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

**Kehua Feng**[1,2*], **Keyan Ding**[1,2*], **Hongzhi Tan**[3], **Kede Ma**[4†], **Zhihua Wang**[4], **Shuangquan Guo**[3], **Yuzhou Cheng**[5], **Ge Sun**[5], **Guozhou Zheng**[6], **Qiang Zhang**[1,2†], **Huajun Chen**[1,2†]
[1] Zhejiang University [2]ZJU-Hangzhou Global Scientific and Technological Innovation Center
[3]Shanghai Electric Group Co., Ltd. Central Academe [4]City University of Hong Kong
[5]Shanghai Institute for Advanced Study of Zhejiang University
[6]Zhoushan-Zhejiang University Ocean Research Center
{kehuafeng, dingkeyan, huajunsir}@zju.edu.cn

## Abstract

Reliable evaluation of large language models (LLMs) is impeded by two key challenges: objective metrics often fail to reflect human perception of natural language, and exhaustive human labeling is prohibitively expensive. Here, we propose a sample-efficient human evaluation method for LLMs based on the principle of MAximum Discrepancy (MAD) Competition. Our method automatically and adaptively selects a compact set of input instructions that maximize semantic discrepancy between pairs of LLM responses. Human evaluators then perform three-alternative forced choices on these paired responses, which are aggregated into a global ranking using Elo rating. We apply our approach to compare eight widely used LLMs across four tasks: scientific knowledge understanding, mathematical reasoning, creative and functional writing, and code generation and explanation. Experimental results show that our sample-efficient evaluation method recovers "gold-standard" model rankings with a handful of MAD-selected instructions, reveals respective strengths and weaknesses of each LLM, and offers nuanced insights to guide future LLM development. Code is available at `https://github.com/weiji-Feng/MAD-Eval`.

## 1 Introduction

Since the advent of ChatGPT, there has been an unprecedented surge in the development of large language models (LLMs) (Touvron et al., 2023; Bai et al., 2023; OpenAI, 2023; Jiang et al., 2023; Team et al., 2023; Chen, 2023; Azaria et al., 2024), driven by self-supervised pretraining (Jaiswal et al., 2020), supervised fine-tuning (Wang et al., 2022; Chiang et al., 2023; Xu et al., 2023) and reinforcement learning (Ouyang et al., 2022). These models now exhibit remarkable general-purpose capabilities in language generation, understanding, and reasoning. Yet, with so many "competitive" LLMs emerging in rapid succession, establishing a reliable, scalable evaluation paradigm that reveals their strengths and weaknesses has become critical (Guo et al., 2023; Li et al., 2023c; Chang et al., 2024).

Traditional evaluation relies on fixed, human-annotated benchmarks—such as MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023) and BIG-bench (Srivastava et al., 2022)—and on objective metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). However, human annotation is costly and slow, and consequently, these benchmarks can only cover a narrow slice of possible task scenarios. Moreover, standard metrics frequently misalign with human perception of natural language (*e.g.*, faithfulness, fluency, creativity, and semantic equivalence). More importantly, repeated testing on such static benchmarks may invite overfitting (Schaeffer, 2023; Zhou et al., 2023c; Grattafiori et al., 2024), aptly encapsulated by Goodhart's Law (Elton, 2004), suggesting that LLMs learn to game specific benchmarks rather than truly improve.

An alternative is to enlist powerful LLMs themselves as automated judges (Zhang et al., 2023b; Zeng et al., 2023), known as LLM-as-a-judge. For instance, systems such as LIMA (Zhou et al., 2023a) and AlpacaFarm (Dubois et al., 2023) leverage proprietary models like GPT-4 as the judge through API calls, while open-source evaluators, *e.g.*, PandaLM (Wang et al., 2023e), Shepherd (Wang et al., 2023d), AUTO-J (Li et al., 2023b), Prometheus (Kim et al., 2023, 2024), SaMer (Feng et al., 2025), and J1 (Whitehouse et al., 2025) have also been proposed. Although these LLM-based judges offer speed and interpretability, they also come with various biases (Zhu et al., 2023; Wang et al., 2023c; Chen et al., 2024) toward certain positions (*i.e.*, position bias), specific formats (*i.e.*, format bias), lengthy outputs (*i.e.*, verbosity bias), polished responses (*i.e.*, beauty bias), familiar content encountered during training (*i.e.*, knowledge bias), or self-generated answers (*i.e.*, self-enhancement bias). Additionally, they often struggle in specialized domains like mathematical reasoning and scientific comprehension.

Despite these advances, human evaluation remains the gold standard as natural language is created and consumed by humans. Platforms like Chatbot Arena (Chi-

[*]Equal contribution.
[†]Corresponding authors.



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10913–10947
July 27 - August 1, 2025 

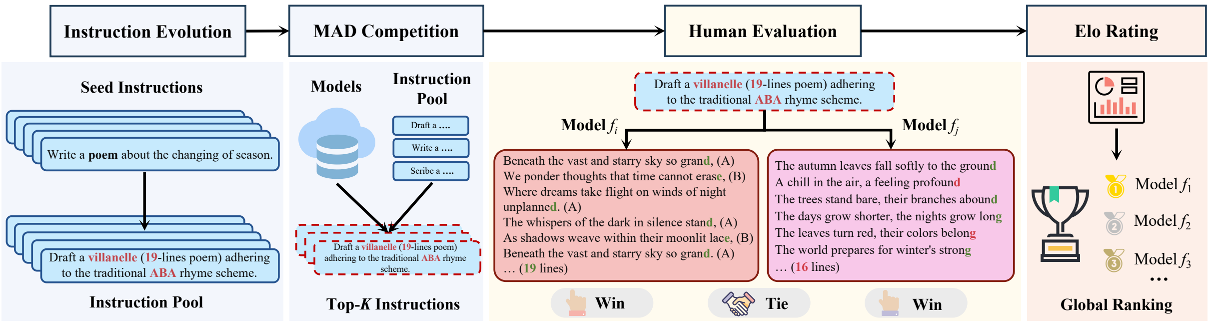



Figure 1: Overview of the proposed sample-efficient human evaluation method for comparing LLMs adaptively. Starting from a small set of task-specific seed instructions, we apply an instruction evolution procedure to generate a large-scale pool of diverse instructions. For any two competing LLMs, we then conduct MAD Competition to automatically and adaptively select the top-$K$ instructions (and their corresponding responses) that most effectively distinguish model behaviors. These selected response pairs are presented to human evaluators (along with the input instruction), who express pairwise preferences. Finally, we feed these comparison outcomes into an Elo rating system to produce a global ranking of all evaluated LLMs.

