of LLMs. As shown in Figure 3(b), for each example $(q, r1, r2)$, BPC additionally creates a query prompt $T_{EC}(q, r2, r1)$ by swapping the position of two responses in the original query prompt $T_{EC}(q, r1, r2)$. Combined with MEC, we can achieve $2k$ scores $\{S_{r1}^1, \dots, S_{r1}^k, S_{r1}^1, \dots, S_{r1}^k\}$ and $\{S_{r2}^1, \dots, S_{r2}^k, S_{r2}^1, \dots, S_{r2}^k\}$ for $r1$ and $r2$,

respectively. The final calibrated scores of two responses (CS_{r1} and CS_{r2}) are the average of the $2k$ scores:

$$CS_R = \sum_{i=1}^k \frac{S_R^i + S'_R{}^i}{2k}, R = r1, r2 \quad (2)$$

and we regard the response with the higher average score as the better response.

2.3.3 Human-in-the-Loop Calibration

In addition to the automatic calibration strategies, another interesting question we want to explore is whether Human-In-The-Loop Calibration (HITLC) which performs the cooperation of humans and LLMs as evaluators, could stabilize the evaluation result. The key point of human-in-the-loop calibration is when humans should be involved in the evaluation and calibrate the evaluation result on which LLM evaluators do not perform well.

To target the “when” problem, we introduce a **Balanced Position Diversity Entropy (BPDE)** score to find examples requiring auxiliary human calibration based on the evaluation results of MEC and BPC. Specifically, as shown in Figure 3(c), we first compute $2k$ evaluation results $\{\mathbf{ER}_i\}_{i=1}^{2k}$ based on the $2k$ pairs of scores.

$$\mathbf{ER}_i = \begin{cases} \text{win}, S_{r1}^i > S_{r2}^i \\ \text{tie}, S_{r1}^i = S_{r2}^i \\ \text{lose}, S_{r1}^i < S_{r2}^i \end{cases}, \mathbf{ER}'_i = \begin{cases} \text{win}, S'_{r1}{}^i > S'_{r2}{}^i \\ \text{tie}, S'_{r1}{}^i = S'_{r2}{}^i \\ \text{lose}, S'_{r1}{}^i < S'_{r2}{}^i \end{cases}, \quad (3)$$

and BPDE is defined as the entropy of the evaluation results:

$$\text{BPDE} = \sum_{\text{er} \in \{\text{win}, \text{tie}, \text{lose}\}} -\text{p}_{\text{er}} \log \text{p}_{\text{er}} \quad (4)$$

$$\text{p}_{\text{er}} = \frac{\sum_{i=1}^k \mathbb{I}(\mathbf{ER}_i = \text{er}) + \mathbb{I}(\mathbf{ER}'_i = \text{er})}{2k}. \quad (5)$$

A higher BPDE score indicates that it is more likely the evaluation requires manual correction. A threshold is needed for BPDE as the hyper-parameter to select the top- β most likely biased evaluations. After selection based on the BPDE score, the annotators will evaluate the selected examples and integrate the human annotations based on the majority opinion as described in Section 3.1.

3 Experiments

3.1 Human Annotation

To assess the effectiveness of our proposed strategies, three of the authors manually annotate the

“win/tie/lose” outcomes of responses from ChatGPT and Vicuna-13B independently in all 80 Vicuna Benchmark questions. All of the annotators are researchers familiar with Artificial Intelligence and are well-equipped to assess the quality of the responses. Following the same template as the original Vicuna, the annotators are instructed to assess the responses provided by Vicuna-13B and ChatGPT from four different perspectives: helpfulness, relevance, accuracy, and level of detail. The responses of Vicuna and ChatGPT are presented to the annotators in random order. The evaluation process for each example took an average of three minutes. **The final result is based on the majority opinion among three annotators.**

3.2 Experimental Setup and Metric

We use the OpenAI API to conduct our experiments (“gpt-3.5-turbo” for ChatGPT, and “gpt-4” for GPT-4). For the methods that do not need to sample multiple generation results, we set the generated temperature to 0 for deterministic generation results. For the multiple evidence strategy, we set the temperature to 1 and $k = 3$. We use the accuracy and kappa correlation coefficient (McHugh, 2012) with the final majority of human annotation results to measure the performance of different evaluators and evaluation methods. When calculating the results for methods that do not utilize BPC, we randomize the order of the two responses from the assistants and calculate the average results of 100 runs to ensure stable results.

