

# Sample-Efficient Human Evaluation of Large Language Models via Maximum Discrepancy Competition

Kehua Feng<sup>1,2</sup>, Keyan Ding<sup>1,2</sup>, Kede Ma<sup>3</sup>, Zhihua Wang<sup>3</sup>, Qiang Zhang<sup>1,2</sup>, Huajun Chen<sup>1,2</sup>

<sup>1</sup>College of Computer Science and Technology, Zhejiang University

<sup>2</sup>ZJU-Hangzhou Global Scientific and Technological Innovation Center

<sup>3</sup>Department of Computer Science, City University of Hong Kong

kehufeng2002@gmail.com, {dingkeyan, qiang.zhang.cs, huajunsir}@zju.edu.cn  
kedema@cityu.edu.hk, zhihua.wang@my.cityu.edu.hk

## Abstract

The past years have witnessed a proliferation of large language models (LLMs). Yet, automated and unbiased evaluation of LLMs is challenging due to the inaccuracy of standard metrics in reflecting human preferences and the inefficiency in sampling informative and diverse test examples. While human evaluation remains the gold standard, it is expensive and time-consuming, especially when dealing with a large number of testing samples. To address this problem, we propose a sample-efficient human evaluation method based on MAximum Discrepancy (MAD) competition. MAD automatically selects a small set of informative and diverse instructions, each adapted to two LLMs, whose responses are subject to three-alternative forced choice by human subjects. The pairwise comparison results are then aggregated into a global ranking using the Elo rating system. We select eight representative LLMs and compare them in terms of four skills: knowledge understanding, mathematical reasoning, writing, and coding. Experimental results show that the proposed method achieves a reliable and sensible ranking of LLMs' capabilities, identifies their relative strengths and weaknesses, and offers valuable insights for further LLM advancement. Codes can be found at <https://github.com/weiji-Feng/MAD-Eval>.

## 1 Introduction

Since the inception of ChatGPT, there has been a surge in the development of large language models (LLMs) to achieve general-purpose language generation and understanding (Touvron et al., 2023; Bai et al., 2023; OpenAI, 2023; Jiang et al., 2023; Team et al., 2023) through self-supervised learning, supervised fine-tuning (Chiang et al., 2023; Xu et al., 2023; Wang et al., 2022) or reinforcement learning (Ouyang et al., 2022). Given the abundance of LLMs, it is crucial to establish a reliable and fair evaluation to fully showcase their strengths and weaknesses (Guo et al., 2023; Chang et al., 2023).

As a standard practice in machine learning, many LLM benchmarks such as MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023) and BIG-bench (Srivastava et al., 2022), have been constructed to provide a leaderboard for straightforward comparison of LLMs. However, due to the expensive (and nontrivial) human annotations, they only involve a limited set of tasks. Moreover, excessive repeated testing on these *fixed* benchmarks may lead to overfitting and give a false sense of improvement. Recent new benchmarks (Qin et al., 2024; Zheng et al., 2023; Duan et al., 2023) have shifted the task distribution to align with human needs. Nevertheless, there is still a high risk of overfitting and biased evaluation caused by data contamination or leakage (Schaeffer, 2023; Zhou et al., 2023b).

As current objective evaluation metrics provide a poor account for the human perception of language quality, researchers have begun using strong LLMs as judges. For example, Zhou et al. (2023a), Dubois et al. (2023) and Zheng et al. (2023) employed closed-source LLMs (e.g., GPT-4) through API calls for evaluation. Alternatively, open-source professional evaluators, e.g., PandaLM (Wang et al., 2023d), Shepherd (Wang et al., 2023c), and AUTO-J (Li et al., 2023b) have been trained. Although LLM-based judges can automatically evaluate text quality in an interpretable and human-like way and are suitable for different evaluation settings, they come with their own drawbacks. (i) Bias: LLM-based evaluators may exhibit preferences, favoring content in certain positions (position bias), longer responses (verbosity bias), or self-generated answers (self-enhancement bias). (ii) Unreliability: Due to their poor capabilities in certain areas (e.g. math reasoning or professional scientific knowledge understanding), LLM-based evaluators are unable to provide accurate and comprehensive evaluations that closely align with human agreements. These have hindered LLMs from being fair judges.

Human evaluation has always been considered the gold standard for evaluating LLMs. Chatbot Arena (Zheng et al., 2023) provides a crowdsourced platform for human users to engage in conversations with two LLM-based chatbots simultaneously and rate their responses based on preferences. Another benchmarking platform, Dynabench (Kiela et al., 2021), enables users to submit counterexamples to identify and address the limitations of models. However, human evaluation methods are subject to significant limitations: Evaluating large sample sets through human evaluators incurs huge costs and time consumption while evaluating small ones leads to pronounced sampling bias. This contradiction in human evaluation underscores the core problem driving this paper: *How to select the minimum number of informative samples that humans can use to accurately and fairly evaluate LLMs?*

In this study, we introduce a labor-saving evaluation approach by an *automated, adaptive* and *sample-efficient* mechanism to select testing samples. Our approach draws inspiration from previous works in the domains of software testing (McKeeeman, 1998) and computational vision (Wang and Simoncelli, 2008; Pei et al., 2017; Ma et al., 2018; Wang et al., 2020; Yan et al., 2021; Cao et al., 2021), guided by the principle of "Model Falsification as Model Comparison". We aim to automatically identify a minimum set of samples that are most likely to serve as counterexamples for falsifying an LLM, where higher difficulty in falsification indicates the superiority of the LLM.

To achieve this, we propose a novel evaluation approach based on Maximum Discrepancy (MAD) competition. Specifically, we start from a large-scale self-generated instruction pool as a finite approximation to the input space of a scenario-specific LLM evaluation. Based on the available human annotation budget, our approach automatically selects an optimal subset of samples from the instruction pool that maximally differentiates the performance of LLMs, while having the minimum intra-group similarity to ensure content diversity. These selected instructions, along with paired LLM responses, are subsequently subjected to human-centric assessment for pairwise comparison, followed by the utilization of the Elo rating system to transform the results into a global ranking of competing LLMs. Our evaluation method can be applied to a wide range of natural language processing tasks. We choose four real-world scenarios

as exemplars, including 1) scientific knowledge understanding, 2) math reasoning, 3) creative and functional writing, and 4) code generation and explanation, which assess the capabilities of LLMs at four primary levels: *understanding*, *reasoning*, *writing*, and *coding*, respectively. Given the constraints of a limited human budget, this study carefully selects eight representative LLMs as exemplars to demonstrate the efficacy of the proposed evaluation methodology.

The contributions of this paper are summarized as follows:

- We present a sample-efficient human evaluation approach by introducing an automated and adaptive mechanism to select minimal yet informative testing samples for LLM evaluation.
- Our approach fairly evaluates the capabilities of the advanced LLMs across multiple dimensions, providing a solid ranking of their relative performance.
- We identify some pros and cons of state-of-the-art LLMs in specific scenarios, offering valuable suggestions to further enhance their capabilities.

## 2 Related Works

Here we introduce the related works of LLM evaluation, including objective evaluation benchmarks, automated evaluation by LLM judges, and subjective evaluations by humans.

**Benchmarks as Evaluators** With the advancements in LLMs, a plethora of benchmarks have emerged for evaluating their performance comprehensively. Most existing benchmarks can be categorized as follows: 1) Core-knowledge benchmarks, e.g., MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023) and AGIEval (Zhong et al., 2023); 2) Instruction-following benchmarks, like LLMBAR (Zeng et al., 2023), Flan (Longpre et al., 2023) and NaturalInstructions (Mishra et al., 2021); 3) Conversational benchmarks, including CoQA (Reddy et al., 2019), MMDialog (Feng et al., 2022) and OpenAssistant (Köpf et al., 2023); 4) Human preference benchmarks, such as MT-Bench (Zheng et al., 2023); 5) Safety and risk benchmarks, including DecodingTrust (Wang et al., 2023a) and AdvGLUE (Wang et al., 2021). However, the automatic evaluation standards (e.g. BLEU(Papineni

et al., 2002), ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019)) have demonstrated a weak correlation with human agreements, especially when evaluating open-ended tasks or those requiring domain expertise (Novikova et al., 2017; Wei et al., 2024). Moreover, the extensive utilization of these benchmarks may result in biased evaluations due to data contamination and information leakage, while constructing a new benchmark with annotations often incurs exceptionally high costs.

**LLMs as Evaluators** Inspired by the strong instruction-following capabilities of advanced closed-source LLMs, many works (Zeng et al., 2023; Chan et al., 2023; Zhou et al., 2023a; Zheng et al., 2023; Dubois et al., 2023) employ them (e.g., GPT-4) as evaluators to assess the performance of other LLMs through a well-crafted prompt. To eliminate the dependence on proprietary LLMs, recent works have delved into exploring open-source LLM evaluators. PandaLM (Wang et al., 2023d) is trained on LLaMA (Touvron et al., 2023) with instruction tuning, conducting pairwise comparisons and providing comparative results along with a brief explanation. Shepard (Wang et al., 2023c) developed a model through meticulously processed data that can output critiques for question-answer pairs across diverse scenarios. AUTO-J (Li et al., 2023b) trained a model on user queries and LLM-generated responses under 58 real-world scenarios and diverse evaluation protocols to produce well-structured natural language critiques and corresponding ratings. However, despite its flexibility, LLM evaluators reveal potential issues, particularly concerning biases related to position and verbosity. Moreover, performance deficiencies of LLMs in professional or vertical domains can lead to unreliable evaluations (Zheng et al., 2023; Chen et al., 2024).

**Humans as Evaluators** Human evaluation is often considered the gold standard for natural language generation tasks, particularly in assessing whether LLM’s response meets human preferences. Chatbot Arena (Zheng et al., 2023; Chiang et al., 2024) develops a crowdsourcing platform to provide anonymous battles between LLM-based chatbots in real-world scenarios. Users can engage in pairwise comparisons with two chatbots simultaneously. By regularly collecting extensive user annotations, the Elo rating system can be used to calculate the relative performance level of LLMs, yield-

ing a convincing global ranking of LLMs. However, this approach necessitates a lengthy process to acquire human annotations, and employing crowdsourcing methods may introduce labeling noise. Dynabench (Kiela et al., 2021) is another platform used for dynamic data collection and model benchmarking, allowing users to submit counterexamples to identify models’ shortcomings. However, this process is still manual. It is acknowledged that annotating a large test set can be time-consuming and labor-intensive, while a small set may introduce significant sampling bias. To address this issue, this study optimizes the utilization of human evaluators by autonomously selecting a minimal set of samples that effectively distinguish the performance of LLMs.

### 3 Proposed Method

In this section, we present the proposed LLM evaluation approach based on maximum discrepancy competition. The pipeline of our method is shown in Figure 1. To provide a clear elucidation of our method, we formulate it within a mathematical framework as follows.

**Problem Formulation** Given a real-world scenario, we assume there is an instruction pool  $\mathcal{X}$  containing a large number of instances  $x \in \mathcal{X}$  that can approximate all of the cases within this scenario. We are provided with a set of LLMs denoted as  $\mathcal{F} = \{f_n\}_{n=1}^N$ , where each model takes an instruction  $x \in \mathcal{X}$  as input and generates a corresponding response represented as  $y_n = f_n(x)$ . Additionally, we consider a human evaluation environment  $\mathcal{H}$  wherein human subjects can reliably rate the quality of these responses. The ultimate objective is to compare the performance of  $N$  LLMs on the given instruction pool  $\mathcal{X}$  while adhering to strict constraints imposed by the limited availability of human annotation resources.

**MAD Competition of LLMs** To achieve this, we employ the well-established principle of Maximum Discrepancy (MAD) competition (Wang and Simoncelli, 2008) to falsify an LLM by identifying a minimum set of instructions that are most likely to serve as counterexamples. An LLM is considered superior when it possesses a higher level of resistance to falsification.

Let us consider a simple case in which we compare two LLMs  $f_i$  and  $f_j$  with the instruction pool  $\mathcal{X}$  and the human evaluation environment  $\mathcal{H}$ . Ac-

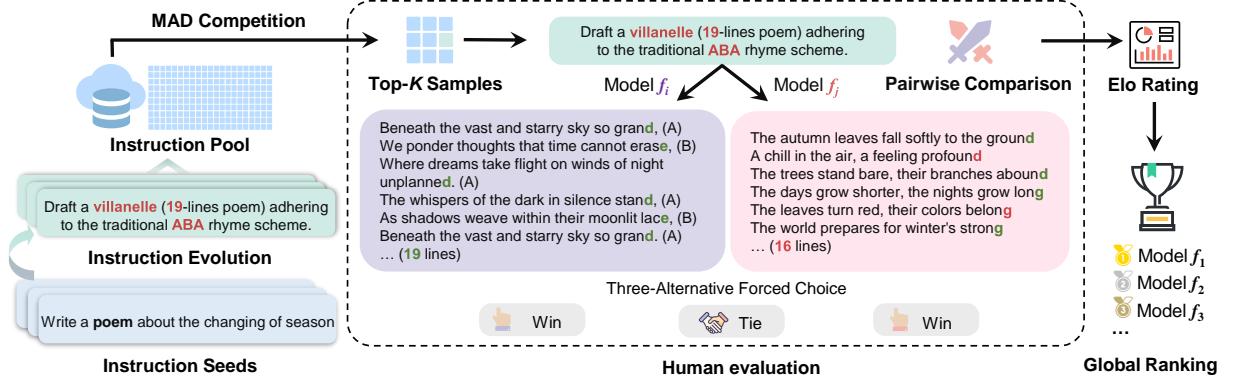


Figure 1: Illustration of our method. We first generate a large-scale instruction pool with seeds from a specific scenario by an instruction evolution method. Given any two competing LLMs, We then utilize the principle of MAD competition to automatically sample top- $K$  instructions (and corresponding responses) that can differentiate the performance of these LLMs, followed by the collection of human preferences indicating which response is better. Finally, we employ the Elo rating system to transform pairwise comparisons into a global ranking of LLMs.

cording to the principle of MAD competition, we select the instruction  $\hat{x} \in \mathcal{X}$  that optimally differentiates between  $f_i$  and  $f_j$  by

$$\hat{x} = \operatorname{argmin}_{x \in \mathcal{X}} \mathcal{M}(f_i(x), f_j(x)), \quad (1)$$

where  $\mathcal{M}(\cdot, \cdot)$  represents a metric that quantifies the semantic similarity between two sentences. The comparative analysis between  $f_i(\hat{x})$  and  $f_j(\hat{x})$  yields three plausible outcomes:

- $\mathcal{H}(f_i(\hat{x})) \gg \mathcal{H}(f_j(\hat{x}))$ , i.e., the winner is  $f_i$ , which means the majority of human evaluator prefer  $f_i(\hat{x})$  over  $f_j(\hat{x})$ .
- $\mathcal{H}(f_i(\hat{x})) \ll \mathcal{H}(f_j(\hat{x}))$ , i.e., the winner is  $f_j$ , which means the majority of human evaluator prefer  $f_j(\hat{x})$  over  $f_i(\hat{x})$ .
- $\mathcal{H}(f_i(\hat{x})) \approx \mathcal{H}(f_j(\hat{x}))$ , indicating a tie that human evaluators give them similar ratings.