ang et al., 2024) allow crowdworkers to compare model outputs directly, but they scale poorly in terms of time and cost. This raises a fundamental question: *How can we automatically select the smallest, most informative set of instructions*[1] *from an essentially infinite pool, so that human judgments yield a definitive performance ranking of LLMs with minimal annotation effort*?

In this work, we answer this question by leveraging MAximum Discrepancy (MAD) Competition (Wang and Simoncelli, 2008) from the fields of computational vision and software testing. Starting from a large, self-generated, and unlabeled instruction pool, our method automatically identifies a minimum set of test instructions that maximize the semantic discrepancy of pairs of LLM responses (*i.e.*, where the models disagree the most), and presents only these potential counterexamples to human evaluators. By enforcing diversity among selected instructions, we ensure coverage of varied failure modes. Finally, we aggregate the resulting human preferences via Elo rating (Elo and Sloan, 1978) to a reliable global ranking. We demonstrate our sample-efficient evaluation approach across four tasks—scientific knowledge understanding, mathematical reasoning, creative and functional writing, and code generation and explanation—on eight widely used LLMs (Du et al., 2021; Ouyang et al., 2022; OpenAI, 2023; Team et al., 2023; Xu et al., 2023; Bai et al., 2023; Chiang et al., 2024) under a strict human-labeling budget.

Our contributions are threefold:

- **Sample-efficient evaluation**. We present a reliable, scalable method to automatically and adaptively select minimal yet maximally informative test instructions for LLM comparison.
- **Multi-dimensional benchmarking**. We apply our method across four distinct tasks, showing that reliable rankings emerge from only a handful of human judgments.
- **Insights into model behaviors**. By examining counterexamples that lead LLMs to fail, we reveal their relative strengths and weaknesses, offering guidance for future model improvement.

[1] In this paper, the terms "instructions," "prompts," and "queries" are used interchangeably to denote the inputs provided to LLMs.

## 2 Related Work

Evaluation of LLMs has been addressed through three approaches: standardized benchmark evaluation, automated LLM-based evaluation, and human-centric evaluation. Each paradigm offers distinct insights into model capabilities, yet also bears methodological limitations.

### 2.1 Standardized Benchmark Evaluation

A rich ecosystem of standardized benchmarks has emerged to quantify LLM performance across knowledge, instruction-following, dialogue, preference alignment, and safety dimensions. Core knowledge evaluation, such as MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023), GPQA (Rein et al., 2023), and AGIEval (Zhong et al., 2023), use elementary to professional-level examination questions to test factual recall and reasoning under multiple-choice settings. Instruction-following suites like NaturalInstructions (Mishra et al., 2021), LLMBAR (Zeng et al., 2023), and IFEval (Zhou et al., 2023b) present open-ended tasks formulated in natural language, measuring a model's ability to execute diverse directives. Conversational benchmarks—xDial-Eval (Zhang et al., 2023a), MT-Bench (Zheng et al., 2023), LongMemEval (Wu et al., 2024), and MultiChallenge (Sirdeshmukh et al., 2025)—simulate multi-turn dialogues to evaluate coherence, context retention, and response appropriateness in conversational interactions. Human preference collections such as HHH Alignment (Askell et al., 2021), PPE (Frick et al., 2024), and JudgeBench (Tan et al., 2024) gather pairwise ranking judgments to directly compare model outputs. Safety and robustness tests like AdvGLUE (Wang et al., 2021), DecodingTrust (Wang et al., 2023a), and SG-Bench (Mou et al., 2024) probe adversarial vulnerabilities and the propensity for harmful content generation.

Despite their breadth, these benchmarks often depend on objective metrics (*e.g.*, BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and BERTScore (Zhang et al., 2019)), which correlate weakly with human judgments in open-ended or specialized tasks (Novikova et al., 2017; Wei et al., 2024). Furthermore, repeated public exposure leads to data contamination and overfitting, while the static nature of fixed test sets limits adaptation to emerging use cases.

### 2.2 Automated LLM-based Evaluation

Inspired by the strong instruction-following capabilities of closed-source models such as GPT-4, several recent studies (Zeng et al., 2023; Chan et al., 2023; Zhou et al., 2023a; Zheng et al., 2023; Dubois et al., 2023) have repurposed these advanced LLMs as automated judges. In parallel, the community has begun to develop open-source alternatives to reduce reliance on black-box systems. PandaLM (Wang et al., 2023e) fine-tunes LLaMA (Touvron et al., 2023) to perform pairwise comparisons and generate brief justifications. Shepard (Wang et al., 2023d) produces critiques across diverse question-answer scenarios, while Prometheus (Kim et al., 2023) incorporates a thousand fine-grained scoring rubrics during training, enabling customization to specific evaluation criteria. AUTO-J (Li et al., 2023b) is trained on user queries paired with LLM-generated responses under fifty-eight real-world scenarios, yielding both critiques and numerical ratings. More recently, SaMer introduces a scenario-aware, multi-dimensional method that delivers both overall scores and fine-grained feedback on LLM replies (Feng et al., 2025). Despite their flexibility, automated evaluators remain susceptible to various biases, and their lack of domain-specific expertise can undermine reliability in specialized contexts (Zheng et al., 2023; Chen et al., 2024).