3.3 Main Results

Table 4 illustrates the performance of different methods on our manually annotated 80 annotated examples. As is shown: **1)** There is a good correlation coefficient between the annotations provided by each human annotator and the final voting results. In detail, the average accuracy and the kappa correlation coefficient of human annotations are 71.7% and 0.54, respectively; **2)** Overall, GPT-4 achieves higher alignment with human judgments compared with ChatGPT, showing its powerful alignment ability with humans; **3)** Compared to the commonly used VANILLA evaluation method, our proposed automatic calibration strategies (i.e., EC, MEC and BPC) significantly enhance the alignment between GPT-4 and ChatGPT with human judgments; For instance, by employing the MEC and BPC calibration strategies, ChatGPT demonstrates a notable improvement in both accuracy and

| EVALUATORS | METHODS | ACCURACY | KAPPA | COST |
|---------------|--|----------|-------|--------|
| Human 1 | - | 68.8% | 0.50 | \$30.0 |
| Human 2 | - | 76.3% | 0.62 | \$30.0 |
| Human 3 | - | 70.0% | 0.50 | \$30.0 |
| Human Average | - | 71.7% | 0.54 | \$30.0 |
| GPT-4 | VANILLA | 52.7% | 0.24 | \$2.00 |
| GPT-4 | EC ($k = 1$) | 56.5% | 0.29 | \$2.00 |
| GPT-4 | MEC ($k = 3$) | 58.7% | 0.30 | \$3.19 |
| GPT-4 | MEC ($k = 6$) | 60.9% | 0.33 | \$4.98 |
| GPT-4 | MEC ($k = 3$) + BPC ($k = 3$) | 62.5% | 0.37 | \$6.38 |
| GPT-4 | MEC ($k = 3$) + BPC ($k = 3$) + HITLC ($\beta = 20\%$) | 73.8% | 0.56 | \$23.1 |
| ChatGPT | VANILLA | 44.4% | 0.06 | \$0.10 |
| ChatGPT | EC ($k = 1$) | 52.6% | 0.23 | \$0.10 |
| ChatGPT | MEC ($k = 3$) | 53.2% | 0.24 | \$0.17 |
| ChatGPT | MEC ($k = 6$) | 55.6% | 0.27 | \$0.28 |
| ChatGPT | MEC ($k = 3$) + BPC ($k = 3$) | 58.8% | 0.31 | \$0.34 |
| ChatGPT | MEC ($k = 3$) + BPC ($k = 3$) + HITLC ($\beta = 20\%$) | 71.3% | 0.52 | \$18.3 |

Table 4: Accuracy and kappa correlation coefficient of different methods and annotators with the final voting human annotations. The VANILLA evaluation method was commonly used in previous works, which provided the conclusion first and then followed with the explanation. (M)EC, BPC, and HITLC denote our proposed (multiple) evidence calibration, balanced position calibration, and human-in-the-loop calibration respectively. $\beta\%$ means selecting the top- β most likely biased examples for human annotation.

the kappa correlation coefficient. Specifically, the accuracy is improved by 14.3%, and the kappa correlation coefficient is increased from 0.06 to 0.31; **4**) “MEC ($k = 3$) + BPC ($k = 3$)” outperforms “MEC ($k = 6$)”, demonstrating that LLMs are affected by positional bias, and BPC effectively ensures that LLMs serve as fair evaluators; **5**) Our proposed HITLC can effectively enhance the alignment between GPT-4 and ChatGPT with human judgments, requiring only a small amount of human labor. For example, by incorporating just 20% ($\beta = 20\%$) human assistance, ChatGPT attains comparable Human Average accuracy, while reducing the annotation cost from \$30 to \$18.3, a 39% reduction.²

In conclusion, our proposed calibration methods are simple yet very effective in improving the evaluation performance with LLM as evaluators, while maintaining low costs.

²The minimum hourly wage in the United States is near \$7.5, which can be found at <https://www.worker.gov/>. On average, annotating an example takes 3 minutes, and the Vicuna evaluation benchmark comprises 80 examples in total. Consequently, the cost per annotator amounts to \$30.