In the first two results, our approach successfully finds a counterexample for a specific LLM, leading to a clear winner. The selected  $\hat{x}$  is the most informative in ranking the relative performance between  $f_i$  and  $f_j$ . In the third result, we encounter two types of ties: (i) Both  $f_i(\hat{x})$  and  $f_j(\hat{x})$  receive high ratings from human evaluators, indicating that both LLMs generate diverse yet satisfactory responses. This aligns with real-world scenarios where multiple plausible outputs are acceptable. The selected  $\hat{x}$  highlights the strengths exhibited by  $f_i$  and  $f_j$ , but contributes less to their relative performance ranking. (ii) Human evaluators assign low ratings to both  $f_i(\hat{x})$  and  $f_j(\hat{x})$ , suggesting that they fail in

distinct ways to follow the given instruction. The selected  $\hat{x}$  showcases the weaknesses of  $f_i$  and  $f_j$ , and provides less insight for relative ranking.

By optimizing Eq. (1), we can identify a small subset of instructions with the top- $K$  lowest similarities, i.e., the MAD instruction set  $\mathcal{I} = \{\hat{x}^{(k)}\}_{k=1}^K \subset \mathcal{X}$ . However, this naive implementation may lead to a homogeneous instruction set that exposes LLMs to the same type of failures, resulting in less diverse comparisons. To promote the identification of a wider array of failures between competing models, we impose a diversity constraint that encourages the selection of more varied instructions. This can be implemented by incorporating the metric  $\mathcal{M}$  to quantify the semantic similarity between an instruction candidate and the selected instructions. Thus, Eq. (1) is rewritten as

$$\hat{x}^{(k)} = \operatorname{argmin}_{x \in \mathcal{X} \setminus \mathcal{I}} \mathcal{M}(f_i(x), f_j(x)) + \lambda \mathcal{M}(x, \mathcal{I}), \quad (2)$$

where  $\mathcal{I} = \{\hat{x}^{(k)}\}_{k=1}^{K-1}$  represents the collection of  $K - 1$  instructions that have been previously identified based on Eq. (2). The parameter  $\lambda$  controls the balance between the two terms. Once an instruction  $\hat{x}^{(k)}$  has been identified, it will be incorporated into the set  $\mathcal{I}$ .

Given a set of  $N$  LLMs, we extend the pairwise MAD competition by selecting top- $K$  instructions for each competing LLM pair ( $\binom{N}{2}$  pairs in total). Finally, we generate a MAD response set denoted as  $\mathcal{R}$  containing  $N(N - 1)K$  response pairs. Importantly, it should be noted that the size of  $\mathcal{R}$  is independent of the scale of instruction pool  $\mathcal{X}$ . Therefore, it is advisable to encompass a wide range of

---

**Algorithm 1:** Proposed LLM Evaluation Method

---

**Input:** An instruction set  $\mathcal{X}$ , a set of competing LLMs  $\mathcal{F} = \{f_n\}_{n=1}^N$ , and a semantic similarity metric  $\mathcal{M}$

**Output:** Global ranking scores  $\mathbf{s} \in \mathbb{R}^N$  of LLMs

```

1    $\mathcal{R} \leftarrow \emptyset$ 
2   for  $n \leftarrow 1$  to  $N$  do
3       | Generate the responses  $\{f_n(x) | x \in \mathcal{X}\}$ 
4   end
5   for  $i \leftarrow 1$  to  $N - 1$  do
6       | for  $j \leftarrow i + 1$  to  $N$  do
7           |   |  $\mathcal{I} \leftarrow \emptyset$ 
8           |   | for  $k \leftarrow 1$  to  $K$  do
9           |   |   | Select  $\hat{x}^{(k)} \in \mathcal{X} \setminus \mathcal{I}$  by optimizing Eq. (2)
10          |   |   |  $\mathcal{I} \leftarrow \mathcal{I} \cup \hat{x}^{(k)}$ 
11          |   |   |  $\mathcal{R} \leftarrow \mathcal{R} \cup \{f_i(\hat{x}^{(k)}), f_j(\hat{x}^{(k)})\}$ 
12      |   | end
13  | end
14 end
15 Collect comparison results on  $\mathcal{R}$  via 3-AFC in Eq. (3)
16 Compute  $\mathbf{s}$  via Elo rating system in Eq. (4)

```

---

instructions, assuming that the computational cost of LLM response is acceptable, to effectively capture diverse instances where competing LLMs may fail.

**Ranking of LLMs** Given the human evaluation environment  $\mathcal{H}$ , we conduct pairwise comparisons on the MAD response set  $\mathcal{R}$  to gather human judgments on which response  $\{f_i(\hat{x}^{(k)}), f_j(\hat{x}^{(k)})\}$  performs better according to the instruction  $\hat{x}^{(k)} \in \mathcal{I}$ . In this study, we employ the three-alternative forced choice (3-AFC) method, wherein participants are presented with  $f_i(\hat{x}^{(k)})$  and  $f_j(\hat{x}^{(k)})$  (along with  $\hat{x}^{(k)}$ ), and required to indicate which response is of higher quality or declare a tie. The result  $w$  is recorded as:

$$w = \begin{cases} 1, & \text{if winner} = f_i, \\ 0, & \text{if winner} = f_j, \\ 0.5, & \text{if tie.} \end{cases} \quad (3)$$

We then collect the human results and adopt the Elo rating system (Elo and Sloan, 2008) to infer the global ranking of  $\mathcal{F}$ . Specifically, let  $\mathbf{s}^{(0)} = \{s_n^{(0)}\}_{n=1}^N$  be the initial ranking scores of  $N$  LLMs. For the  $t$ -th comparison,  $s_i^{(t)}$  and  $s_j^{(t)}$  are updated by

$$\begin{aligned} s_i^{(t)} &= s_i^{(t-1)} + \eta \times \left( w^{(t)} - \frac{1}{1 + 10^{(s_j^{(t-1)} - s_i^{(t-1)})/\tau}} \right), \\ s_j^{(t)} &= s_j^{(t-1)} + \eta \times \left( 1 - w^{(t)} - \frac{1}{1 + 10^{(s_i^{(t-1)} - s_j^{(t-1)})/\tau}} \right), \end{aligned} \quad (4)$$

where  $\eta$  and  $\tau$  are scaling parameters. To mitigate the sensitivity of the linear update method to comparison order, following Chatbot Arena (Zheng et al., 2023), we employ the bootstrap technique to yield a more stable result (see Appendix A.1 for details). We summarize the procedure of proposed LLM evaluation method in Algorithm 1.

Finally, it is simple and cost-effective to incorporate a new LLM (i.e.,  $f_{N+1}$ ) into the MAD competition. No modifications are required for the current sampled instruction set  $\mathcal{I}$  and the MAD response set  $\mathcal{R}$  with the associated human preferences. The additional work involves sampling new  $N \times K$  instructions (and corresponding responses) that maximize the difference between  $f_{N+1}$  and  $\mathcal{F} = \{f_n\}_{n=1}^N$ , collecting human preferences, and updating the global ranking scores using Eq. (4). The procedure of ranking a new LLM is summarized in Algorithm 2 (see Appendix A.2).

## 4 Experiments

In this section, we apply the proposed method to evaluate eight advanced LLMs across four scenarios, and present a comparative analysis on performance ranking.

### 4.1 Experimental Setups

**Construction of Instruction Pool  $\mathcal{X}$**  To ensure the fairness of comparisons and avoid potential data leakage issues, we first develop a large-scale dataset with diverse instructions across various scenarios, namely the instruction pool  $\mathcal{X}$ . This is achieved by the following steps: 1) determining evaluation scenarios, 2) collecting instruction seed data from various benchmark datasets of these scenarios, and 3) generating more instructions that mimic real-world human-chatbot interactions through the instruction evolution method. In specific, we assess the capabilities of LLMs at four levels: *understanding*, *reasoning*, *writing*, and *coding*, by designing four scenarios: 1) scientific knowledge understanding, 2) math reasoning, 3) creative and functional writing, and 4) code generation and explanation. For each scenario, we collect instruction seeds from multiple benchmark datasets, and employ the instruction evolution method (Xu et al., 2023) to generate new massive instructions that are aligned with real-world scenarios and avoid potential data leakage (see Appendix B.1 for more details). In our work, we sampled 3K instruction seeds per scenario and iteratively performed 10

Table 1: The global ranking results of eight LLMs on four scenarios

Model	Overall		Understanding		Reasoning		Writing		Coding	
	Rank	Elo Rating	Rank	Elo Rating	Rank	Elo Rating	Rank	Elo Rating	Rank	Elo Rating
GPT-4-Turbo	1	1132	2	1065	1	1123	1	1162	1	1103
Gemini-Pro	2	1107	1	1091	2	1094	2	1097	3	1085
OpenChat-3.5	3	1035	3	1047	3	1087	3	1025	4	971
GPT-3.5-Turbo	4	1034	4	988	4	1069	5	976	2	1095
WizardLM-13B	5	937	5	986	8	823	4	1001	6	961
QWen-14B-Chat	6	932	6	967	6	939	7	918	5	963
ChatGLM3-6B	7	929	8	924	5	998	8	861	7	958
Vicuna-13B	8	894	7	932	7	869	6	962	8	865

evolutions, ultimately obtaining 30K evolved instructions for each scenario.

**Selection of Completing LLMs  $\mathcal{F}$**  Considering the constraints imposed by limited human annotation budget, in this experiment, we select eight widely-recognized LLMs as exemplars to demonstrate the effectiveness of our evaluation method, including GPT-3.5-Turbo (Ouyang et al., 2022), GPT-4-Turbo (OpenAI, 2023), Gemini-Pro (Team et al., 2023), WizardLM-13B (Xu et al., 2023), Vicuna-13B (Chiang et al., 2023), OpenChat-3.5 (Wang et al., 2023b), Qwen-14B-Chat (Bai et al., 2023), and ChatGLM3-6B (Du et al., 2021), among which the first three are proprietary models, while subsequent three are high-capacity open-source English models, and the remaining two are cutting-edge open-source Chinese models (see Appendix B.2 for their implementations).

**Selection of Similarity Metric  $\mathcal{M}$**  To assess the semantic similarity between two sentences, a popular approach involves comparing their text embedding vectors by calculating the cosine similarity of embeddings, such as BERTScore (Zhang et al., 2019). An alternative method is prompting powerful LLMs (like GPT-4) to generate the similarity score. In our experiment, we adopt the former strategy where the embedding model is *text-embedding-ada-002* released by OpenAI. Consequently, the similarity metric  $\mathcal{M} = \cos(\mathbf{h}_A, \mathbf{h}_B)$ , where  $\mathbf{h}_A$  and  $\mathbf{h}_B$  are the embedding vectors of two sentences  $A$  and  $B$ , respectively. For 30K instructions in each scenario, a total of  $\binom{8}{2} \times 30K = 840K$  pairwise comparisons need to be conducted.

**Collection of Human Preferences** We conduct user studies to collect human preference results from the pairwise responses in  $\mathcal{R}$ . Following the 3-AFC method, participants are instructed to choose the response that exhibits superior quality in rela-

tion to the given instruction. For each scenario, we set  $K = 10$  for each competing LLM pair in our experiments, resulting in a total of  $\binom{8}{2} \times 10 = 280$  paired comparisons, which is independent of the scale of  $\mathcal{X}$ . We gather data from a group of 13 volunteer postgraduates with a strong background in computer science and engineering. More details of human evaluators are shown in Appendix C.

## 4.2 Ranking Results

We provide our ranking results (overall and respective scenarios) in Table 1, which unveils several interesting phenomena.

For **scientific knowledge understanding**, the main observation is that proprietary LLMs (i.e., GPT-4-Turbo, GPT-3.5-Turbo, and Gemini-Pro) often outperform open-source LLMs due to their precise and comprehensive understanding of scientific knowledge, such as the correct application of theorems and properties. Interestingly, OpenChat-3.5, with only 7 billion parameters, tends to provide fine-grained explanations while encompassing the required core knowledge, garnering preference from the majority of human subjects. This ultimately results in a higher ranking compared to the proprietary model GPT-3.5-Turbo with relatively concise responses. On the other hand, other open-source models like Vicuna-13B, although delivering longer responses, tend to provide redundant explanations.

For **math reasoning**, our ranking result exhibits a high consistency with the leaderboard of GSM8K dataset (Cobbe et al., 2021), as the instruction pool is derived from it. Through analyzing the response pairs selected by MAD competition, we observed two kinds of differences between them: 1) different reasoning paths and 2) different computational results during the similar reasoning process. In the first case, since the generated instructions primarily focus on question types and difficulty encountered

Table 2: Some strengths and weaknesses of LLMs discovered by our experiments

Model	Strengths	Weaknesses
GPT-4-Turbo	<b>Writing:</b> powerful writing capability. <b>Reasoning:</b> clear reasoning logic. <b>Coding:</b> high success rate in code execution.	<b>Others:</b> laziness, slight deviation in instruction comprehension.
Gemini-Pro & OpenChat-3.5	<b>Writing:</b> strong writing capability. <b>Reasoning:</b> clear reasoning logic.	<b>Reasoning:</b> limited ability in complex arithmetic computations. <b>Coding:</b> insufficient accuracy in writing complex code.
GPT-3.5-Turbo	<b>Coding:</b> strong coding proficiency.	<b>Understanding:</b> short reply length, lack detailed analysis. <b>Others:</b> laziness
WizardLM-13B	<b>Writing:</b> relatively strong writing capability.	<b>Reasoning:</b> unclear reasoning logic, weak arithmetic ability. <b>Coding:</b> limited coding proficiency.
QWen-14B	<b>Coding:</b> relatively high success rate in execution.	<b>Reasoning:</b> limited ability in complex arithmetic computations. <b>Writing:</b> short reply length, simple content. <b>Others:</b> (somewhat) laziness.
ChatGLM3-6B	<b>Reasoning:</b> relatively rigorous logic, relatively accurate arithmetic operation.	<b>Understanding:</b> limited knowledge reserve, explanation lacks details. <b>Reasoning:</b> limited ability in complex arithmetic computations. <b>Writing:</b> short reply length, simple content. <b>Coding:</b> low success rate in code execution, disregard for instruction requirements. <b>Others:</b> laziness.
Vicuna-13B	<b>Writing:</b> exceptional instruction-following capability.	<b>Understanding:</b> lack of expertise in explanation, moderate knowledge reserve. <b>Reasoning:</b> limited computational and reasoning capabilities. <b>Coding:</b> low success rate in code execution, disregard for instruction requirements.

in grade school, their responses lead to singular and concise reasoning paths. As a result, divergent outcomes typically arise from significant differences in these reasoning paths. In the second case, variations in model arithmetic capabilities may lead to different results for the same reasoning steps, causing outcomes to collapse due to errors in intermediate steps. Among all evaluated models, WizardLM-13B exhibited comparatively inferior performance. We attribute this to the fact that WizardLM adopts an instruction evolution method to generate training data; however, the seed instructions come from the open-ended instruction set Alpaca 52K (Taori et al., 2023), which is not specifically optimized for math reasoning. Vicuna-13B faces a similar issue.