### 2.3 Human-Centric Evaluation

Human evaluation persists as the gold standard for judging LLMs, particularly when aligning model outputs with subjective preferences. Platforms like Chatbot Arena (Chiang et al., 2024) employ large-scale, crowdsourced "battles," in which human evaluators compare two chatbots on randomly drawn real-world instructions to produce an Elo-based ranking. While this approach can yield robust aggregation results, it does not optimize instructions to maximally differentiate competing LLMs, leading to redundant or trivial comparisons and, consequently, wasted human effort. Furthermore, without an explicit mechanism to ensure coverage of diverse failure modes, random instruction sampling risks both sampling bias and under-representation of important edge cases. Additionally, running thousands of battles at scale incurs substantial time and financial costs, and crowdsourced judgments introduce noise and variability that must be countered by still more annotations. Dynabench offers another paradigm of "adversarial" data collection, requiring users to manually submit potential counterexamples that expose model weaknesses (Kiela et al., 2021). To mitigate these challenges, our study proposes an automated, adaptive sampling strategy based on MAD Competition.

## 3 Proposed Sample-Efficient Human Evaluation Method

In this section, we present our sample-efficient human evaluation method for comparing LLMs adaptively. Building on the principle of MAD Competition (Wang and Simoncelli, 2008), our approach automatically selects a small yet highly informative subset of test instructions that maximally differentiate model behaviors. Human evaluators then perform pairwise comparisons on the selected response pairs, and we aggregate their judgments into a global ranking via Elo rating. The overall pipeline is depicted in Figure 1.

### 3.1 Problem Formulation

Let $\mathcal{X}$ denote a large, unlabeled pool of instructions, assembled from diverse, practical natural language processing tasks. We wish to compare $N$ competing LLMs, denoted as $\mathcal{F} = \{f_n\}_{n=1}^{N}$, where each $f_n$ produces an output $y_n = f_n(x)$ for any $x \in \mathcal{X}$. Human evaluators operate in a subjective assessment environment $H$, wherein they can reliably judge the relative quality of two responses for the same input. Under a strict budget on the number of human annotations, our goal is to produce a definitive, global ranking of the LLMs in $\mathcal{F}$ based on only a handful of carefully chosen comparisons.

### 3.2 MAD Competition for LLMs

Let us first consider a simple case in which we compare two LLMs $f_i$ and $f_j$. According to the principle of MAD Competition, we seek the instruction

$$\hat{x} = \underset{x \in \mathcal{X}}{\operatorname{argmax}}\, D(f_i(x), f_j(x)), \tag{1}$$

where $D(\cdot, \cdot)$ represents a distance measure that quantifies the semantic discrepancy between two responses. The comparative analysis of $f_i(\hat{x})$ versus $f_j(\hat{x})$ can produce three distinct outcomes:

- $H\left(f_i(\hat{x}), f_j(\hat{x})\right) \approx 1$. The majority of human evaluators prefer $f_i(\hat{x})$ over $f_j(\hat{x})$, making $f_i(\hat{x})$ the clear winner. The chosen $\hat{x}$ thus serves as a counterexample for $f_j$ and is highly informative for ranking the relative performance of two models.

- $H\left(f_i(\hat{x}), f_j(\hat{x})\right) \approx 0$. Conversely, $f_j$ prevails, indicating that human evaluators overwhelmingly favor $f_j(\hat{x})$ over $f_i(\hat{x})$. Again, $\hat{x}$ functions as a counterexample, this time for $f_i$, and maximally discriminates between the two models.

- $H(f_i(\hat{x}), f_j(\hat{x})) \approx 0.5$. Here, the evaluators assign similar ratings to both responses, resulting in a tie. Such ties fall into two subcategories:
  - High-rating tie. Both $f_i(\hat{x})$ and $f_j(\hat{x})$ receive strong evaluation, suggesting that each model can generate diverse yet satisfactory responses. This scenario reflects real-world task complexity in which multiple plausible outputs exist. Although $\hat{x}$ underscores each model's strengths, it contributes little to distinguishing their overall performance.
  - Low-rating tie. Both outputs garner poor scores, indicating that each model fails, albeit in different ways, to follow the instruction. In this case, $\hat{x}$ illustrates their respective weaknesses but again offers limited insight for ranking.

Selecting only the $K$ instructions with the most discrepant responses may yield a narrow set of failure cases. To encourage a broader exploration of model behaviors, we add a diversity term. When selecting the $k$-th instruction, we solve

$$\hat{x}^{(k)} = \mathrm{argmax}_{x \in \mathcal{X} \setminus \mathcal{I}} D(f_i(x), f_j(x)) + \lambda D(x, \mathcal{I}), \tag{2}$$

where $\mathcal{I} = \{\hat{x}^{(k')}\}_{k'=1}^{K-1}$ is the set of previously chosen instructions, $D(x, \mathcal{I})$ measures the maximum discrepancy between $x$ and any element in $\mathcal{I}$, and $\lambda > 0$ balances discrepancy versus diversity. Once choosing $\hat{x}^{(k)}$, we add it to $\mathcal{I}$ before proceeding to the next iteration.

By iterating this procedure for all $\binom{N}{2}$ model pairs and retaining $K$ instructions per pair, we construct a MAD response set:

$$\mathcal{R} = \{\{\{f_i(\hat{x}^{(k)}), f_j(\hat{x}^{(k)})\}_{k=1}^{K}\}_{i=1}^{N-1}\}_{j=i+1}^{N}, \tag{3}$$

whose size grows only with $O(N^2K)$, independent of the instruction pool $|\mathcal{X}|$.