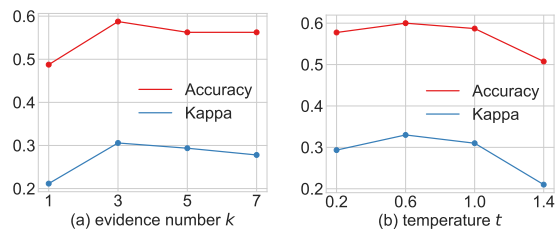


Figure 4: Variation of accuracy and kappa coefficient with a different number of evidence k and sampling temperature t when ChatGPT is used as the evaluator.

4 Analysis

4.1 Ablation on Evidence Number k and Temperature t

In the MEC and BPC strategy, we sample k evaluation results for each query prompt and ensemble them to enhance the evaluation process. We conduct an analysis to examine the influence of the number of evidence k , on the model’s evaluation performance. As illustrated in Figure 4(a), we compared the performance of ChatGPT with different values of k , namely 1, 3, 5, and 7. The model’s performance increases and then tends to be

constant or decreases slightly as k becomes larger. Despite the slight decrease, the enhancement of the model effect by the MCE strategy is still significant, illustrating the stability of the MEC strategy. Consequently, we found that a k value of 3 yields an optimal performance. With this value, the model achieves a notable level of performance while keeping the API cost relatively low. We further investigate the impact of sampling temperature t on evaluation performance. Figure 4(b) illustrates that both low temperature (i.e., 0.2) and high temperature (i.e., 1.4) result in sub-optimal evaluation alignment. We believe that low temperature eliminates the randomness of sampling, weakening the effect of MEC, while high temperature compromises the quality of generation results, leading to poor performance. Hence, it is crucial to select an appropriate temperature (e.g., 0.6 or 1.0 in our experiments) for the LLM evaluators.

4.2 Effectiveness of the BPDE

Our HITLC strategy utilizes BPDE score to select examples for human annotations. In order to analyze the efficiency of BPDE score, we compare BPDE with two typical baselines, *Random* and *Vanilla Diversity Entropy*, where *Random* denotes randomly select examples for human annotations, and *Vanilla Diversity Entropy* is calculated by using only the evaluation results of one position without swapping the position of two responses. To ensure fairness, the total number of evaluation results is 6 for both BPDE and *Vanilla Diversity Entropy*. As shown in Figure 5: **1)** Two Diversity Entropy methods outperform *Random*, showing the effectiveness of selecting examples based on the diversity entropy; **2)** BPDE outperforms *Vanilla DE*, which shows LLMs are sensitive to position exchange, and the results of BPC can significantly improve the performance of HITLC compared to relying solely on the results of MEC.

4.3 Generalization on the Pairwise Comparison Evaluation Template

To provide a more comprehensive validation of our proposed calibration methods, in addition to the previous SCORING evaluation template that rates each response, we extend our analysis to incorporate the COMPARING evaluation template. This template facilitates a direct comparison between two responses, eschewing explicit scores in its assessment. Specifically, we prompt LLMs to produce results labeled as “Assistant 1”, “Assistant 2”,

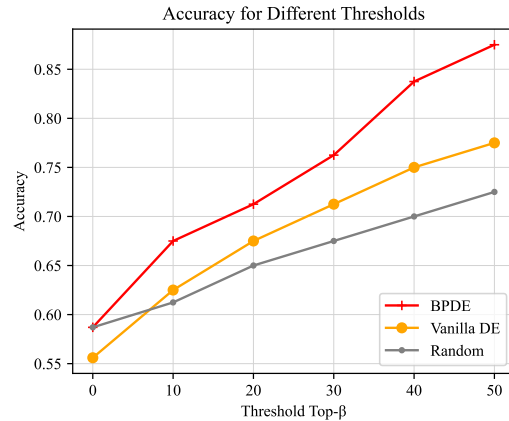


Figure 5: The accuracy of various methods changes with different human assistant thresholds (Top- β) when ChatGPT is used as the evaluator.

| TEMPLATES | METHODS | ACC. | KAP. | C.R |
|-----------|-----------|-------|------|-------|
| SCORING | VANILLA | 44.4% | 0.06 | 82.5% |
| SCORING | MEC | 53.2% | 0.24 | 35.0% |
| SCORING | MEC + BPC | 58.8% | 0.31 | N/A |
| COMPARING | VANILLA | 50.2% | 0.18 | 50.0% |
| COMPARING | MEC | 54.8% | 0.27 | 42.5% |
| COMPARING | MEC + BPC | 60.0% | 0.35 | N/A |