For **creative and functional writing**, we observed that the majority of instructions selected by MAD competition are unconstrained, open-ended prompts for free-form writing, such as composing a short story or crafting a holiday recipe. Consequently, human subjects prefer LLMs with extended response lengths, containing rich details,

tend to outperform those with shorter responses. A direct illustration is that ChatGLM3-6B produces an average response of 221.2 words, while GPT-4-Turbo generates an average response of 454.8 words. Furthermore, responses with abundant details often embody profound insights, making them more preferred by humans.

For **code generation and explanation**, human subjects assess not only the correctness of code, but also their capability of instruction-following (e.g., adherence to specified line limits, usage of designated Python libraries, conformity to application scenarios, etc.). We observed that LLMs tend to exhibit greater disparities in code generation tasks compared to code explanation. Our results align more closely with code benchmarks like HumanEval (Chen et al., 2021). For instance, the models GPT-4-Turbo (76.83), GPT-3.5-Turbo (74.39) and Gemini Pro (59.76) have high accuracy on HumanEval according to Akter et al. (2023), also garnered top preferences by human subjects.

In summary, Table 2 provides an overview of

the strengths and weaknesses of each LLM in specific scenarios, from which we can obtain valuable insights to enhance their response quality. The section of **case study** in Appendix D.2 shows several cases that validate our findings, especially the counterexamples of the SOTA model GPT-4-Turbo. Furthermore, the counterexamples identified through the MAD competition and validated by human judgments can facilitate the training of more robust LLMs, e.g., using an active learning framework with adversarial samples (Sinha et al., 2019).

### 4.3 Comparison with Existing Leaderboards

We compare our overall ranking results with existing LLM evaluation methods, including 1) human evaluator-based *Chatbot Arena leaderboard*, 2) LLM judge-based *AlpacaEval 2.0*, and 3) standard metric-based *OpenCompass 2.0*. Table 3 reports the ranking results of these methods. It is expected that our ranking closely resembles those obtained by Chatbot Arena due to the human-centric evaluation, although slight discrepancies in the ranking of Vicuna-13B (maybe caused by differing proportions of scenario data). It is noteworthy that the result of Chatbot Arena relies on massive battles of LLMs annotated by crowds, whereas our method is based on an automated mechanism to pick a few informative samples for human annotation, saving both labor and time.

In AlpacaEval 2.0, we noticed that WizardLM-13B ranked higher than Openchat-3.5 and GPT-3.5-Turbo, which is contrary to our ranking result. This discrepancy may stem from AlpacaEval’s emphasis on assessing LLMs’ instruction-following capability in open-ended tasks, whereas WizardLM-13B has been fine-tuned by 520K instructions covering a wide range of scenarios and difficulties.

When comparing the rankings obtained from OpenCompass 2.0, disparities are evident in Qwen-14B-Chat and ChatGLM3-6B models. This can be attributed to their specialized training tailored to a wide range of benchmarks (e.g., HumanEval, MMLU and GSM8K), enabling them to excel in comprehensive benchmark leaderboards. It should be noted that different evaluation methods may prioritize distinct aspects and scopes, resulting in various leaderboards.

### 4.4 Comparison with Different Sampling Algorithms

We compare the MAD competition sampling algorithm with three additional baselines, including the

Table 3: Comparison with existing leaderboard results (recorded on Feb 7, 2024)

Model	Ranking			
	Ours	Chatbot-Arena	Alpaca-Eval 2.0	OpenCompass 2.0
GPT-4-Turbo	1	1	1	1
Gemini-Pro	2	2	2	-
OpenChat-3.5	3	3	4	-
GPT-3.5-Turbo	4	4	5	2
WizardLM-13B	5	5	3	5
QWen-14B-Chat	6	7	6	3
ChatGLM3-6B	7	8	8	4
Vicuna-13B	8	6	7	6

1) KL divergence and 2) cross-entropy-based sampling algorithms proposed in Boubdir et al. (2023), as well as 3) random sampling. However, due to the reliance on token log probabilities, KL divergence and cross-entropy sampling algorithms are not suitable for certain API-based models (e.g., Gemini-Pro). Therefore, we selected seven LLMs that meet the requirements of all sampling algorithms and evaluated them in the *reasoning* scenario. The specific model names and final ranking results are shown in Table 4. It is worth noting that our prompt pool evolved from the GSM8K dataset, and we instructed human annotators to prioritize inference accuracy as the most important evaluation criterion. Thus, we consider the model’s accuracy on the GSM8K test set as the “golden” ranking. Notably, the MAD competition strategy closely approximates the golden ranking result, using just 10 optimally selected samples compared to 8K in the original test set. However, there are some inconsistencies in the rankings obtained by KL divergence, cross-entropy, or random strategies. For example, in the KL divergence ranking, GPT-4-Turbo is no longer the best model, and in the cross-entropy ranking, ChatGLM3-6B even surpasses GPT-4-Turbo and GPT-3.5-Turbo.

In addition, we qualitatively analyzed the differences among the four strategies in instruction selection. Table 12 in the Appendix presents the specific content of the Top-10 instructions chosen by different strategies in the *writing* scenario. We found that the instructions selected by the KL divergence strategy are almost all related to poetry (9 out of 10), while the cross-entropy strategy favors tasks such as writing papers and stories. The random strategy often introduces varying numbers of overlapping task types. In our work, the MAD competition strategy prioritizes instruction diver-

Table 4: Comparison of ranking results using different sampling algorithms in the *reasoning* scenario

Model	MAD Competition <b>(Ours)</b>		KL Divergence		Cross-Entropy		Random		"Golden" ranking (GSM8K)	
	Rank	Elo Rating	Rank	Elo Rating	Rank	Elo Rating	Rank	Elo Rating	Rank	Accuracy
GPT-4-Turbo	1	1157	2	1020	4	983	1	1028	1	92.7
OpenChat-3.5	2	1132	4	1005	2	1030	5	1000	2	77.3
GPT-3.5-Turbo	3	1079	1	1036	3	1025	2	1025	3	74.9
ChatGLM3-6B	4	1018	3	1017	1	1045	3	1007	4	72.3
QWen-14B-Chat	5	953	5	993	5	982	4	1007	5	60.1
Vicuna-13B	6	858	7	957	7	974	7	947	7	11.3
WizardLM-13B	7	802	6	972	6	974	6	987	6	13.5

Table 5: The ranking results of *writing* scenario under different similarity measures  $\mathcal{M}$

Model	Elo Rating		
	Ada-002	BERTScore	GPT-4
GPT-4-Turbo	1 (1162)	<b>2 (1060)</b>	1 (1084)
Gemini-Pro	2 (1097)	<b>1 (1061)</b>	2 (1040)
OpenChat-3.5	3 (1025)	3 (1020)	3 (1010)
WizardLM-13B	4 (1001)	4 (990)	<b>5 (997)</b>
GPT-3.5-Turbo	5 (976)	5 (989)	<b>4 (998)</b>
Vicuna-13B	6 (962)	6 (982)	6 (995)
QWen-14B-Chat	7 (918)	7 (951)	7 (974)
ChatGLM3-6B	8 (861)	8 (946)	8 (902)

sity, thereby minimizing the occurrence of repeated task types in the selected instructions as much as possible, facilitating the exposure of failures of the LLM across a wider range of tasks.

#### 4.5 Ablation Studies

**Sensitivity of  $\mathcal{M}$**  We first examine the sensitivity of the obtained results to different semantic similarity measurements  $\mathcal{M}$  in Eq. (2). Besides the used *text-embedding-ada-002* similarity measurement, we employ another two popular metrics, BERTScore (Zhang et al., 2019) and GPT-4 judgment, to quantify sentence similarities. BERTScore uses text representation from the pre-trained BERT to calculate the similarity between the two responses, and GPT-4 judgment assesses the similarity via prompt engineering. Taking the *writing* scenario as an example, we identify a nearly consistent global ranking across the three metrics, as shown in Table 5. This can be attributed to the fact that MAD competition selects instructions that effectively differentiate between two models with the maximum discrepancy, which can be well approximated by all three metrics.

**Sensitivity of  $K$**  We then examine the robustness of our results to the number of sampled instructions

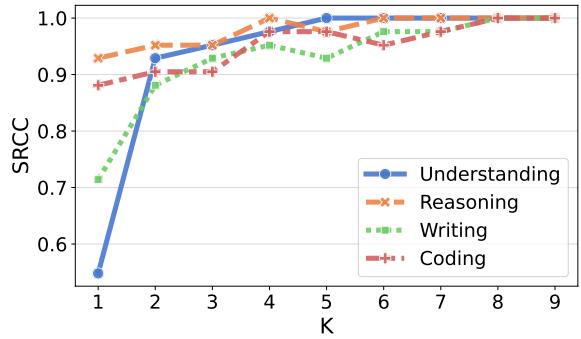


Figure 2: The SRCC between the default top-10 and other top- $K$  rankings, where  $K \in \{1, 2, \dots, 9\}$ , and SRCC = 1 indicates an identical ranking.

$K$  for human judgments. Figure 2 shows the Spearman Rank Correlation Coefficient (SRCC) between the top-10 (default) ranking and other top- $K$  rankings, where  $K = \{1, 2, \dots, 9\}$ . The ranking results exhibit a high level of stability  $SRCC > 0.95$  ( $SRCC = 1$ ) when  $K > 5$  ( $K \geq 8$ ) across all scenarios, providing compelling evidence for the sample efficiency of the proposed method. It is worth mentioning that the value of  $K$  can be adjusted flexibly and may vary for each competing LLM pair. For example, if we know that the performance of two LLMs is relatively similar, we can increase  $K$  to conduct more comparisons and obtain a more reliable ranking. Conversely, if there is a significant disparity in performance between these two LLMs, we can decrease  $K$  (even setting it to zero) to minimize human costs. This practice is analogous to Chatbot Arena where the number of battles between two LLMs is not fixed.

#### 4.6 Significance of Diversity Measurement

We also explore the significance of diversity measurement in Eq. (2). Table 11 in the Appendix presents the Top-10 instructions selected by the MAD competition before and after considering diversity in the *writing* scenario. Without consider-

ing diversity, we found that four instructions were related to poetry, resulting in a relatively homogeneous selection of scenarios and tasks. However, after considering diversity, each instruction represents almost unique tasks and scenarios. We believe that diversity in scenarios is beneficial for exploring more types of model failures, thereby providing more instructive insights for evaluation.

## 5 Conclusion

This paper presents a sample-efficient human evaluation approach for ranking LLMs by the methodology of maximum discrepancy competition. We emphasize that, instead of manually curating a fixed test set with human annotations, our method automatically samples a small set of informative and unbiased instructions to differentiate the performance of LLMs, thereby reducing human labor significantly. Moreover, the counterexample data collected via the MAD competition not only serves as a means to evaluate cutting-edge LLMs, but also facilitates the training of even more robust models (e.g., adversarial training). Furthermore, the proposed method has broad applicability beyond general LLMs. It can also be extended to multimodal LLMs where the input includes multiple types of data, such as text, images, audio, and video. In this extension, we need to consider the other modalities when constructing the instruction pool, while minimal changes are required for other procedures. In the future, we will strive to increase the number of LLMs and expand the scope of evaluation by incorporating more scenarios, ultimately creating a comprehensive leaderboard that will be made available to the public.

## 6 Limitations

Our evaluation method involves human evaluators for conducting pairwise comparisons. When faced with dozens or hundreds of competing LLMs and the need to obtain the relative ranking of a new LLM, our method still requires substantial human resources to finish these comparisons. To alleviate this problem, we can reduce the number of sampled instructions, as described in the ablation study (Section 4.5). Additionally, we can employ a coarse-to-fine mechanism (i.e., a collaborative LLM-human evaluation method). That is, we first utilize a strong LLM as an evaluator to replace human judgments, leading to a preliminary ranking. We then select a few LLMs that are ranked near the

targeted LLM, and apply the proposed method for fine ranking within this LLM subset.

## References

- Syeda Nahida Akter, Zichun Yu, Aashiq Muhammed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. An In-depth Look at Gemini’s Language Abilities. *arXiv:2312.11444*, 2023.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program Synthesis with Large Language Models. *arXiv:2108.07732*, 2021.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen Technical Report. *arXiv:2309.16609*, 2023.
- Meriem Boubdir, Edward Kim, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. Which prompts make the difference? data prioritization for efficient human llm evaluation. *arXiv preprint arXiv:2310.14424*, 2023.
- Peibei Cao, Zhangyang Wang, and Kede Ma. Debiased Subjective Assessment of Real-World Image Enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 711–721, 2021.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. *arXiv:2308.07201*, 2023.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 2023.
- Sahil Chaudhary. Code Alpaca: An Instruction-following LLaMA model for code generation. GitHub repository <https://github.com/sahil280114/codealpaca>, 2023.
- Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or llms as the judge? a study on judgement biases. *arXiv:2402.10669*, 2024.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating Large Language Models Trained on Code. *arXiv:2107.03374*, 2021.
- Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. INSTRUCTEVAL: Towards Holistic Evaluation of Instruction-Tuned Large Language Models. *arXiv:2306.04757*, 2023.

- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%\* Chatgpt Quality. Blog post <https://lmsys.org/blog/2023-03-30-vicuna/>, 2023.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv:2403.04132*, 2024.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training Verifiers to Solve Math Word Problems. *arXiv:2110.14168*, 2021.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General Language Model Pretraining with Autoregressive Blank Infilling. *arXiv:2103.10360*, 2021.
- Haodong Duan, Jueqi Wei, Chonghua Wang, Hongwei Liu, Yixiao Fang, Songyang Zhang, Dahua Lin, and Kai Chen. BotChat: Evaluating LLMs' Capabilities of Having Multi-Turn Dialogues. *arXiv:2310.13650*, 2023.
- Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback. *arXiv:2305.14387*, 2023.
- Arpad E. Elo and Sam Sloan. *The Rating of Chessplayers: Past and Present*. Ishi Press International, 2008.
- Jiazhan Feng, Qingfeng Sun, Can Xu, Pu Zhao, Yaming Yang, Chongyang Tao, Dongyan Zhao, and Qingwei Lin. MMDialog: A Large-scale Multi-turn Dialogue Dataset Towards Multi-modal Open-domain Conversation. *arXiv:2211.05719*, 2022.
- Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. Evaluating Large Language Models: A Comprehensive Survey. *arXiv:2310.19736*, 2023.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *arXiv:2009.03300*, 2020.
- Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junting Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. C-Eval: A Multi-Level Multi-Discipline Chinese Evaluation Suite for Foundation Models. *arXiv:2305.08322*, 2023.
- Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. CodeSearchNet Challenge: Evaluating the State of Semantic Code Search. *arXiv:1909.09436*, 2019.
- Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv:2310.06825*, 2023.
- Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringolia, et al. Dynabench: Rethinking Benchmarking in NLP. *arXiv:2104.14337*, 2021.
- Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. OpenAssistant Conversations – Democratizing Large Language Model Alignment. *arXiv:2304.07327*, 2023.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with Pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, page 611–626. Association for Computing Machinery, 2023.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. *arXiv:2303.17760*, 2023a.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative Judge for Evaluating Alignment. *arXiv:2310.05470*, 2023b.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An Automatic Evaluator of Instruction-following Models. GitHub repository [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval), 2023c.
- Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.
- Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The Flan Collection: Designing Data and Methods for Effective Instruction Tuning. *arXiv:2301.13688*, 2023.
- Kede Ma, Zhengfang Duanmu, Zhou Wang, Qingbo Wu, Wentao Liu, Hongwei Yong, Hongliang Li, and Lei Zhang. Group Maximum Differentiation Competition: Model Comparison with Few Samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 851–864, 2018.

- William M. McKeeman. Differential Testing for Software. *Digital Technical Journal*, pages 100–107, 1998.
- Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. *arXiv:2104.08773*, 2021.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, 2017.
- OpenAI. GPT-4 Technical Report. *arXiv:2303.08774*, 2023.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744, 2022.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002.
- Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In *Symposium on Operating Systems Principles*, pages 1–18, 2017.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. InFoBench: Evaluating Instruction Following Ability in Large Language Models. *arXiv:2401.03601*, 2024.
- Siva Reddy, Danqi Chen, and Christopher D Manning. CoQA: A Conversational Question Answering Challenge. *Transactions of the Association for Computational Linguistics*, pages 249–266, 2019.
- Rylan Schaeffer. Pretraining on the Test Set Is All You Need. *arXiv:2309.08632*, 2023.
- Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational Adversarial Active Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *arXiv:2206.04615*, 2022.
- Rohan Taori, Ishaaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An Instruction-following LLaMA Model. GitHub repository [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: A Family of Highly Capable Multimodal Models. *arXiv:2312.11805*, 2023.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*, 2023.
- Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial GLUE: A Multi-Task Benchmark for Robustness Evaluation of Language Models. *arXiv:2111.02840*, 2021.
- Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. DecodingTrust: A Comprehensive Assessment of Trustworthiness in GPT Models. *arXiv:2306.11698*, 2023a.
- Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. OpenChat: Advancing Open-source Language Models with Mixed-Quality Data. *arXiv:2309.11235*, 2023b.
- Haotao Wang, Tianlong Chen, Zhangyang Wang, and Kede Ma. I Am Going MAD: Maximum Discrepancy Competition for Comparing Classifiers Adaptively. *arXiv:2002.10648*, 2020.
- Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O’Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Shepherd: A Critic for Language Model Generation. *arXiv:2308.04592*, 2023c.
- Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. PandaLM: An Automatic Evaluation Benchmark for LLM Instruction Tuning Optimization. *arXiv:2306.05087*, 2023d.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-Instruct: Aligning Language Models with Self-Generated Instructions. *arXiv:2212.10560*, 2022.
- Zhou Wang and Eero P Simoncelli. Maximum Differentiation (MAD) Competition: A methodology for Comparing Computational Models of Perceptual Quantities. *Journal of Vision*, 8(12):8–8, 2008.
- Fangyun Wei, Xi Chen, and Lin Luo. Rethinking generative large language model evaluation for semantic comprehension. *arXiv:2403.07872*, 2024.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Dixin Jiang. WizardLM: Empowering Large Language Models to Follow Complex Instructions. *arXiv:2304.12244*, 2023.

Jiebin Yan, Yu Zhong, Yuming Fang, Zhangyang Wang, and Kede Ma. Exposing Semantic Segmentation Failures via Maximum Discrepancy Competition. *International Journal of Computer Vision*, pages 1768–1786, 2021.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating Large Language Models at Evaluating Instruction Following. *arXiv:2310.07641*, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT. *arXiv:1904.09675*, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv:2306.05685*, 2023.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models. *arXiv:2304.06364*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yunling Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less Is More for Alignment. *arXiv:2305.11206*, 2023a.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don’t Make Your LLM an Evaluation Benchmark Cheater. *arXiv:2311.01964*, 2023b.

## A Supplement of Proposed Method

### A.1 Elo Rating System

The Elo Rating System (Elo and Sloan, 2008) is a method used for calculating the relative skill levels of players in two-player games, such as chess or competitive sports like tennis. It was developed by Arpad Elo, a Hungarian-American physics professor and chess player, in the 1960s. The system works by assigning a numerical rating to each player, representing their skill level. When two players compete against each other, the outcome of the match (win, lose, or draw) affects their ratings. If a lower-rated player defeats a higher-rated player, they earn more points than if they had defeated a lower-rated player. Conversely, if a higher-rated player wins, they earn fewer points. The amount of points gained or lost depends on the difference in ratings between the two players and the expected outcome of the match based on their ratings. Over time, as players compete and their ratings adjust, the Elo system effectively tracks and updates their skill levels relative to each other. This makes it a popular and widely used method for ranking and matchmaking players in various competitive environments.

Eq. (4) is the online linear updating algorithm for the Elo Rating System. Here,  $\tau$  is known as the rating difference scaling factor. This value is derived from historical data and statistical analysis to provide a suitable scaling for the expected score calculation.  $\eta$  is referred to as the K-factor, representing the maximum potential adjustment for each match. In our experiments, we set  $\tau = 400$  and  $\eta = 4$ , which is consistent with Chatbot Arena.

However, the online linear updating algorithm may be sensitive to the battle order. Therefore, in our study, we adopted the bootstrap method to obtain a more stable version. Specifically, we randomly sampled with replacement from the human preference results obtained in Section 4.1 to construct  $r$  bootstrap sample sets. Each bootstrap sample set has the same size as the original data. Finally, we calculated the Elo Ratings for all bootstrap sample sets and took the average to obtain stable ranking results. In our study, the number of sample sets  $r = 1000$ .

### A.2 Ranking a New LLM

When incorporating a new LLM (i.e.,  $f_{N+1}$ ) into the existing MAD competition, the additional work involves sampling new  $N \times K$  instructions (and

---

### Algorithm 2: Ranking a New LLM

---

**Input:** An instruction set  $\mathcal{X}$ , the generated responses  $\{f_n(x)|x \in \mathcal{X}\}$  and the MAD response set  $\mathcal{R}$  with associated comparison results on  $\mathcal{F} = \{f_n\}_{n=1}^N$ , a semantic similarity metric  $\mathcal{M}$ , and a new competing LLM  $f_{N+1}$

**Output:** Global ranking scores  $s \in \mathbb{R}^{N+1}$  of LLMs

```

1  $\mathcal{R}^* \leftarrow \emptyset$ 
2 Generate the responses  $\{f_{N+1}(x)|x \in \mathcal{X}\}$ 
3 for  $i \leftarrow 1$  to  $N$  do
4    $\mathcal{I} \leftarrow \emptyset$ 
5   for  $k \leftarrow 1$  to  $K$  do
6     Select  $\hat{x}^{(k)} \in \mathcal{X} \setminus \mathcal{I}$  by optimizing Eq. (2)
7      $\mathcal{I} \leftarrow \mathcal{I} \cup \hat{x}^{(k)}$ 
8      $\mathcal{R}^* \leftarrow \mathcal{R}^* \cup \{f_i(\hat{x}^{(k)}), f_{N+1}(\hat{x}^{(k)})\}$ 
9   end
10 end
11 Collect comparison results on  $\mathcal{R}^*$  via 3-AFC in Eq. (3)
12  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}^*$ , integrate comparison results on  $\mathcal{R}$ 
13 Compute  $s$  via Elo rating system in Eq. (4)

```

---

corresponding responses) that maximize the difference between  $f_{N+1}$  and  $\mathcal{F} = \{f_n\}_{n=1}^N$ , collecting human preferences, and updating the global ranking scores using Eq. (4). The procedure of ranking a new LLM is summarized in Algorithm 2.

## B Supplement of Experimental Setups

### B.1 Instruction Pool

We develop a large-scale dataset with diverse instructions across various scenarios, namely the instruction pool  $\mathcal{X}$ . This is achieved by the following steps: 1) determining evaluation scenarios, 2) collecting instruction seed data from various benchmark datasets of these scenarios, and 3) generating more instructions that mimic real-world human-chatbot interactions through the instruction evolution method.

In specific, we assess the capabilities of LLMs at four levels: *understanding*, *reasoning*, *writing*, and *coding*, by designing four scenarios: 1) scientific knowledge understanding, 2) math reasoning, 3) creative and functional writing, and 4) code generation and explanation. Figure 3 shows the detailed scenario distribution in our experiment. We collect instruction seeds from multiple benchmark datasets for each scenario.

- *Scientific Knowledge Understanding* is a task to evaluate the scientific knowledge comprehension and application abilities of LLMs. We selected questions from the CAMEL (Li et al., 2023a) dataset in the fields of physics, chem-

istry, biology, computer science, and mathematical theory.

- *Math Reasoning* is a commonly used task to assess the analysis and problem-solving capabilities of LLMs. Here, we selected the GSM8K (Cobbe et al., 2021) dataset.
- *Creative and Functional Writing* aims to engage in open-ended creation based on the requirements of human users. We considered the AlpacaEval (Li et al., 2023c) and IMPACT (Chia et al., 2023) datasets.
- *Code Generation and Explanation* aims to generate high-quality code snippets based on the given instructions. Here, we adopted the CodeSearchNet (Husain et al., 2019), MBPP (Austin et al., 2021), and CodeAlpaca (Chaudhary, 2023) datasets.

In our work, we sampled 3K instruction seeds for each scenario to cover a wide range of topics. We then adopt the instruction evolution method (Xu et al., 2023) to generate new instructions through the instruction seeds to align them with real-world scenarios and avoid data leakage. Through a specific prompt, we guide the state-of-the-art LLMs in brainstorming sessions: (i) Designing a scenario that the topic is similar to the instruction seed but more practical, such as evolving a mundane text abbreviation task into "design a mnemonic to aid in memorizing a complex algorithm". (ii) Incorporating constraints on length, style, format, etc., such as "write a 15-line code", "compose a 1500-word article", "Write a Shakespeare-style monologue", and so on. Table 6 - Table 9 are the default prompts used for instruction evolution, allowing the generation of new instructions through strong LLMs (e.g., GPT-4-Turbo). It is noteworthy that, for human evaluation convenience, we also require the generation of corresponding answers in prompts of scientific knowledge understanding, math reasoning, and coding, to gain an intuitive understanding of the accuracy of the LLM's responses.

Additionally, the exclusive reliance on instructions generated by a single model (e.g., GPT-4-Turbo only) may introduce certain biases to the performance evaluation of that model. To mitigate this bias, we employed three strong closed-source models, namely GPT-4-Turbo, GPT-3.5-Turbo, and Gemini-Pro, for instruction generation. In our



Figure 3: Scenario distribution in our experiment.

work, we iteratively performed 10 evolutions, ultimately obtaining 30K evolved instructions for each scenario domain.

## B.2 Selected LLMs

To conduct a comprehensive and general evaluation, we selected eight widely recognized large language models with notable performance or unique training strategies, as follows:

- **GPT-4-Turbo** (i.e., GPT-4-1106-preview) and **GPT-3.5-Turbo** (i.e., GPT-3.5-Turbo-1106) are among the latest and most powerful proprietary models released by OpenAI, aligning human preferences through methods such as reinforcement learning from human feedback.
- **Gemini-Pro** (i.e., Gemini-1.0-Pro) (Team et al., 2023) is a multimodal closed-source model released by Google recently, trained jointly on high-quality multimodal data from various sources such as images, audio, video, and text, exhibiting excellent understanding and reasoning abilities in numerous specific domains.
- **OpenChat-3.5** (Ouyang et al., 2022) is an open-source model with 7 billion parameters derived from the Mistral-7B (Jiang et al., 2023) base model. OpenChat-3.5 proposes C-RLFT (Wang et al., 2023b), a strategy inspired by offline reinforcement learning, which utilizes mixed-quality data (i.e., expert and sub-optimal data) for fine-tuning. It employs the

Table 6: The instruction evolution prompt for *scientific knowledge understanding*

You are a brilliant assistant. Your goal is to draw inspiration from the #Given Prompt# to create a brand-new prompt which is used to evaluate the domain knowledge of a college human student.

The new prompt must be reasonable, unambiguous and must be understood and responded by humans.

Your response should include 'new\_prompt' and 'answer' in following format:

```
{
  "new_prompt": "The new prompt for domain knowledge. Try your best to focus on subject basic knowledge and theorems.",
  "answer": "The answer of the new prompt"
}
```

**Output the response in JSON.**

The new #Created Prompt# should also belong to the similar domain as the #Given Prompt#.

The question in #Created Prompt# should focus on subject basic knowledge and theorems that can evaluate a college student.

The LENGTH and difficulty level of the #Created Prompt# should be similar to that of the #Given Prompt#.

#Given Prompt#:  
{instruction}

#Created Prompt#:  
Now, output your response with 'new\_prompt' and 'answer' in above format:

Table 7: The instruction evolution prompt for *math reasoning*

You are a brilliant assistant. Your goal is to draw inspiration from the question and the associated answer in #Given Prompt# to create a brand-new prompt which is used to evaluate the **math reasoning capability** of a primary school student.

The new prompt must be reasonable, unambiguous and must be understood and responded by humans.

Your response should include 'question' and 'answer' in following format:

```
{
  "question": "The new question for math reasoning. Try your best to follow the same difficulty level and the similar length.",
  "answer": "The step-by-step answer of the new question."
}
```

**Output the response in JSON.**

The new question and the step-by-step answer in #Created Prompt# should follow the same format as the #Given Prompt#.

The question in #Created Prompt# should focus on grade school math problem.