### 3.3 Ranking Aggregation via Elo Rating

For each selected instruction $\hat{x}^{(k)}$ and its paired outputs $\{f_i(\hat{x}^{(k)}), f_j(\hat{x}^{(k)})\}$, human evaluators perform a three-alternative forced choice (3-AFC) task, whose outcome is recorded as

$$w = \begin{cases} 1, & \text{if } f_i \text{wins} \\ 0, & \text{if } f_j \text{wins} \\ 0.5, & \text{otherwise.} \end{cases} \tag{4}$$

Elo score (Elo and Sloan, 1978) updates at the $t$-th comparison:

$$s_i^{(t)} = s_i^{(t-1)} + \eta \left( w_{ij}^{(t)} - \frac{1}{1 + 10^{-d_{ij}^{(t-1)}}} \right) \tag{5}$$

and

$$s_j^{(t)} = s_j^{(t-1)} + \eta \left( 1 - w_{ij}^{(t)} - \frac{1}{1 + 10^{d_{ij}^{(t-1)}}} \right), \tag{6}$$

**Algorithm 1:** Sample-Efficient Human Evaluation of LLMs via MAD Competition

**Input:** An instruction pool $\mathcal{X}$, a set of competing LLMs $\mathcal{F} = \{f_n\}_{n=1}^{N}$, and a semantic discrepancy measure $D(\cdot, \cdot)$

**Output:** Global ranking scores, $s \in \mathbb{R}^{N \times 1}$

$\mathcal{R} \leftarrow \emptyset$
**for** $n \leftarrow 1$ **to** $N$ **do**
  Generate responses $\{f_n(x) | x \in \mathcal{X}\}$
**end**
**for** $i \leftarrow 1$ **to** $N-1$ **do**
  **for** $j \leftarrow i+1$ **to** $N$ **do**
    $\mathcal{I} \leftarrow \emptyset$ // $f_i$ versus $f_j$
    **for** $k \leftarrow 1$ **to** $K$ **do**
      Select $\hat{x}^{(k)} \in \mathcal{X} \setminus \mathcal{I}$ by solving Eq. (2)
      $\mathcal{I} \leftarrow \mathcal{I} \cup \hat{x}^{(k)}$
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{f_i(\hat{x}^{(k)}), f_j(\hat{x}^{(k)})\}$
    **end**
  **end**
**end**
Collect human judgments on $\mathcal{R}$ via 3-AFC
Compute $s$ via Elo rating with bootstrapping

where

$$d_{ij}^{(t-1)} = \frac{s_i^{(t-1)} - s_j^{(t-1)}}{\tau}. \tag{7}$$

$\eta$ controls the learning rate and $\tau$ sets the rating scale. $w_{ij}^{(t)}$ is the result of the $t$-th human comparison of $f_i$ and $f_j$ as defined in Eq. (4). $s^{(0)} = \{s_n^{(0)}\}_{n=1}^{N}$ are the initial ranking scores of $N$ LLMs. To mitigate the sensitivity of the online linear update to comparison order, we apply a bootstrapping procedure as suggested in (Chiang et al., 2024). Specifically, we generate a thousand bootstrap datasets by sampling with replacement from the human judgments, each dataset being the same size as the original. For each bootstrap dataset, we compute the Elo ratings and then average these ratings across all datasets to produce our final ranking. The full procedure is summarized in Algorithm 1.

### 3.4 Incorproating New LLMs

Integrating an additional LLM $f_{N+1}$ into MAD Competition is both straightforward and cost-effective. The identified MAD response set $\mathcal{R}$, along with their associated human preferences, remains unchanged. One needs only to sample a new collection of $N \times K$ instruction-response pairs to accentuate behavioral differences between $f_{N+1}$ and $\mathcal{F} = \{f_n\}_{n=1}^{N}$, obtain human preferences on these new pairs, and then update the global ranking via Eqs. (5) to (7). The complete procedure for adding a new LLM is detailed in Algorithm 2.

## 4 Experiments

In this section, we demonstrate the effectiveness of our sample-efficient human evaluation method on eight widely used LLMs across four distinct tasks. We first describe our experimental setups (Section 4.1), then present the resulting model rankings (Section 4.2), and

**Algorithm 2:** Adding a New LLM into MAD Competition

**Input:** An instruction pool $\mathcal{X}$, global ranking scores of previous $N$ LLMs $s \in \mathbb{R}^{N\times 1}$, a semantic discrepancy measure $D(\cdot,\cdot)$, and a new competing LLM $f_{N+1}$

**Output:** Global ranking scores, $\tilde{s} \in \mathbb{R}^{(N+1)\times 1}$

$\mathcal{R} \leftarrow \emptyset$
Generate responses $\{f_{N+1}(x) | x \in \mathcal{X}\}$
**for** $i \leftarrow 1$ **to** $N$ **do**
    $\mathcal{I} \leftarrow \emptyset$ // $f_i$ versus $f_{N+1}$
    **for** $k \leftarrow 1$ **to** $K$ **do**
        Select $\hat{x}^{(k)} \in \mathcal{X} \setminus \mathcal{I}$ by solving Eq. (2)
        $\mathcal{I} \leftarrow \mathcal{I} \cup \hat{x}^{(k)}$
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{f_i(\hat{x}^{(k)}), f_{N+1}(\hat{x}^{(k)})\}$
    **end**
**end**
Collect human judgments on $\mathcal{R}$ via 3-AFC
Compute $\tilde{s}$ via Elo rating with bootstrapping, starting from $s$


Figure 2: Task distribution in our experiment.

compare these rankings against established leaderboards (Section 4.3). We further evaluate the reliability and efficiency of our method against alternatives (Section 4.4), under different key hyperparameter configurations (Section 4.5), and with extended experimentation in real-world scenarios (Section 4.6).