Table 5: Effectiveness of our proposed two automatic calibrated methods on two different evaluation templates with ChatGPT as the evaluator. ACC., KAP. and C.R are short for Accuracy, Kappa correlation coefficient, and Conflict Rate, respectively. N/A means the Conflict Rate is not valid for BPC methods.

or “Same”, indicating whether the response from Assistant 1 is better, worse, or equal to that of Assistant 2. As is shown in Table 5: **1)** Our proposed methods are applicable to both of these templates, leading to enhanced accuracy and a heightened correlation coefficient for ChatGPT; **2)** The significant performance gap (nearly 6% accuracy) between the VANILLA method of two templates, coupled with the high conflict rate, highlights the sensitivity and unreliability of LLMs. However, our methods effectively narrow this performance gap and reduce conflict, showcasing how calibration enhances LLM robustness.

4.4 Fine-Grained Analysis of Evaluation Quality

To further analyze the evaluation capabilities of the model, we perform a fine-grained analysis of the questions by dividing them into 9 categories following Zheng et al. (2023). We calculate the perfor-

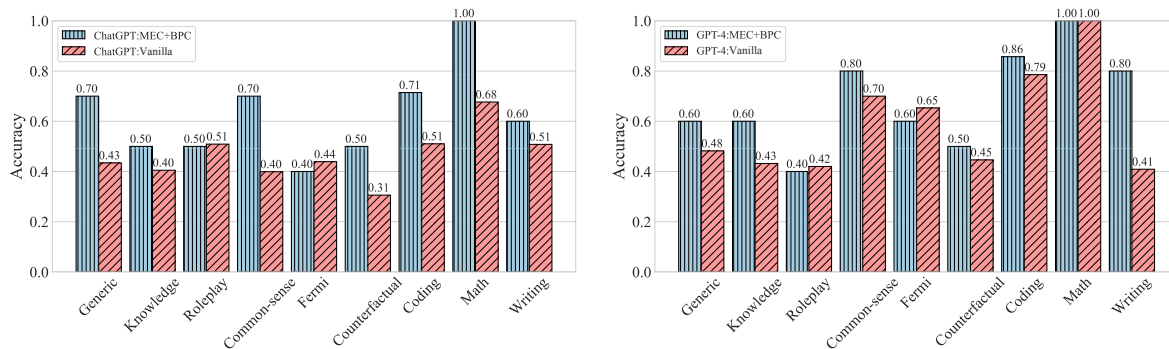


Figure 6: Fine-grained analysis of evaluation quality. Our MEC and BPC improve the evaluation performance of ChatGPT and GPT-4 in nearly all categories. Especially on the complex task categories such as common sense, coding, and math for ChatGPT.

mance of different evaluators. As shown in Figure 6, we find that: **1)** In certain complex tasks such as common sense, coding, and math, GPT-4 performs significantly better than ChatGPT, highlighting the strength of GPT-4 as a more fair evaluator in these scenarios; **2)** Our proposed MEC+BPC strategy demonstrates noticeable improvement in evaluating ChatGPT’s performance on complex tasks, allowing us to obtain satisfactory evaluation results with a low API cost.

5 Related Work

Large Language Models as Evaluators How to achieve reliable evaluations of generative AI assistants is one of the most important problems in AI research. Traditional metrics, like BLEU, ROUGE, BERTScore, or BARTScore fail to measure the alignment of the generated response with human intent (He et al., 2023). Considering the potent capabilities of LLMs, researchers have started utilizing LLMs to evaluate the proficiency of generative models in adhering to human instructions (Zheng et al., 2023; Lu et al., 2023; Li et al., 2023). There are two typical LLM-as-evaluator variations: point-wise evaluation and pair-wise evaluation. Point-wise evaluation scores the responses one by one, while pair-wise evaluation achieves the evaluation results by comparing two responses. Point-wise evaluation is more convenient, because para-wise evaluation may lack scalability when the number of responses increases. However, Pair-wise evaluation can give more accurate evaluation results than point-wise evaluation (Zheng et al., 2023). Specifically, assigning a precise score to a response is challenging as it requires providing evaluators with a detailed and accurate scoring standard that

is difficult to establish. In contrast, just making the comparison is much easier and more stable. In this work, we discover and address the positional bias of LLMs-as-evaluators for pair-wise evaluation. We believe it is essential for the whole community to achieve reliable evaluations.