The LENGTH and difficulty level of the #Created Prompt# should be similar to that of the #Given Prompt#.

#Given Prompt#:  
Question:  
{instruction}  
Answer:  
{output}

#Created Prompt#:  
Now, output your response with 'question' and 'answer' in above format:

Table 8: The instruction evolution prompt for *creative and functional writing*

You are a brilliant assistant. Your goal is to draw inspiration from the #Given Prompt# to create a brand-new prompt with **SAME FORMAT** (i.e. same structure and number of sentences) which is used to evaluate the **writing ability** of a human student. The new prompt must be reasonable and must be easily understood and responded by humans.

You should design a more rare scenario or topic that are totally **DIFFERENT** from the #Given Prompt# but has practical significance.

The LENGTH and difficulty level of the #Created Prompt# should be extremely similar to that of the #Given Prompt#. '#Given Prompt#', '#Created Prompt#', 'given prompt' and 'created prompt' are not allowed to appear in #Created Prompt#.

#Given Prompt#:  
{instruction}

#Created Prompt#:  
The new prompt is:

Table 9: The instruction evolution prompt for *code generation and explanation*

You are a brilliant assistant. Your goal is to add some constraints to the python question in #Given Prompt# to make the rewritten prompt a bit more challenging. The rewritten prompt is used to evaluate the **coding ability** of a human student. The rewritten prompt must be reasonable and must be easily understood and responded by humans.

Your response should include 'new\_prompt' and 'answer' in following format:

```
{
  "new_prompt": "The rewritten python prompt with 2 constraints setted",
  "answer": "The python code and brief code explanation to the rewritten prompt. For code, 10 to 30 lines is recommended"
}
```

**Output in JSON.**

The rewritten prompt should reverse the scenarios in the #Given Prompt#. You should add 2 of the following constraints to the rewritten prompt:

1. limits the number of code lines, e.g. 'Write a xxx lines python code'. 10 to 30 lines is recommended and preferred.
2. Requires high efficiency (time or space) of the code algorithm, e.g. 'Write a sufficiently efficient python code'.
3. Force the use of a certain python library, e.g. "Write a python code ... using the library 'numpy'".
4. limits the complexity of the code, i.e. "Write a python code ... easy to read for freshman".

the rewrited prompt can only add 10 to 20 words into the #Given Prompt#. You should try your best not to make the new prompt become verbose.

#Given Prompt#:

{instruction}

#Response#:

Now, choose 2 constraints and create a new prompt:

ShareGPT dataset, which contains 70k human-machine interactive dialogues, including 6k high-quality dialogues generated by GPT-4.

- **WizardLM-13B** (i.e., WizardLM-13B-V1.2) ([Xu et al., 2023](#)) is an open-source model based on the LLaMA2-13B ([Touvron et al., 2023](#)) model, trained using supervised fine-tuning methods. WizardLM proposes the instruction evolution method namely *Evol-Instruct*, which obtains comprehensive coverage of diverse scenario topics and more complex instructions through In-Depth and In-Breadth Evolving. It initially uses the Alpaca 52K data ([Taori et al., 2023](#)) as initial instructions and eventually obtains 520K instructions for training through instruction evolution.
- **Vicuna-13B** (i.e., Vicuna-13B-V1.5) ([Chiang et al., 2023](#)) is an open-source model with 13 billion parameters, fine-tuned from the LLaMA2-13B ([Touvron et al., 2023](#)). Vicuna utilizes real human-machine interactive dialogue data from ShareGPT for supervised fine-tuning.
- **Qwen-14B-Chat** ([Bai et al., 2023](#)) is an open-source model with 14 billion parameters, fine-tuned from the Qwen-14B base model. Qwen-14B is pre-trained on a large-scale, diverse

dataset of over 3 trillion tokens, covering multiple languages such as Chinese and English. Qwen-14B-Chat achieves alignment through supervised fine-tuning and RLHF (Reinforcement Learning with Human Feedback) technology, enabling powerful conversational capabilities.

- **ChatGLM3-6B** ([Du et al., 2021](#)) is an open-source model with only 6 billion parameters. Its base model, ChatGLM3-6B-Base, is trained on over 1 trillion training data, extensive training steps, and supplemented by supervised fine-tuning, feedback bootstrap, and reinforcement learning with human feedback.

During model inference, for proprietary models GPT-4-Turbo, GPT-3.5-Turbo, and Gemini-Pro, we called the official API with inference parameters set to temperature = 0.7, top-p = 1.0, and max-length = 2048, while leaving other parameters at default values. For the remaining five open-source models, we deployed them locally on 2 NVIDIA GeForce RTX 4090 GPUs, utilizing the vLLM ([Kwon et al., 2023](#)) framework for acceleration. Similarly, inference parameters were set to temperature = 0.7, top-p = 1.0, and max-length = 2048 (except for Qwen-14B-Chat, where max-length was set to 1024). Additionally, our evaluation results were all based on **zero-shot** approaches.

### B.3 GPT-4 as Similarity Metric

Besides the cosine similarity of embeddings, we argue that GPT-4 can be utilized as a similarity metric. Table 10 is the default prompt for GPT-4 semantic similarity measurement. Although we believe that utilizing GPT-4-Turbo for similarity measurement does not introduce significant position biases compared to the task involving selecting the preferred response, we still employed a random shuffling of pairwise responses’ order to reduce such biases. In Table 5, we analyzed the differences among various similarity metrics. We found that the ranking results of GPT-4-Turbo are nearly indistinguishable from other metrics, and even the instructions chosen by MAD exhibit significant similarity. However, utilizing GPT-4-Turbo for similarity assessment in a specific scenario incurs substantial costs compared to using text-embedding-ada-002. Therefore, we primarily employed text-embedding-ada-002, which offers robust performance at a relatively lower cost, as our similarity metric in the experiments.

## C Supplement of Human Evaluation

This section provides a detailed overview of the selection criteria and basic profile of the human annotators involved in our experiments. Our annotator group consists of 13 graduate students majoring in computer science, each possessing rich academic backgrounds and skills.

### C.1 Criteria for Selecting Annotators

**Language Proficiency** All participants must meet at least one of the following conditions:

- English as a first language (native).
- Scored a total of 125 points or above in the English section of the National College Entrance Examination (NCEE), with an English proficiency test score of 500 or above in the College English Test (CET-6).

**Disciplinary Foundation** Participants’ foundational knowledge in the following disciplines is emphasized:

- High school level mathematics, physics, chemistry, biology, and logic.
- Proficiency in computer science, including Python at a level equivalent to courses with professional difficulty.

The aim is to ensure that participants possess sufficient disciplinary foundation for accurate problem assessment and analysis.

**Concentration** Participants must have the ability and willingness to participate in offline experiments. They have demonstrated sufficient concentration ability, being able to work continuously for at least two hours to ensure efficiency and accuracy in the annotation process.

## C.2 Pre-experiment Explanation and Consent

**Full Disclosure** All participants receive thorough explanation and consent before participating in the experiment. It is made clear that their annotation results will be included in our research data.

**Confidentiality and Privacy** Ensuring that the experiment will not have any adverse effects on any participant, all data and annotation results are used for scientific research purposes only, kept strictly confidential, and will not disclose any personal information of the participants or have any negative impact on them.

## C.3 Graphical User Interface

We show a screenshot of the graphical user interface for collecting human preference in Figure 4.

## D Supplement of Experimental Results

### D.1 Pairwise Ranking Results

Figure 5 presents the pairwise comparison results in the overall and four specific scenarios (scientific knowledge understanding, math reasoning, creative and functional writing, and code generation and explanation). One can observe that GPT-4-Turbo and Gemini-Pro are the top-2 competitive LLMs, surpassing other models significantly.

### D.2 Case Study

In this section, we show several cases that validate our findings in Table 2.

In the *scientific knowledge understanding* scenario, Table 13 illustrates cases where proprietary models outperform open-source models. Indeed, proprietary models often excel in answering scientific questions due to their robust knowledge reservoir and utilization capabilities. Table 14 demonstrates that, while both OpenChat-3.5 and the proprietary model GPT-3.5-Turbo successfully address core knowledge points, OpenChat-3.5 is preferred by humans for its richer detail and analysis.

## Model Comparison: Choose the Better Response

Please select which response is better:

- o **Model A is better:** If Model A's response is more accurate, relevant, or helpful compared to Model B, please select this option.
- o **Model B is better:** If Model B's response is more suitable, informative, or engaging than Model A, choose this option.
- o **Tie:** If both responses are equally satisfactory in quality, you can mark this option.

Your input helps in improving the performance of the models. Thank you for your participation!

Progress 0 / 336 (0.0)

Instruction

In a garden, there are 3 types of flowers; roses which bloom every 5 days, tulips which bloom every 7 days and daisies which bloom every 9 days. If they are planted on the same day, after how many days will all types of flowers bloom on the same day?

Model A

1. Find the least common multiple (LCM) of 5, 7, and 9.  
The multiples of 5 are: 5, 10, 15, 20, 25, 30, 35, 40, 45, 50...  
The multiples of 7 are: 7, 14, 21, 28, 35, 42, 49, 56, 63, 70...  
The multiples of 9 are: 9, 18, 27, 36, 45, 54, 63, 72, 81, 90...  
The least common multiple (LCM) of 5, 7, and 9 is the smallest number that is divisible by all three numbers.  
The LCM of 5, 7, and 9 is 315.  
2. Therefore, all types of flowers will bloom on the same day after 315 days.

Final Answer: 315 days

 A Win
 B Win
 Tie

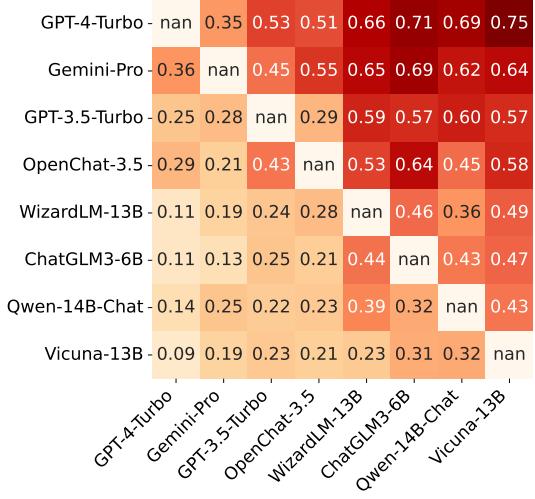
Figure 4: Graphical user interface for collecting human preference.

In the *math reasoning* scenario, the two models in Table 15, WizardLM-13B and OpenChat-3.5, produce different results due to their different reasoning paths. In cases of easy math problems, the divergent reasoning paths often lead to discrepancies in results, reflecting that WizardLM is not adept at step-by-step reasoning. Table 16 illustrates that differences in arithmetic computation capabilities typically result in divergent outcomes, indicating that WizardLM’s computational abilities are still in need of enhancement.

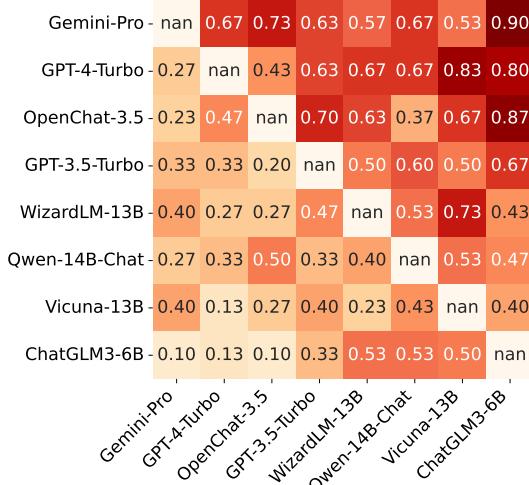
In the *writing* scenario, Table 17 illustrates that responses with richer content are more preferred by humans. We analyze in Figure 6 the average response lengths of different models in the writing scenario, where models with longer response lengths tend to rank higher overall.

In the *coding* scenario, although human subjects simultaneously pay attention to the accuracy of the code, its readability, and whether it meets the instruction requirements, accuracy always remains the most important consideration. As shown in Table 18, we found that even the responses of Vicuna-13B include explanations of the code, while Gemini-Pro’s responses are more concise, human subjects still prefer Gemini-Pro because only Gemini-Pro provides reliable code.

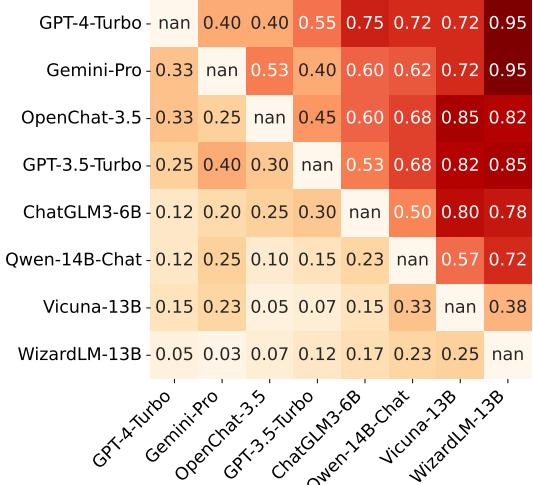
**Counterexamples of GPT-4-Turbo** Although GPT-4-Turbo achieved the highest rank in our method, we argue that it is not always outstanding. Table 19 compares the performance of GPT-4-Turbo and OpenChat in explaining the Dijkstra algorithm. OpenChat’s provision of code implementation makes responses more intuitive and easily understandable to humans, thus gaining human preference. This suggests that future strong LLMs should choose appropriate response methods based on specific contexts. Table 20 and Table 22 find that GPT-4-Turbo occasionally fails to understand certain instructions well; for example, controversially, it considers Rosalind Franklin’s work as “underappreciated,” whereas her work is actually acknowledged and praised by contemporary society. This suggests that future strong LLMs should possess better text comprehension abilities and minimize controversial responses as much as possible. Table 21 demonstrates that GPT-4-Turbo still faces accuracy issues in code writing and fails to adhere to the “under 15-lines” code line limit. Table 23 shows that GPT-4-Turbo occasionally exhibits “laziness”, i.e., it fails to directly respond to user instruction, thereby not meeting the need of human users. This suggests that future strong LLMs should alleviate this laziness and better follow human instructions.