### 4.1 Experimental Setups

**Instruction pool construction** We curate a large-scale, diverse pool $\mathcal{X}$ of 120K natural language instructions, spanning four core tasks: scientific knowledge understanding, mathematical reasoning, creative and functional writing, and code generation and explanation (see Figure 2). For each task, we begin with 3K seed instructions drawn from established datasets (*i.e.*, GSM8K (Cobbe et al., 2021), CAMEL (Li et al., 2023a), AlpacaEval (Li et al., 2023d), and CodeAlpaca (Chaudhary, 2023)) and apply ten rounds of automated instruction evolution (Xu et al., 2023) to synthesize 30K "realistic" instructions that mimic real-world human-chatbot interactions (see Appendix B for more details). This ensures both breadth across task types and depth in covering potential failure modes.

**Model selection** Under a strict human annotation budget, we evaluate eight widely used LLMs, containing three proprietary models: GPT-3.5-Turbo (Ouyang et al., 2022), GPT-4-Turbo (OpenAI, 2023) and Gemini-Pro (Team et al., 2023), and five open-source models: ChatGLM3-6B (Du et al., 2021), WizardLM-13B (Xu et al., 2023), Vicuna-13B (Chiang et al., 2023), OpenChat-3.5 (Wang et al., 2023b), and Qwen-14B-Chat (Bai et al., 2023) (see Appendix C for their implementations). These span the current spectrum of performance and parameter scales, allowing us to assess both cutting-edge and accessible systems.

**Semantic discrepancy measure** To quantify the semantic discrepancy between paired responses, we embed each using the OpenAI `text-embedding-ada-002` model and compute cosine dissimilarity (Zhang et al., 2019). This embedding-based measure to implement $D(\cdot,\cdot)$ in Eq. (2) reliably captures semantic content discrepancy, and is less prone to model-specific biases.

**Human preference collection** For each unordered model pair $(f_i, f_j)$, we select $K = 10$ instructions via MAD Competition, yielding $\binom{8}{2} \times 10 = 280$ pairwise comparisons. Thirteen STEM-trained postgraduates perform 3-AFC tasks on each paired response, indicating "win," or "tie." These results are then aggregated via Elo rating. More details are given in Appendix D.

### 4.2 Main Results

We summarize the evaluation outcomes in Table 1, reporting both overall and task-specific rankings. These results illuminate distinct performance patterns across four core tasks.

For **scientific knowledge understanding**, proprietary models—GPT-4-Turbo, GPT-3.5-Turbo, and Gemini-Pro—dominate this task, reflecting their precise command of domain concepts and robust application of scientific theorems. Remarkably, OpenChat-3.5 (with only 7B parameters) attains a higher ranking than GPT-3.5-Turbo by offering concise yet thorough explanation. In contrast, larger open-source models (*e.g.*, Vicuna-13B) tend to generate redundantly detailed responses.

For **mathematical reasoning**, our ranking results align closely with the GSM8K leaderboard (Cobbe et al., 2021), reflecting the source of our instruction pool. Differences in model outputs arise from two primary factors: (1) divergent reasoning paths and (2) arithmetic inaccuracies despite similar reasoning. First, because our generated instructions target problem types and difficulty levels typical of grade-school mathematics, models generally follow singular, concise solution paths; thus, substantial deviations in final answers correspond to noticeably different reasoning trajectories. Second,

Table 1: Global ranking results obtained by our method for eight LLMs on four different tasks.

| Model | Understanding | | Reasoning | | Writing | | Coding | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Rank | Elo Rating | Rank | Elo Rating | Rank | Elo Rating | Rank | Elo Rating | Rank | Elo Rating |
| GPT-4-Turbo | 2 | 1,065 | 1 | 1,123 | 1 | 1,162 | 1 | 1,103 | 1 | 1,132 |
| Gemini-Pro | 1 | 1,091 | 2 | 1,094 | 2 | 1,097 | 3 | 1,085 | 2 | 1,107 |
| OpenChat-3.5 | 3 | 1,047 | 3 | 1,087 | 3 | 1,025 | 4 | 971 | 3 | 1,035 |
| GPT-3.5-Turbo | 4 | 988 | 4 | 1,069 | 5 | 976 | 2 | 1,095 | 4 | 1,034 |
| WizardLM-13B | 5 | 986 | 8 | 823 | 4 | 1,001 | 6 | 961 | 5 | 937 |
| Qwen-14B-Chat | 6 | 967 | 6 | 939 | 7 | 918 | 5 | 963 | 6 | 932 |
| ChatGLM3-6B | 8 | 924 | 5 | 998 | 8 | 861 | 7 | 958 | 7 | 929 |
| Vicuna-13B | 7 | 932 | 7 | 869 | 6 | 962 | 8 | 865 | 8 | 894 |

variations in each model's computational fidelity can introduce errors in intermediate steps, so even when following analogous logical procedures, minor arithmetic mistakes may propagate to yield incorrect results. Among the models, WizardLM-13B performs relatively poorly. We attribute this to its instruction-evolution training, which uses Alpaca-52K seed data that is not specifically optimized for mathematical tasks (Taori et al., 2023). Vicuna-13B exhibits a similar limitation.

For **creative and functional writing**, the MAD-selected instructions are mostly open-ended, such as "compose a short story" or "craft a holiday recipe" that invites free-form expression. Consequently, human evaluators consistently favor models that produce longer, more richly detailed outputs over those with terser responses. For example, ChatGLM3-6B generates an average of 221.2 words per response, whereas GPT-4-Turbo delivers roughly 454.8 words. This greater verbosity not only provides more descriptive content but also often reflects deeper insights, which in turn drives higher user preference.

For **code generation and explanation**, human evaluators consider not only the functional correctness of the code but also its fidelity to the given instructions, such as respecting specified line limits, employing designated Python libraries, and conforming to the intended application contexts. We find that LLMs exhibit greater variability in code generation tasks than in code explanation, reflecting the complex interplay between problem specification and implementation. Notably, our human preference rankings correspond closely to established coding benchmarks. For instance, GPT-4-Turbo (76.83% pass@1), GPT-3.5-Turbo (74.39% pass@1), and Gemini Pro (59.76% pass@1) achieve the highest accuracies on HumanEval (Akter et al., 2023) and are likewise favored in our evaluation method. This alignment underscores the validity of our method to assess LLM performance in realistic coding scenarios.