Bias of Deep Neural Networks Deep Neural Networks have been proven to easily learn biases from the data, which significantly impacts their reliability. Specifically, bias has also been investigated in natural language inference (Gururangan et al., 2018; McCoy, Pavlick, and Linzen, 2019; Belinkov et al., 2019; Liu et al., 2020a,b), question answering (Min et al., 2019), ROC story cloze (Cai, Tu, and Gimpel, 2017; Schwartz et al., 2017), lexical inference (Levy et al., 2015), visual question answering (Goyal et al., 2017), information extraction (Wang et al., 2021, 2022; Song et al., 2023; Xia et al., 2023) and so on. LLMs are pre-trained using a vast amount of data from the internet, making it highly likely for them to learn biases present in those materials. Although the LLMs are already widely adopted as a proxy of human evaluators, the reliability of this paradigm is not well explored. In this paper, we critically examine the LLMs-as-evaluator paradigm and uncover a significant positional bias. Furthermore, we propose three simple yet effective methods to calibrate the positional bias to achieve reliable and fair evaluation results.

6 Conclusion

In this paper, we reveal a positional bias of the LLMs-as-evaluators paradigm, i.e., by manipulating the order of candidate responses during evaluation, the quality ranking results can be significantly

influenced. The positional bias seriously affects the reliability of the evaluation. We propose a calibration framework with three simple yet effective strategies to address the discovered bias. We also manually annotate the “win/tie/lose” outcomes of responses from ChatGPT and Vicuna-13B in the Vicuna Benchmark’s question prompt to evaluate the effectiveness of our framework. Extensive experiments demonstrate that our approach successfully alleviates evaluation bias, resulting in closer alignment with human judgments.

Limitation

In this paper, we identified the positional bias of LLM evaluators, which is essential for building reliable LLM evaluators. However, we did not investigate the underlying causes of this bias in LLMs. Future research should delve deeper into this issue.

Acknowledgements

This paper is supported by the National Key Research and Development Program of China 2020AAA0106700. The contact author is Zhifang Sui.

References

- Belinkov, Y.; Poliak, A.; Shieber, S.; Van Durme, B.; and Rush, A. 2019. Don’t Take the Premise for Granted: Mitigating Artifacts in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Bowman, S. R. 2023. Eight things to know about large language models. *arXiv preprint arXiv:2304.00612*.
- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Cai, Z.; Tu, L.; and Gimpel, K. 2017. Pay Attention to the Ending: Strong Neural Baselines for the ROC Story Cloze Task. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N. M.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B. C.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; García, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omernick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Díaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K. S.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2022. PaLM: Scaling Language Modeling with Pathways. *ArXiv*, abs/2204.02311.
- Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A Survey for In-context Learning. *arXiv preprint arXiv:2301.00234*.
- Dubois, Y.; Li, X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*.
- Gao, P.; Han, J.; Zhang, R.; Lin, Z.; Geng, S.; Zhou, A.; Zhang, W.; Lu, P.; He, C.; Yue, X.; Li, H.; and Qiao, Y. J. 2023. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model. *ArXiv*, abs/2304.15010.
- Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Gururangan, S.; Swayamdipta, S.; Levy, O.; Schwartz, R.; Bowman, S.; and Smith, N. A. 2018. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*.
- He, T.; Zhang, J.; Wang, T.; Kumar, S.; Cho, K.; Glass, J.; and Tsvetkov, Y. 2023. On the Blind Spots of Model-Based Evaluation Metrics for Text Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12067–12097. Toronto, Canada: Association for Computational Linguistics.
- Levy, O.; Remus, S.; Biemann, C.; and Dagan, I. 2015. Do Supervised Distributional Methods Really Learn Lexical Inference Relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*.
- Li, L.; Yin, Y.; Li, S.; Chen, L.; Wang, P.; Ren, S.; Li, M.; Yang, Y.; Xu, J.; Sun, X.; et al. 2023. M3IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning. *arXiv preprint arXiv:2306.04387*.

- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.
- Liu, T.; Xin, Z.; Chang, B.; and Sui, Z. 2020a. HypoNLI: Exploring the Artificial Patterns of Hypothesis-only Bias in Natural Language Inference. In *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association. ISBN 979-10-95546-34-4.
- Liu, T.; Xin, Z.; Ding, X.; Chang, B.; and Sui, Z. 2020b. An Empirical Study on Model-agnostic Debiasing Strategies for Robust Natural Language Inference. In *Proceedings of the 24th Conference on Computational Natural Language Learning*. Online: Association for Computational Linguistics.
- Lu, Q.; Qiu, B.; Ding, L.; Xie, L.; and Tao, D. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint arXiv:2303.13809*.
- McCoy, T.; Pavlick, E.; and Linzen, T. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- McHugh, M. L. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3): 276–282.
- Min, S.; Wallace, E.; Singh, S.; Gardner, M.; Hajishirzi, H.; and Zettlemoyer, L. 2019. Compositional Questions Do Not Necessitate Multi-hop Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- OpenAI. 2022. Introducing ChatGPT.
- OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.
- Peng, B.; Li, C.; He, P.; Galley, M.; and Gao, J. 2023. Instruction Tuning with GPT-4. *ArXiv*, abs/2304.03277.
- Schwartz, R.; Sap, M.; Konstas, I.; Zilles, L.; Choi, Y.; and Smith, N. A. 2017. The Effect of Different Writing Tasks on Linguistic Style: A Case Study of the ROC Story Cloze Task. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*.
- Song, Y.; Wang, P.; Zhu, D.; Liu, T.; Sui, Z.; and Li, S. 2023. RepCL: Exploring Effective Representation for Continual Text Classification. *arXiv preprint arXiv:2305.07289*.
- Sun, Z.; Shen, Y.; Zhou, Q.; Zhang, H.; Chen, Z.; Cox, D. D.; Yang, Y.; and Gan, C. 2023. Principle-Driven Self-Alignment of Language Models from Scratch with Minimal Human Supervision. *ArXiv*, abs/2305.03047.
- Turpin, M.; Michael, J.; Perez, E.; and Bowman, S. R. 2023. Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. *CoRR*, abs/2305.04388.
- Wang, P.; Song, Y.; Liu, T.; Lin, B.; Cao, Y.; Li, S.; and Sui, Z. 2022. Learning Robust Representations for Continual Relation Extraction via Adversarial Class Augmentation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6264–6278. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.
- Wang, P.; Xun, R.; Liu, T.; Dai, D.; Chang, B.; and Sui, Z. 2021. Behind the Scenes: An Exploration of Trigger Biases Problem in Few-Shot Event Classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1969–1978.
- Wang, Y.; Ivison, H.; Dasigi, P.; Hessel, J.; Khot, T.; Chandu, K. R.; Wadden, D.; MacMillan, K.; Smith, N. A.; Beltagy, I.; et al. 2023a. How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources. *arXiv preprint arXiv:2306.04751*.
- Wang, Y.; Yu, Z.; Zeng, Z.; Yang, L.; Wang, C.; Chen, H.; Jiang, C.; Xie, R.; Wang, J.; Xie, X.; et al. 2023b. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization. *arXiv preprint arXiv:2306.05087*.
- Xia, H.; Wang, P.; Liu, T.; Lin, B.; Cao, Y.; and Sui, Z. 2023. Enhancing Continual Relation Extraction via Classifier Decomposition. In *Findings of the Association for Computational Linguistics: ACL 2023*, 10053–10062. Toronto, Canada: Association for Computational Linguistics.
- Xu, C.; Guo, D.; Duan, N.; and McAuley, J. 2023. Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data. *ArXiv*, abs/2304.01196.
- Yuan, W.; Neubig, G.; and Liu, P. 2021. BARTScore: Evaluating Generated Text as Text Generation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y. N.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 27263–27277.
- Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv preprint arXiv:2306.05685*.

Zhou, C.; Liu, P.; Xu, P.; Iyer, S.; Sun, J.; Mao, Y.; Ma, X.; Efrat, A.; Yu, P.; Yu, L.; Zhang, S.; Ghosh, G.; Lewis, M.; Zettlemoyer, L.; and Levy, O. 2023. LIMA: Less Is More for Alignment.