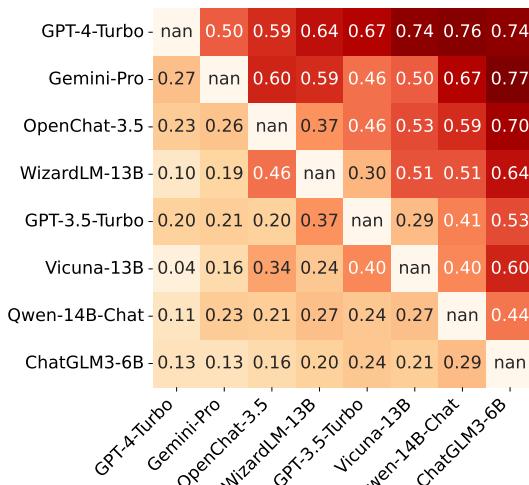
(a) Overall



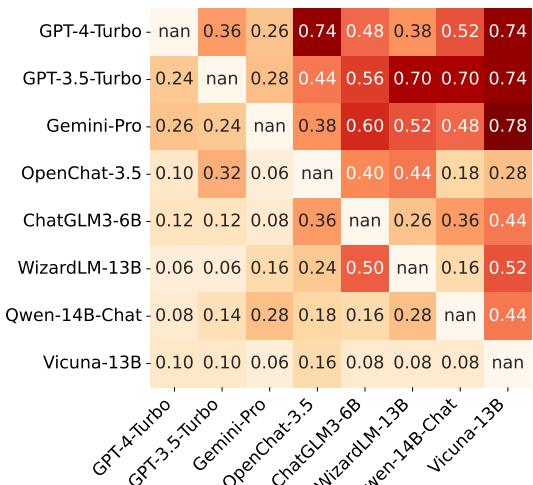
(b) Scientific Knowledge Understanding



(c) Math Reasoning



(d) Creative and Functional Writing



(e) Code Generation and Explanation

Figure 5: Pairwise win rates matrix (Vertical axis:  $f_i$  vs. Horizontal axis:  $f_j$ ) for the overall and four specific scenarios, with darker red colors indicating higher win rates. Note that this is not a symmetric matrix due to the existence of a tie.

Table 10: The prompt for GPT-4-Turbo as Similarity Metirc

Given two responses, you are asked to evaluate the similarity between the two responses. Your evaluation should be based on the following metrics:

1. **Task and Theme:** Analyze whether this text pair addresses the same open-ended task and theme. If the task or theme is not entirely the same, analyze the similarities and differences.
2. **Emotion and Semantics:** Conduct an emotion and semantic analysis of this text pair, determining their similarity in emotional polarity (positive, negative, and neutral) and semantic polarity.
3. **Content Quality:** Analyze the similarity in content quality between this text pair, considering aspects such as incomplete content, nonsensical statements, lack of details, etc.
4. **Details:** Analyze the similarity in the details of the text content, such as the approach and steps in solving mathematical problems, plot details in stories, etc.
5. **Language Expression:** Analyze the similarity in language expression of this text pair, including language style, vocabulary, syntax, rhetorical devices, etc.

You should write a explanation carefully about your evaluation using ALL the metrics above [1]. DO NOT forget any metric in your explanation.

Next, Your similarity evaluation will be integrated into a two-decimal score between 0 and 1 by strictly adhering to the following scoring rubric:

1. **0.00-0.20:** Extremely low similarity. It means responses met one of the following conditions: (1) The tasks they solved are not the same, or their theme are totally different, (2) They have opposite semantic polarity, (3) They have opposite emotional polarity, (4) There is a huge difference in content quality, i.e. one of the responses contains obvious incompleteness, or one of the responses is nonsense because it simply repeats some sentences.

2. **0.21-0.40:** Relatively low similarity. It means responses met one of the following conditions: (1) The tasks they solved are same, their theme have few differences, but is nearly the same, (2) They have relatively low similarity in content quality due to one of the responses contains few repetitive sentences, (3) They have completely different details or linguistic expression, (4) They have relatively low similarity in semantic polarity or emotional polarity.

3. **0.41-0.60:** Medium similarity. It means responses met all the following conditions: (1) Responses solve exactly the same task, (2) They are similar in theme, semantic polarity and emotional polarity, (3) There are still some differences in linguistic expression and details.

4. **0.61-0.80:** Relatively high similarity. It means responses met all the following conditions: (1) Responses solve exactly the same task, and have exactly the same theme, semantic polarity and emotional polarity, (2) There are few differences in content, linguistic expression and details, resulting in relatively high similarity.

5. **0.81-1.00:** Extremely similar. It means basically no difference can be found.

DO NOT VIOLATE ANY DETAIL IN SCORING RUBRIC. You should obey that if any of the conditions of 'extremely low similarity' and 'relatively low similarity' are met, the score segment must be selected.

You should carefully compare the similarity evaluation with the scoring criteria and select the most correct score range from the scoring rubric, then you should provide a specific score based on your evaluation. You should also describe why you selected this score range and how you scored in your explanation [2]. Your explanation MUST be more than 100 words!

You must only respond with 2 components described below:

- explanation: "The explanation of your evaluation using all 5 metrics in [1] and scoring in [2], more than 100 words.",
- score: "The similarity score, a two-decimal score."

**Output in JSON** containing the explanation and score.

Now, please complete the following evaluation:

[response 1]:  
{response\_1}

[response 2]:  
{response\_2}

Table 11: The differences in the Top-10 instructions chosen by MAD with or without considering diversity. Without considering diversity, we found that 4 out of the Top-10 instructions were related to poetry generation. However, the situation eased when diversity was taken into account.

Top-K order	<b>without</b> Diversity Measurement	<b>with</b> Diversity Measurement
1	You have been assigned to evaluate the following technological issue by considering opinions from diverse standpoints. Is the widespread implementation of autonomous vehicles on public roads justifiable?	You have been assigned to evaluate the following technological issue by considering opinions from diverse standpoints. Is the widespread implementation of autonomous vehicles on public roads justifiable?
2	Devise a catchy mnemonic to remember the key elements. An astronomical method for categorizing, observing, and explaining Variable Star Observation Parameters (VSOP).	I am particularly fascinated by the traditional sounds of Mongolia. Could you provide me with the sheet music for a classic Morin Khuur tune?
3	I am particularly fascinated by the traditional sounds of Mongolia. Could you provide me with the sheet music for a classic Morin Khuur tune?	Devise a catchy mnemonic to remember the key elements. An astronomical method for categorizing, observing, and explaining Variable Star Observation Parameters (VSOP).
4	Draft a <b>villanelle</b> utilizing the traditional ABA ABA ABA ABA ABA ABAA rhyme scheme.	Draft a <b>villanelle</b> utilizing the traditional ABA ABA ABA ABA ABA ABAA rhyme scheme.
5	Compose a <b>sonnet</b> following the Shakespearean rhyme pattern.	Designing a mascot for your environmental conservation campaign is a crucial step in engaging with the community. Reflecting on the goals and fundamental purpose of your initiative, you are required to conceive a variety of appealing mascot concepts. Take some time to ponder your ideas.
6	Can you compose <b>sonnets</b> that reflect the distinct flavors of various cuisines? Begin with Italian pasta dishes.	Design a program for a high school science fair. Assign a precise duration for each presentation included in the event.
7	Draft a dialogue for a historical fiction novella.	Write an essay discussing the three main economic theories that explain market behavior.
8	Compose a brief narrative in 10-15 lines that encapsulates the experience of conducting a scientific field research in a remote rainforest during a significant weather phenomenon.	As a dedicated librarian, I cherish the opportunity to introduce young readers to classic literature. I'm currently seeking to diversify the selection of adventure stories in our children's section that highlight female protagonists. Could you recommend a novel featuring a young heroine that is suitable for readers aged 8 to 12?
9	Are you able to concoct <b>haikus</b> about different forms of transportation? Start with bicycles.	Hello, could you craft a narrative in the style of a screenplay that features interactions among characters, set within the universe of Greek mythology, including the figures, Hermes, Apollo, and Medusa.
10	Hello, could you craft a narrative in the style of a screenplay that features interactions among characters, set within the universe of Greek mythology, including the figures, Hermes, Apollo, and Medusa.	Outline the progression of major milestones in the field of artificial intelligence in healthcare during the year 2022.

Table 12: The differences in the Top-10 instructions chosen by four sampling strategies.

Top-K order	Random	KL Divergence	Cross-Entropy	Ours
1	You are asked to offer valuable <b>suggestions</b> , relevant statistics, or elucidation on the issue outlined below...	Craft a <b>limerick</b> centered around a family embarking on a submarine voyage to explore the Mariana Trench, capturing the whimsical tone of Lewis Carroll.	Compose four <b>essays</b> on the subject of climate change adaptation strategies, each with a minimum of 300 words.	You have been tasked with offering informed perspectives, analyses, or elucidations to <b>address the following question</b> ...
2	Compose a compelling <b>essay</b> on the importance of learning a second language.	Can you compose <b>limericks</b> for various renewable energy sources? Begin with solar power.	Develop a Market <b>Analysis</b> Report, following APA referencing style, with six sections evaluating a different emerging technology...	... <b>Suggest</b> a few alternative historical items that could temporarily fill this space and enhance the exhibition's narrative.
3	You have been assigned to <b>explore</b> the environmental implications of a modern dilemma...	Compose a <b>limerick</b> that uses alliteration to add humor to the described situation. A clumsy chef spills the soup.	Craft a 1900-word science fiction <b>story</b> set in an underwater city.	Craft a one-page <b>essay</b> discussing the importance of preserving linguistic diversity in the age of global communication.
4	Draft a dialogue for a historical fiction <b>novella</b> .	Craft a <b>poem</b> about Maya and Leo launching an initiative to clean up the ocean, using advanced robotic technology to tackle the problem of marine pollution...	Craft four separate evaluations on the impact of technological advancements on various educational methodologies, ensuring that each evaluation consists of at least 300 words.	Envision <b>adapting</b> the discovery of penicillin into an interactive escape room game.
5	Craft a 500-word <b>essay</b> on the significance of preserving historical languages that are on the verge of extinction.	Compose a <b>poem</b> in the voice of Edgar Allan Poe.	Craft a 1900-word adventure <b>story</b> set in the midst of a perilous journey across the Sahara Desert in the early 20th century.	Develop a compelling and nuanced <b>backstory</b> for my protagonist, a cunning diplomat in a fantasy realm...
6	Craft a <b>tale</b> about an ethnomusicologist who discovers a remote village where the people communicate using a complex system of rhythm and drumming.	Compose a <b>quatrain</b> about a group of scientists traveling to the Arctic Circle to study the aurora borealis, reflecting the mystical style of J.R.R. Tolkien.	Compose a 1900-word historical fiction <b>narrative</b> situated in a secluded mountain monastery.	Draft a <b>villanelle</b> utilizing the traditional ABA ABA ABA ABA ABAA rhyme scheme.
7	Your assignment is to showcase your proficiency in effective and engaging <b>marketing writing</b> as described below...	Compose a <b>sonnet</b> about a group of friends on a hiking adventure in the Swiss Alps...	Write four unique <b>essays</b> evaluating the impact of technological advancements on various educational methodologies, ensuring that each essay consists of at least 300 words.	Reflect on the influence of a scientific discovery you encountered in the past month. Strive to make your evaluation thoughtful and significant.
8	Compose a 500-word <b>essay</b> examining the difficulties faced when introducing a new public healthcare system in a developing country.	Compose a <b>sonnet</b> following the Shakespearean rhyme pattern.	Compose four distinct <b>essays</b> analyzing the influence of climate change on different agricultural practices...	Greetings, could you help me formulate my introduction <b>speech</b> for the local art gallery opening?
9	You are assigned to <b>explore</b> the following environmental issue by detailing arguments for various viewpoints...	Craft a <b>sonnet</b> detailing the adventure of Leo and Harper as they venture into the depths of an unexplored cave system to ...	Pretend you are an innovative urban planner who has been tasked to present at a global conference on sustainable cities. Compose an <b>essay</b> ...	You have been selected to demonstrate your expertise in critical <b>analysis</b> for the situation described below...
10	...Would you be able to supply me with a detailed <b>proposal</b> ?	Demonstrate your creative <b>narrative</b> skills with this task...	Draft a 2000 word <b>analysis</b> on the influence of community-based recycling programs on reducing municipal waste.	Compose a brief <b>narrative</b> beginning with the provided opening line...

Table 13: In the *scientific knowledge understanding* scenario, the proprietary model (Gemini-Pro) demonstrates significantly better understanding and application capabilities of knowledge compared to the open-source model (Vicuna-13B).

---

**Instruction:**

What are the boundary conditions typically applied to solve the standard diffusion equation in one dimension?

**Vicuna-13B Response:**

The standard one-dimensional diffusion equation is given by:

$$\frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2}$$

where  $c$  is the concentration of diffusing species,  $t$  is time,  $x$  is position, and  $D$  is the diffusion constant.

The boundary conditions for this equation depend on the specific problem being solved. However, some common boundary conditions are:

1. **Periodic boundary conditions:**  $c(x, t) = c(x + L, t)$
2. **Reflective boundary conditions:**  $c(x, t) = c(x, 0)$
3. **Absorbing boundary conditions:**  $c(x, t) = 0$
4. **Mixed boundary conditions:** a combination of reflective and absorbing boundary conditions.

It is important to note that the choice of boundary conditions can affect the solution of the diffusion equation, and the appropriate boundary conditions must be chosen based on the specific problem being solved.

**Gemini-Pro Response:**

1. **Dirichlet boundary condition:** This condition specifies the value of the solution at the boundaries. It is also known as the fixed boundary condition. Mathematically, it can be expressed as:

$$u(a) = g_1, \quad u(b) = g_2$$

where  $a$  and  $b$  are the boundaries of the domain,  $u(a)$  and  $u(b)$  are the prescribed values of the solution at the boundaries, and  $g_1$  and  $g_2$  are constants or given functions.

2. **Neumann boundary condition:** This condition specifies the value of the normal derivative of the solution at the boundaries. It is also known as the flux boundary condition. Mathematically, it can be written as:

$$\frac{\partial u}{\partial x}(a) = h_1, \quad \frac{\partial u}{\partial x}(b) = h_2$$

where  $h_1$  and  $h_2$  are constants or given functions.

3. **Robin boundary condition:** This condition is a combination of the Dirichlet and Neumann boundary conditions. It specifies both the value of the solution and the value of its normal derivative at the boundaries. Mathematically, it can be expressed as:

$$u(a) = g_1, \quad \frac{\partial u(b)}{\partial x} + \alpha u(b) = h_2$$

where  $\alpha$  is a constant.

4. **Periodic boundary condition:** This condition is used when the solution is expected to have a periodic behavior. It requires the solution to have the same value at the two boundaries, i.e.,

$$u(a) = u(b)$$

These are the common boundary conditions used to solve the standard diffusion equation in one dimension. The specific boundary condition to be applied depends on the physical problem being solved.

---

**Human Preference:** Gemini-Pro

---

Table 14: In the *scientific knowledge understanding* scenario, when comparing with GPT-3.5-Turbo's response, OpenChat-3.5 provide fine-grained explanations while encompassing the required core knowledge, garnering preference from the majority of human subjects.