Table 9 in the Appendix summarizes each model's strengths and weaknesses across the four evaluated tasks, yielding actionable insights for enhancing response quality. Appendix E further presents a series of case studies—most notably counterexamples to the otherwise high-performing GPT-4-Turbo—that empirically validate these observations. Moreover, the failure instances uncovered by MAD Competition and confirmed via human evaluation constitute a valuable corpus for developing more reliable LLMs, for example by integrating them into an active learning paradigm (Sinha et al., 2019).

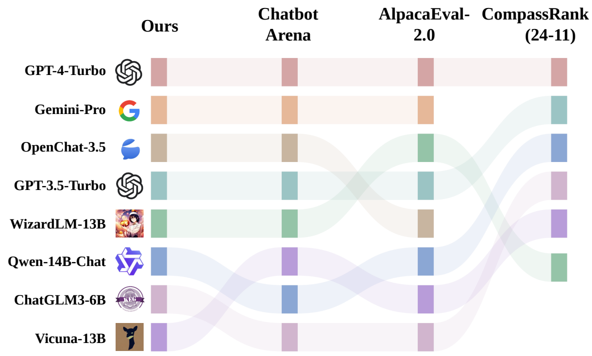



Figure 3: Sankey diagram of eight LLMs' ranking shifts across our sample-efficient human evaluation method, Chatbot Arena, AlpacaEval-2.0, and CompassRank (Nov. 2024 snapshot).

### 4.3 Comparison with Established Leaderboards

To validate the reliability of our sample-efficient human evaluation method, we compare its global rankings against three prominent LLM leaderboards: (1) *Chatbot Arena*[2], which aggregates large-scale human preference "battles" via Elo rating; (2) *AlpacaEval-2.0*[3], which leverages LLM-based judges to score open-ended instruction-following abilities; and (3) *CompassRank* (Nov. 2024 snapshot)[4], which measures performance using standard objective metrics (see Figure 3).

Chatbot Arena draws on extensive, crowdsourced pairwise human judgments, aggregating them via Elo rating to establish a "gold standard" benchmark for LLM evaluation. Remarkably, our sample-efficient approach, though engaging only thousands of model "battles," yields rankings that align almost perfectly with those from Chatbot Arena. The lone discrepancy, in Vicuna-13B's placement, is likely attributable to differences in task sampling, underscoring our method's

[2] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

[3] https://tatsu-lab.github.io/alpaca_eval/

[4] https://rank.opencompass.org.cn/leaderboard-llm/?m=24-04

Table 2: Comparison of global ranking results on *mathematical reasoning* using different sampling strategies. Cells with more saturated colors indicate more significant ranking discrepancies from the "gold standard."

| Model | Random | | KL Divergence | | Cross-Entropy | | Anchor Points | | DiffUse | | MAD Competition | | "Golden" ranking (on GSM8K) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Accuracy |
| GPT-4-Turbo | 1 | 1,028 | 2 | 1,020 | 4 | 983 | 1 | 1,057 | 1 | 1,048 | 1 | 1,157 | 1 | 92.7 |
| OpenChat-3.5 | 5 | 1,000 | 4 | 1,005 | 2 | 1,030 | 3 | 1,044 | 2 | 1,041 | 2 | 1,132 | 2 | 77.3 |
| GPT-3.5-Turbo | 2 | 1,025 | 1 | 1,036 | 3 | 1,025 | 2 | 1,037 | 3 | 1,041 | 3 | 1,079 | 3 | 74.9 |
| ChatGLM3-6B | 3 | 1,007 | 3 | 1,017 | 1 | 1,045 | 7 | 1,023 | 7 | 1,028 | 4 | 1,018 | 4 | 72.3 |
| Qwen-14B-Chat | 4 | 1,007 | 5 | 993 | 5 | 982 | 5 | 1,012 | 5 | 1,012 | 5 | 953 | 5 | 60.1 |
| Vicuna-13B | 7 | 947 | 7 | 957 | 7 | 974 | 6 | 952 | 6 | 942 | 6 | 858 | 7 | 11.3 |
| WizardLM-13B | 6 | 987 | 6 | 972 | 6 | 974 | 4 | 877 | 4 | 886 | 7 | 802 | 6 | 13.5 |

ability to faithfully reproduce large-scale human evaluation with minimal annotation effort.

In AlpacaEval 2.0, WizardLM-13B is ranked above OpenChat-3.5 and GPT-3.5-Turbo, in contrast to our human-grounded results. This discrepancy stems from AlpacaEval's emphasis on assessing LLMs' instruction-following capabilities in unconstrained, open-ended tasks, whereas WizardLM-13B has been fine-tuned on 520K diverse instructions, thus yielding a relative advantage in instruction-following tasks.

When compared to CompassRank, we observe notable shifts for Qwen-14B-Chat and ChatGLM3-6B. Although fine-tuning on MMLU and HumanEval elevates their positions on metric-driven leaderboards, these results alone fail to reflect true human preferences. This gap underscores the necessity of integrating human judgments alongside quantitative metrics to achieve a more comprehensive appraisal of LLM performance.