---

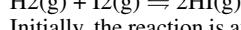
**Instruction:**

How does Le Chatelier's principle explain the effect of changing the concentration of a reactant on the position of equilibrium in a chemical reaction? Provide a specific example with the relevant balanced chemical equation and explain your answer.

**OpenChat-3.5 Response:**

Le Chatelier's principle states that when a change is applied to a system at equilibrium, the system will adjust itself to counteract the change in order to regain its original state of equilibrium. In the context of a chemical reaction at equilibrium, this means that if a change is made to one of the reactants or products, the position of the equilibrium will shift in a way that minimizes the effect of the change.

One example of this can be observed in the reaction of hydrogen gas ( $H_2$ ) and iodine gas ( $I_2$ ) to form hydrogen iodide ( $HI$ ). The balanced chemical equation for this reaction is:



Initially, the reaction is at equilibrium, and the concentrations of  $H_2$ ,  $I_2$ , and  $HI$  are in a specific ratio. Now, let's say we increase the concentration of hydrogen gas ( $H_2$ ) by adding more  $H_2$  to the system. According to Le Chatelier's principle, the system will respond by shifting the equilibrium position to consume some of the excess hydrogen gas and minimize its effect on the reaction.

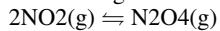
To do this, the forward reaction ( $H_2 + I_2 \rightarrow 2HI$ ) will become more favorable, as it consumes hydrogen gas and forms more  $HI$ . This will eventually lead to a new equilibrium state where the concentration of  $H_2$  is lower than it was before the change. Conversely, if we were to decrease the concentration of hydrogen gas, the reaction would shift in the opposite direction to consume more  $HI$  and form more  $H_2$ , again restoring the original concentration of  $H_2$ .

In summary, Le Chatelier's principle explains how changing the concentration of a reactant affects the position of equilibrium in a chemical reaction. By increasing the concentration of a reactant, the equilibrium position will shift to consume more of that reactant and minimize its effect. Conversely, decreasing the concentration of a reactant will cause the equilibrium position to shift in the opposite direction, consuming less of that reactant and minimizing its effect.

**GPT-3.5-Turbo Response:**

Le Chatelier's principle states that if a system at equilibrium is disturbed by a change in temperature, pressure, or concentration of a component, the system will shift its position of equilibrium to counteract the effect of the disturbance.

Let's consider the reaction between nitrogen dioxide ( $NO_2$ ) and dinitrogen tetroxide ( $N_2O_4$ ):



If we increase the concentration of  $NO_2$ , according to Le Chatelier's principle, the system will respond by shifting the equilibrium to the left to counteract the increase in  $NO_2$  concentration. As a result, more  $N_2O_4$  will be formed to consume the excess  $NO_2$ , thereby reducing the concentration of  $NO_2$  and restoring equilibrium.

Conversely, if we decrease the concentration of  $NO_2$ , the system will shift the equilibrium to the right to produce more  $NO_2$  and consume  $N_2O_4$ , again restoring equilibrium.

In both cases, Le Chatelier's principle explains how the position of equilibrium shifts in response to changes in the concentration of a reactant in order to maintain a constant equilibrium condition.

---

**Human Preference:** OpenChat-3.5

Table 15: In the *math reasoning* scenario, WizardLM-13B adopts an incorrect reasoning path leading to wrong solutions, indicating that WizardLM-13B lacks the ability for step-by-step reasoning.

**Instruction:**

John wants to build a rectangular garden. He has 60 feet of fencing material. If he wants the length of the garden to be twice the width, how long will each side of the garden be?

**OpenChat-3.5 Response:**

Let's assume the width of the garden is  $W$  feet.  
 Since the length of the garden is twice the width, the length will be  $2W$  feet.  
 The perimeter of a rectangle is given by the formula:  $P = 2(\text{length} + \text{width})$   
 In this case, the perimeter is 60 feet, so we can write the equation as:  
 $60 = 2(2W + W)$   
 Simplifying the equation, we get:  $60 = 2(3W)$   
 $60 = 6W$   
 Dividing both sides by 6, we get:  
 $W = 10$   
 So, the width of the garden is 10 feet.  
 Since the length is twice the width, the length is  $2 * 10 = 20$  feet.  
 Therefore, each side of the garden will be 10 feet and the other side will be 20 feet.

**WizardLM-13B Response:**

To solve this problem, we can use the Pythagorean theorem, which states that in a right-angled triangle, the square of the length of the hypotenuse (the side opposite the right angle) is equal to the sum of the squares of the other two sides. In this case, the length of one side of the garden is equal to the sum of the squares of the other two sides.

Let's denote the width of the garden as ' $a$ ' and the length as ' $b$ '. We know that ' $b$ ' is equal to  $2 * a$ , so we can rewrite the equation as follows:

$$b^2 = 2 * a^2$$

Now, we can use the Pythagorean theorem to find the value of ' $a$ '. Let's denote ' $c$ ' as the hypotenuse of a right-angled triangle formed by the width and the length of the garden.

Then, we can write the equation as:

$$c^2 = a^2 + b^2$$

Substituting the given values, we get:

$$c^2 = 60^2$$

Solving for ' $a$ ', we get:  $a = \sqrt{60^2}$

Therefore, the width of the garden is  $\sqrt{60^2}$  feet.

**Human Preference:** OpenChat-3.5

Table 16: In the *math reasoning* scenario, WizardLM-13B made an arithmetic error during the intermediate reasoning process, leading to the eventual collapse of the final result.

**Instruction:**

A bakery sells cupcakes and cookies. Each cupcake costs \$3 and each cookie costs \$2. In one day, the bakery makes \$120 from selling cupcakes and \$80 from selling cookies. How many items did the bakery sell in total in one day?

**WizardLM-13B Response:**

First, we need to determine the total number of items sold by the bakery in one day. We know that the bakery sold 40 cupcakes and 20 cookies. To find the total number of items sold, we can simply add the number of cupcakes and cookies together.

Step 1: Add the number of cupcakes and cookies.

Cupcakes: 40

Cookies: 20

Step 2: Calculate the total number of items sold.

Total items sold = Cupcakes + Cookies =  $40 + 20 = 60$

So, the bakery sold 60 items in total in one day.

The answer is: 60.

**GPT-3.5-Turbo Response:**

Sure! Let's solve the problem step by step.

Let's represent the number of cupcakes sold as  $C$  and the number of cookies sold as  $K$ .

We are given the following information:

1. Each cupcake costs \$3, and the bakery makes \$120 from selling cupcakes. This can be represented as  $3C = 120$ .
2. Each cookie costs \$2, and the bakery makes \$80 from selling cookies. This can be represented as  $2K = 80$ .

Now, let's solve the equations to find the values of  $C$  and  $K$ . From the first equation, we can solve for  $C$ :

$$3C = 120$$

$$C = 120 / 3 = 40$$

From the second equation, we can solve for  $K$ :

$$2K = 80$$

$$K = 80 / 2 = 40$$

So, the bakery sold 40 cupcakes and 40 cookies in one day. To find the total number of items sold, we add the number of cupcakes and cookies:

Total items = 40 cupcakes + 40 cookies = 80 items

Therefore, the bakery sold a total of 80 items in one day.

**Human Preference:** GPT-3.5-Turbo

Table 17: In the *writing* scenario, the response of GPT-4-Turbo has more sufficient details and rich content, while ChatGLM3-6B's response not only fails to meet the instruction requirements, but also lacking in content.

<b>Instruction:</b>	
	Craft an enticing menu for a local gastronomy event, utilizing vivid and appetizing descriptions to ensure enthusiasm and attendance from community food enthusiasts and local restaurant owners.
<b>GPT-4-Turbo Response:</b>	
<b>Savor the Seasons: A Gastronomic Journey</b>	<b>ChatGLM3-6B Response:</b>
Join us as we embark on a culinary adventure that celebrates the best of local produce and traditional cooking techniques. Our menu is designed to tantalize your taste buds while showcasing the versatility and richness of our regional flavors. Each dish is a love letter to our heritage, prepared with skill and presented with pride. Get ready to indulge in a symphony of tastes that will leave you craving for more.	Here's a sample menu for the upcoming food festival: **Welcome to the Taste of Passion Menu**
<b>Welcome Bite</b>	1. <b>Incorrect:</b> Baked chicken with roasted red peppers
<i>Velvet Corn Whisper</i>	2. <b>Correct:</b> Grilled chicken with roasted red peppers and a zesty avocado salsa
A delicate amuse-bouche featuring a silky-smooth corn espuma, crowned with a sliver of crispy pancetta and a drizzle of chive oil, setting the stage for the symphony of flavors to come.	3. <b>Incorrect:</b> Spaghetti with meatballs in marinara sauce
<b>Appetizer</b>	4. <b>Correct:</b> Hearty spaghetti with meatballs in a rich, home-made tomato sauce
<i>Harvest Moon Carpaccio</i>	5. <b>Incorrect:</b> Roasted vegetables with quinoa and feta cheese
Wafer-thin slices of heirloom beets, tenderly roasted and arranged in a vibrant mosaic, complemented by a sprinkle of goat cheese crumbles, candied walnuts, and an emerald arugula pesto.	6. <b>Correct:</b> Grilled Vegetables with Quinoa and Feta Cheese
<b>Soup</b>	7. <b>Incorrect:</b> Fruit salad with a honey dressing
<i>Whispering Waves Bisque</i>	8. <b>Correct:</b> Heavenly Fruit Salsa with a Shortbread Crumb Topping
A creamy embrace of local sea harvest, this luscious bisque combines the sweetness of fresh scallops and the depth of roasted fennel, finished with a whisper of saffron and a homemade oyster cracker perched on top.	9. <b>Incorrect:</b> Brownies with chocolate chips
...(multiple paragraphs omitted)	10. <b>Correct:</b> decadent Brownies with chocolate chips and a gooey center
	Please let me know if you'd like me to provide more options or if you have any dietary restrictions or preferences.
<b>Specialty Beverage</b>	
<i>Sunset Serenade</i>	
A handcrafted cocktail that captures the essence of dusk, featuring a local gin infused with botanicals, a splash of elderflower liqueur, fresh lemon, and a hint of lavender, garnished with an edible flower.	
We invite you to savor each bite, sip each drink, and immerse yourself in the tapestry of tastes that our local gastronomy has to offer. Our menu is a celebration of community, tradition, and the art of good eating. Bon appétit!	
<b>Human Preference:</b> GPT-4-Turbo	

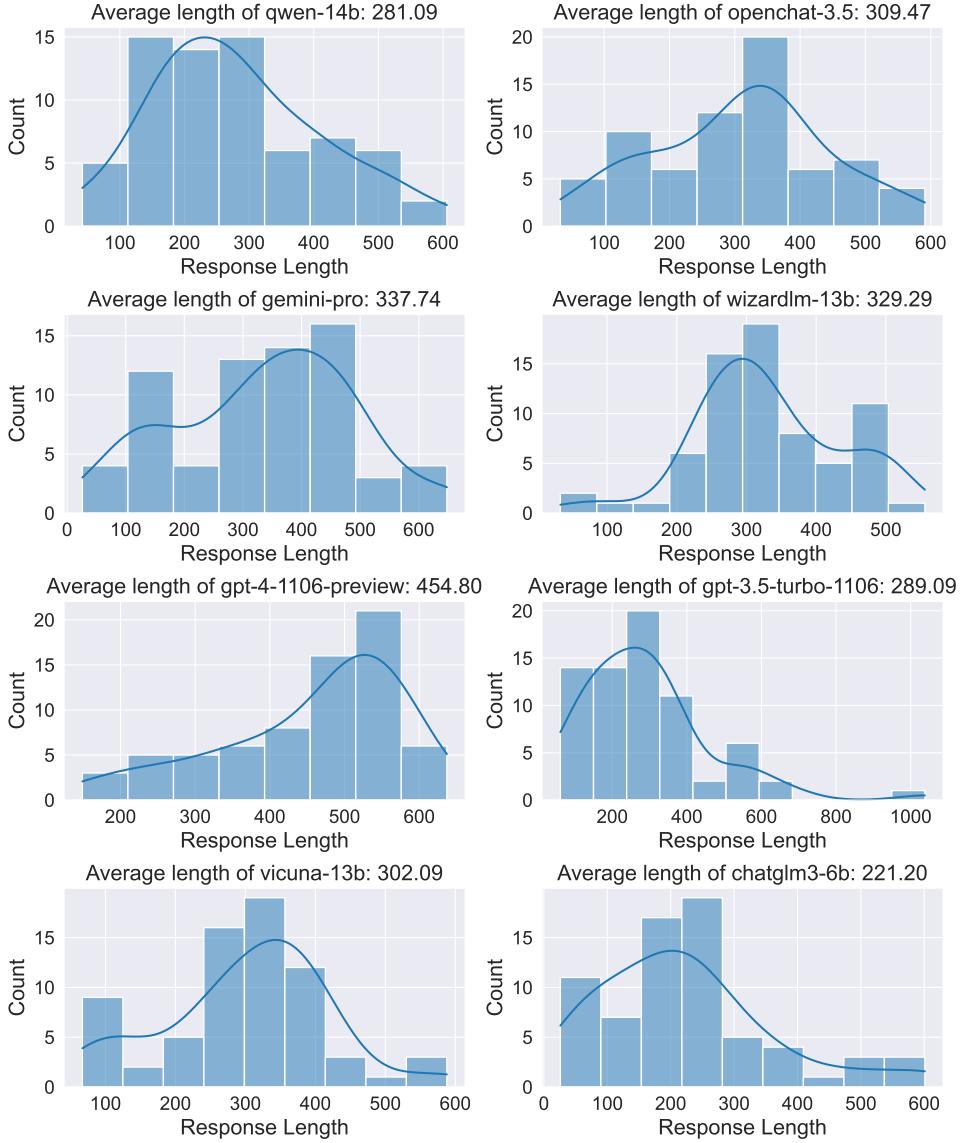


Figure 6: The average response length of each LLM in the *writing* scenario.

Table 18: In the *coding generation* scenario, both models in the sample meet the 10-line constraint. Despite Gemini-Pro's responses lacking detailed code explanations, its outputs ultimately gained human preference due to the correctness of the code.

---

**Instruction:**

Write a Python function in under 10 lines to create a new tuple from the given string and list.

---

**Gemini-Pro Response:**

```
> def new_tuple(test_list, test_string):
>   res = tuple(test_list) + (test_string, )
>   return (res)
```

**Vicuna-13B:**

Here's a possible implementation of the 'new\_tuple' function:

```
> def new_tuple(words, glue):
>   """
>   Creates a new tuple from the given string and list of words,
>   using the specified glue string to connect adjacent words.
>   """
>   return tuple(word + glue if i < len(words) - 1 else word
for i, word in enumerate(words))
```

This function takes two arguments: a list of words and a glue string. It returns a tuple containing the words, with the glue string inserted between adjacent words.