### 4.4 Comparison with Alternative Sampling Strategies

To assess the effectiveness of our adaptive sampling strategy based on MAD Competition, we compare it against five alternatives: (1) uniform random sampling, (2) Kullback–Leibler (KL) divergence-based sampling, (3) cross-entropy-based sampling (Boubdir et al., 2023), (4) Anchor Points (Vivek et al., 2023), and (5) DiffUse (Ashury-Tahan et al., 2024). Since KL divergence and cross-entropy require access to token log probabilities, we restrict this comparison to seven LLMs that expose such information. All LLMs are evaluated on mathematical reasoning using the GSM8K-derived instruction pool. Human evaluators are instructed to prioritize inference accuracy, and the models' performance on the original GSM8K test set serves as the "gold standard." As shown in Table 2, with only $K = 10$ MAD-selected prompts per model pair, our approach reproduces the golden ranking with minimal discrepancy. By contrast, other strategies yield notable ranking errors: KL divergence demotes GPT-4-Turbo below its true position, and cross-entropy erroneously elevates ChatGLM3-6B above stronger baselines such as GPT-3.5-Turbo. These results highlight the effectiveness of MAD Competition in promoting sample efficiency and ranking reliability.

Beyond quantitative ranking fidelity, we conduct a qualitative analysis of the top-10 instructions chosen by each method in the *creative writing* task (see Table 18 in Appendix E). KL divergence overwhelmingly concentrates on near-homogeneous prompts—nine of ten requests involve poetry—whereas cross-entropy tends to favor academic tasks (*e.g.*, paper outlines and story generation). Random sampling, by its nature, produces a noisy, unpredictable mix of task types with occasional redundancies. In contrast, MAD Competition explicitly incorporates a diversity term, ensuring that each selected instruction probes a distinct aspect of model behavior (see Table 19). This diversity-aware selection not only minimizes overlap in task categories but also exposes a broader spectrum of failure modes, thereby maximizing the informativeness of each human comparison.

Table 3: Comparison of global ranking results under different semantic discrepancy measures.

| Model | BERTScore | | GPT-4-Turbo | | Default | |
|---|---|---|---|---|---|---|
| | Rank | Elo | Rank | Elo | Rank | Elo |
| GPT-4-Turbo | 2 | 1,060 | 1 | 1,084 | 1 | 1,162 |
| Gemini-Pro | 1 | 1,061 | 2 | 1,040 | 2 | 1,097 |
| OpenChat-3.5 | 3 | 1,020 | 3 | 1,010 | 3 | 1,025 |
| WizardLM-13B | 4 | 990 | 5 | 997 | 4 | 1,001 |
| GPT-3.5-Turbo | 5 | 989 | 4 | 998 | 5 | 976 |
| Vicuna-13B | 6 | 982 | 6 | 995 | 6 | 962 |
| Qwen-14B-Chat | 7 | 951 | 7 | 974 | 7 | 918 |
| ChatGLM3-6B | 8 | 946 | 8 | 902 | 8 | 861 |

### 4.5 Ablation Studies

**Discrepacny measure sensitivity** We compare our default `text-embedding-ada-002` model-based cosine dissimilarity against BERTScore (Zhang et al., 2019) and an LLM-as-a-judge approach using GPT-4-Turbo[5] (see Table 20 in the Appendix). Despite their different design principles, all three measures yield near-identical global rankings in the *writing* task (see Table 3), demonstrating that the focus of MAD Competition on maximal response discrepancy is effectively captured by diverse semantic discrepancy estimators.

**Sample size robustnenss** We vary $K$ (the number of instructions per model pair) from 1 to 9 and compute

[5]We convert semantic similarity measures into discrepancy scores by negating their values.

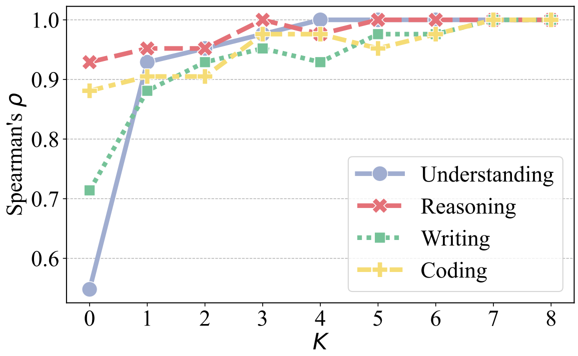



Figure 4: Spearman's $\rho$ between the global model ranking produced using the default top-10 instructions and rankings obtained with reduced prompts ($K \in \{1, \ldots, 9\}$), plotted for each of the four tasks. Correlations exceed 0.95 for $K \geq 5$ and reach 1.0 for $K \geq 8$, illustrating the robustness of our sample-efficient evaluation method even under a constrained annotation budget.

Spearman rank correlations against the default ranking with $K = 10$. Even with as few as five comparisons, correlations exceed 0.95, reaching a perfect agreement for $K \geq 8$ (see Figure 4). This indicates that reliable model rankings emerge from surprisingly small annotation budgets, and that $K = 10$ represents a practical compromise between cost and stability.

Importantly, $K$ is treated as a tunable parameter that can vary for each pair of competing LLMs. When two models demonstrate comparable performance, increasing $K$ allows for additional head-to-head comparisons and thus yields a more robust ranking. Conversely, if one model clearly outperforms the other, it is efficient to reduce $K$ (even to zero) to minimize human evaluation effort. This adaptivity parallels the flexible "battle" counts used in Chatbot Arena, where the number of matchups between any two models is not fixed but instead reflects their relative similarity.

**Diversity weight analysis** To balance instruction diversity against model discrepancy, we sweep $\lambda \in \{0, 0.5, 1.0, 1.5, 2.0\}$ in Eq. (2). With $\lambda \leq 0.5$, the selected prompts in the *writing* task cluster around a few themes (*e.g.*, poetry), yielding redundant comparisons; at $\lambda = 2.0$, diversity increases, but response discrepancy decreases, inflating "tie" outcomes. The intermediate value $\lambda = 1.0$ empirically proves optimal, producing an instruction set that preserves informative counterexamples without sacrificing discrimination power.

Taken together, these ablations confirm that our sample-efficient evaluation method is (1) insensitive to reasonable choices of semantic discrepancy measure, (2) robust under reduced annotation budgets, and (3) enhanced by a carefully tuned diversity term.