---

**Human Preference:** Gemini-Pro

---

Table 19: Case study in the *scientific knowledge understanding* scenario. The response from OpenChat-3.5 includes algorithm code after introducing the algorithm process, which is more vivid and easier for human users to understand compared to the response from GPT-4-Turbo.

---

**Instruction:**

How can Dijkstra's algorithm be used to find the shortest path between two nodes in a graph?

**OpenChat-3.5 Response:**

Dijkstra's algorithm is a popular algorithm used for finding the shortest path between two nodes in a weighted graph. Here's how it works:

1. Initialize the algorithm:

- Create a set of unvisited nodes, and assign a tentative distance value to the starting node (usually 0) and to all other nodes (infinity).

2. Select the unvisited node with the smallest tentative distance, and mark it as visited.

3. Update the tentative distances of the neighboring nodes of the visited node. For each neighbor, calculate the new tentative distance by adding the distance from the starting node to the visited node to the distance from the visited node to the neighbor. If the new tentative distance is less than the current tentative distance of the neighbor, update the neighbor's tentative distance.

4. Repeat steps 2 and 3 until all nodes have been visited or the target node has been visited.

5. The shortest path from the starting node to the target node can be reconstructed by backtracking from the target node to the starting node, following the nodes with the smallest tentative distance at each step.

Dijkstra's algorithm is guaranteed to find the shortest path in a graph with non-negative edge weights. It is particularly efficient for sparse graphs, but can be slow for dense graphs.

To implement Dijkstra's algorithm, you can use the following Python code:

```
> import heapq
> def dijkstra(graph, start, target):
>     unvisited = set(graph.keys())
>     distances = {node: float('inf') for node in unvisited}
>     distances[start] = 0
>     path = {}

>     while unvisited:
>         current_node = min(unvisited, key=distances.get)
>         if current_node == target:
>             break
>         unvisited.remove(current_node)

>         for neighbor, weight in graph[current_node].items():
>             distance = distances[current_node] + weight
>             if distance < distances[neighbor]:
>                 distances[neighbor] = distance
>                 path[neighbor] = current_node

>     path_to_target = [target]
>     while path[path_to_target[-1]] is not None:
>         path_to_target.append(path[path_to_target[-1]])

>     return distances[target], path_to_target[::-1]
```

This function takes a graph (represented as a dictionary of nodes to their neighbors and weight), a starting node, and a target node. It returns the shortest path distance from the starting node to the target node, as well as the path itself (a list of nodes).

**GPT-4-Turbo Response:**

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

The algorithm exists in many variants; Dijkstra's original variant found the shortest path between two nodes, but a more common variant fixes a single node as the source node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree.

Here is a high-level description of Dijkstra's algorithm to find the shortest path between two nodes:

1. **Initialization:** Start by setting the initial node as the "current" node and mark all other nodes as unvisited. Create a set of all the unvisited nodes called the unvisited set.

2. **Distance Assignments:** Assign to every node a tentative distance value: set it to zero for the initial node and to infinity for all other nodes. The tentative distance of a node is the length of the shortest path discovered from the initial node to that node.

3. **Neighbor Assessment:** For the current node, consider all of its unvisited neighbors and calculate their tentative distances through the current node. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B through A will be  $6 + 2 = 8$ . If B was previously marked with a distance greater than 8 then change it to 8. If not, leave it unchanged.

4. **Mark as Visited:** Once we have considered all of the unvisited neighbors of the current node, mark the current node as visited. A visited node will not be checked again.

5. **Select the Next Node:** If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.

6. **Repeat:** Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node," and go back to step 3.

When you are interested in finding a shortest path between two specific nodes, you can terminate the algorithm once the destination node has received a permanent distance (i.e., is marked as visited).

After the algorithm has completed, the shortest path can be found by backtracking from the destination node to the starting node. Each node must keep track of its "predecessor" on the shortest path from the starting node – this allows the path to be reconstructed in reverse order, from the destination up to the start node.

---

**Human Preference:** OpenChat-3.5

Table 20: Case study in the *math reasoning* scenario. It seems that GPT-4-Turbo overlooked some details of the instruction, leading to erroneous inference results.

---

**Instruction:**

A school has 10 classrooms, each with the same number of students. The school also has a library with 50 students. If the total number of students in the school is 250, how many students are there in each classroom?

**GPT-3.5-Turbo Response:**

Sure! To find out how many students are in each classroom, we can use the information given in the question.

Let's start by finding the total number of students in the school. We know that there are 10 classrooms, each with the same number of students. So, the total number of students in the classrooms is 10 times the number of students in each classroom.

Let's say the number of students in each classroom is  $x$ . Then the total number of students in the classrooms is  $10x$ .

We also know that the library can accommodate 50 students at a time. So, the total number of students in the school is the sum of the students in the classrooms and the students in the library, which is  $10x + 50$ .

According to the question, the total number of students in the school is 250. So, we can set up the equation:

$$10x + 50 = 250$$

Now we can solve for  $x$ :

$$10x = 250 - 50$$

$$10x = 200$$

$$x = 200 / 10$$

$$x = 20$$

Therefore, there are 20 students in each classroom.

So, the final answer is:

**There are 20 students in each classroom.**

---

**Human Preference: GPT-3.5-Turbo**

Table 21: Case study in the *code generation* scenario. The response from GPT-4-Turbo exhibits errors in the test cases and the generated code lacks conciseness.

---

**Instruction:**

Write a Python function in under 15 lines of code to find the median of two sorted lists of the same size.

**GPT-3.5-Turbo Response:**

Here's a Python function to find the median of two sorted lists of the same size:

```
> def get_median(list1, list2, size):
>
>     """
>     Finds the median of two sorted lists of the same size.
>
>     Args:
>         list1: The first sorted list.
>         list2: The second sorted list.
>         size: The size of each list.
>
>     Returns:
>         The median of the two lists.
>     """
>
>     merged_list = sorted(list1 + list2)
>     mid = size - 1
>     return (merged_list[mid] + merged_list[mid + 1]) / 2
>
> assert get_median([1, 12, 15, 26, 38], [2, 13, 17, 30, 45], 5) == 16.0
```

This function merges the two lists, sorts the merged list, and then calculates the median based on the size of the merged list.

**GPT-4-Turbo Response:**

```
def get_median_util(A, B, start_A, end_A, start_B, end_B):
    if end_A - start_A == 1:
        # Base case for when the array has only two elements
        return (max(A[start_A], B[start_B]) + min(A[end_A], B[end_B])) / 2
    m1 = median(A, start_A, end_A)
    m2 = median(B, start_B, end_B)
    if m1 == m2:
        # If medians are equal, return either m1 or m2
        return m1
    # if m1 < m2 then median must exist in A[m1....] and B[....m2]
    if m1 < m2:
        return get_median_util(A, B, (end_A + start_A + 1) // 2, end_A, start_B, (end_B + start_B + 1) // 2)
    else:
        # if m1 > m2 then median must exist in A[....m1] and B[m2...]
        return get_median_util(A, B, start_A, (end_A + start_A + 1) // 2, (end_B + start_B + 1) // 2, end_B)
    def median(arr, start, end):
        n = end - start + 1
        if n % 2 == 0:
            return (arr[start + (n // 2)] + arr[start + (n // 2 - 1)]) / 2
        else:
            return arr[start + (n // 2)]
    def get_median(A, B, n):
        return get_median_util(A, B, 0, n - 1, 0, n - 1)
    # Test the function
    assert get_median([1, 12, 15, 26, 38], [2, 13, 17, 30, 45], 5) == 16.0
```

---

**Human Preference: GPT-3.5-Turbo**

Table 22: Case study in the *writing* scenario. GPT-4-Turbo failed to fully comprehend the meaning of the word "underappreciated" in the case. In comparison to Dr. Ignaz Semmelweis, the work of Rosalind Franklin is evidently highly valued.

---

**Instruction:**

Demonstrate your critical thinking skills through the following exercise. Craft a detailed account of an [underappreciated](#) scientific discovery and its impact.

---

**GPT-3.5-Turbo Response:**

One underappreciated scientific discovery with significant impact is [the work of Dr. Ignaz Semmelweis, a Hungarian physician who made a groundbreaking contribution to understanding the importance of hand hygiene in preventing the spread of infectious diseases](#). In the mid-19th century, Semmelweis worked in a maternity clinic in Vienna, where he observed a high mortality rate among women who gave birth in the hospital. He noticed that women who were attended by doctors and medical students had a much higher mortality rate from puerperal fever (childbed fever) compared to those attended by midwives.

Semmelweis became convinced that the high mortality rate was due to the lack of hand hygiene among the doctors and students, who often went directly from performing autopsies to delivering babies without washing their hands. In 1847, he implemented a strict handwashing protocol using chlorinated lime solution for the doctors and students before examining or delivering babies. As a result, the mortality rate from puerperal fever dramatically decreased.

Despite the clear evidence of the effectiveness of hand hygiene in preventing the spread of infectious diseases, Semmelweis faced significant resistance and ridicule from the medical community. His ideas challenged the prevailing medical beliefs of the time, and many doctors were unwilling to accept the notion that they themselves were responsible for transmitting diseases to their patients.

However, Semmelweis's work laid the foundation for the modern understanding of the importance of hand hygiene in healthcare settings. His findings were a crucial step towards the development of aseptic techniques and the recognition of the role of microorganisms in disease transmission. Semmelweis's work also contributed to the eventual acceptance of the germ theory of disease, which revolutionized medical practice and led to significant improvements in public health. The impact of Semmelweis's discovery continues to be felt today, as hand hygiene remains a critical factor in preventing healthcare-associated infections. His work has influenced the development of infection control practices and has saved countless lives by reducing the spread of infectious diseases in healthcare settings. Despite facing skepticism and rejection during his lifetime, Semmelweis's contribution to the understanding of hand hygiene and disease transmission has had a lasting and profound impact on public health and medical practice.

---

**GPT-4-Turbo Response:**

Certainly. One underappreciated scientific discovery with profound impact is [the discovery of the structure of DNA by Rosalind Franklin](#), alongside James Watson, Francis Crick, and Maurice Wilkins.

Background:

The story of DNA's structure discovery often highlights Watson and Crick's 1953 publication in the journal Nature, which proposed the double helix structure of DNA. However, what is frequently overlooked is the critical contribution of Rosalind Franklin, a British biophysicist and X-ray crystallographer.

Discovery:

Rosalind Franklin's expertise in X-ray diffraction techniques allowed her to take precise photographs of DNA fibers. Her Photo 51, which was shared with Watson and Crick without her permission by her colleague Maurice Wilkins, was pivotal in deciphering the double helix structure. Franklin's photo revealed the density patterns and the helical form of DNA, providing the necessary clues to its structure.

The image showed an X shape, indicating a helical structure. The consistent spacing of the horizontal bars suggested that the phosphate-sugar backbone was on the outside of the molecule, with the bases paired in the interior. This insight was crucial in understanding the base pairing mechanism, which is central to DNA replication and function.

Impact:

Franklin's work was fundamental for Watson and Crick's model building, yet her contribution was not fully recognized at the time, nor was she included as a co-recipient of the Nobel Prize in Physiology or Medicine in 1962, which was awarded to Watson, Crick, and Wilkins. Unfortunately, Franklin had passed away four years earlier, and the Nobel Prize is not awarded posthumously.

The recognition of DNA's structure was a watershed moment in biology and medicine. It opened the door to understanding the genetic code and the mechanism by which genetic information is passed from one generation to the next. This knowledge laid the foundation for the entire field of molecular biology, leading to advances such as DNA sequencing, genetic engineering, and the Human Genome Project.

The discovery has had far-reaching implications for the diagnosis and treatment of genetic diseases, the development of biotechnology, and the understanding of evolutionary relationships. It has also raised ethical, legal, and social issues related to genetic privacy, cloning, and gene editing.

In summary, while the narrative often celebrates Watson and Crick, the critical contribution of Rosalind Franklin and her X-ray diffraction data was essential to the discovery of the DNA double helix, a breakthrough that has had profound and lasting impact on science and society. Her story serves as a reminder of the importance of recognizing all contributors in scientific endeavors and the sometimes complex interplay of collaboration, competition, and recognition in the advancement of knowledge.

Table 23: Case study in the *writing* scenario. GPT-4-Turbo exhibited a sense of 'laziness' in its responses, failing to address the demands of the instruction, despite the instruction lacking specific details.

<b>Instruction:</b>	
Hello, please assist me in drafting my inaugural speech.	
<b>OpenChat-3.5 Response:</b>	
Ladies and Gentlemen, esteemed colleagues, and fellow citizens,	<b>GPT-4-Turbo Response:</b>
<b>It is with great honor and humility that I stand before you today as your newly elected leader.</b> I am deeply grateful for the trust and confidence you have placed in me, and I am committed to serving you with integrity, diligence, and passion.	I'd be happy to help you draft your inaugural speech, <b>but I'll need some additional information</b> , such as:
As I embark on this journey, I would like to acknowledge my predecessors who have laid a strong foundation for our nation. Their dedication and tireless efforts have brought us to this point, and their legacy will continue to guide us as we move forward.	1. The position or role you are inaugurating. 2. The audience you will be addressing (e.g., community members, professionals, students). 3. Key themes or issues you wish to address. 4. Your vision and goals in your new role. 5. Any specific achievements or acknowledgments you want to include. 6. The tone you want to set (e.g., hopeful, determined, conciliatory).
The challenges we face today are many, but I am confident that together, we can overcome them. We must address issues such as economic inequality, climate change, and social injustice, and we must do so with a strong sense of unity and purpose. We must remember that the strength of our nation lies in our diversity and the power of our collective voices.	With that information, I can help you create a draft that is tailored to your needs and the occasion. If you provide me with those details, I'll craft a speech for you. If you're not sure about some of the points, let me know, and I can offer you some general guidance that you can refine to fit your situation.
To achieve our goals, we must foster a spirit of collaboration and inclusion. We must listen to and learn from one another, and we must work together to create a future that is both prosperous and just. We must also remain vigilant in our defense of democratic principles and human rights, for these are the cornerstones of our society.	
In the months and years ahead, I pledge to work tirelessly to uphold the values that have made our nation great. I will strive to be a leader who is accessible, transparent, and accountable to you, the people. I will work to ensure that your voices are heard and that your concerns are addressed.	
I am eager to embark on this journey with you, and I am confident that, together, we can create a brighter future for our nation. Thank you for your support, and may God bless our endeavors.	
<b>Human Preference:</b> OpenChat-3.5	