### 4.6 Further Experimentation

To assess whether our method retains its reliability as the number of compared models grows, we first expand our evaluation to 20 LLMs. As shown in Table 7 in the Appendix, even at this larger scale, the MAD-derived

Table 4: Comparison of global ranking results between our method and MATH500 accuracy.

| Model | MATH500 | | Ours | |
|---|---|---|---|---|
| | Rank | Acc | Rank | Elo |
| GPT-4.5 | 1 | 82.6 | 1 | 1,090 |
| Claude-3-5-Sonnet-20240620 | 2 | 76.3 | 2 | 1,076 |
| GPT-4o-2024-05-13 | 3 | 72.4 | 4 ↓ | 1,068 |
| GPT-4o-mini-2024-07-18 | 4 | 71.2 | 3 ↑ | 1,072 |
| GPT-4-Turbo-2024-04-09 | 5 | 68.0 | 5 | 1,067 |
| GPT-4-1106-preview | 6 | 60.8 | 6 | 1,061 |
| GPT-3.5-Turbo-1106 | 7 | 43.1 | 7 | 1,025 |
| Gemma-9b-it | 8 | 34.8 | 8 | 1,012 |
| Qwen1.5-14b-chat | 9 | 29.2 | 9 | 1,000 |
| Llama-8b-it | 10 | 27.2 | 10 | 977 |
| OpenChat-3.5 | 11 | 24.6 | 12 ↓ | 935 |
| Qwen1.5-7b-chat | 12 | 20.3 | 11 ↑ | 958 |
| Mistral-7b-it-v0.2 | 13 | 16.4 | 13 | 903 |
| ChatGLM3-6b | 14 | 16.0 | 14 | 885 |
| Vicuna-13b | 15 | 3.8 | 15 | 870 |

ranking exhibits a very strong correlation with the Chatbot Arena leaderboard (Spearman's $\rho = 0.965$ with $p = 5.93 \times 10^{-12}$), indicating that our method scales gracefully. Nevertheless, because underlying data distributions may differ between our curated instruction set and the Chatbot Arena benchmark, this single comparison may not fully attest to real-world robustness.

Consequently, we further validate our method on two external, real-world datasets: MATH500 (Lightman et al., 2023) and Chatbot Arena Conversations (Zheng et al., 2023). For MATH500, we compare model rankings by raw accuracy against those produced by our method with $K = 10$, obtaining an almost perfect concordance (Spearman's $\rho = 0.993$ with $p = 2.17 \times 10^{-13}$ in Table 4). For Chatbot Arena Conversations, we sample a subset of 15K dialogues covering fifteen open-source LLMs[6], ensuring at least 60 pairwise comparisons per model pair. Using both the full 15K subset and the 1K instances selected by our method (*i.e.*, $\binom{15}{2} \times 10 = 1,050$), we observe Spearman correlations of 0.989 and 0.986, respectively (see Table 5). By contrast, randomly drawing 1,050 comparisons (three trials with seeds 657, 216, and 849) yields a mean $\rho$ of only 0.791 (with a standard deviation of 0.069).

These results demonstrate that (1) our method maintains high fidelity even when scaling to many models, (2) a small, strategically chosen subset of comparisons suffices to replicate full-scale rankings, and (3) our adaptive sampling significantly outperforms naïve random selection in both consistency and usability.

## 5 Conclusion

We have introduced a sample-efficient method for human evaluation of LLMs grounded in the principle of MAD Competition. Unlike traditional benchmarks that

[6] https://huggingface.co/datasets/lmsys/chatbot_arena_conversations

Table 5: Comparison of global ranking results between random sampling (with 1K comparisons), 15K subset from Chatbot Arena Conversations, our method (with 1K comparisons), and the Chatbot Arena leaderboard.

| Model | Random (Seed 657) | | Random (Seed 216) | | Random (Seed 849) | | 15K Subset | | Ours | | Chatbot Arena Leaderboard | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo |
| GPT-4 | 1 | 1,119 | 1 | 1,127 | 1 | 1,120 | 1 | 1,254 | 1 | 1,123 | 1 | 1,163 |
| Claude-v1 | 2 | 1,095 | 2 | 1,090 | 2 | 1,097 | 2 | 1,196 | 2 | 1,042 | 2 | 1,149 |
| GPT-3.5-Turbo | 4 | 1,050 | 5 | 1,024 | 4 | 1,071 | 4 | 1,139 | 3 | 1,044 | 3 | 1,117 |
| Claude-Instant-v1 | 3 | 1,083 | 3 | 1,086 | 6 | 1,005 | 3 | 1,164 | 5 | 1,036 | 4 | 1,111 |
| Vicuna-13B | 8 | 978 | 4 | 1,079 | 3 | 1,073 | 5 | 1,055 | 4 | 1,042 | 5 | 1,042 |
| Vicuna-7B | 7 | 996 | 8 | 977 | 7 | 1,000 | 6 | 1,019 | 6 | 1,019 | 6 | 1,005 |
| Koala-13B | 9 | 970 | 11 | 958 | 11 | 956 | 7 | 1,006 | 7 | 1,008 | 7 | 964 |
| MPT-7B-Chat | 10 | 967 | 7 | 999 | 9 | 988 | 8 | 946 | 8 | 986 | 8 | 928 |
| RWKV-4-Raven-14B | 11 | 963 | 14 | 928 | 13 | 948 | 9 | 938 | 9 | 973 | 9 | 922 |
| Alpaca-13B | 5 | 1,046 | 13 | 946 | 15 | 907 | 11 | 906 | 11 | 966 | 10 | 901 |
| OAsst-pythia-12B | 6 | 1,011 | 6 | 1,003 | 8 | 993 | 10 | 919 | 12 | 965 | 11 | 893 |
| ChatGLM-6B | 14 | 935 | 9 | 961 | 12 | 951 | 13 | 884 | 10 | 968 | 12 | 879 |
| FastChat-T5-3B | 13 | 939 | 12 | 952 | 5 | 1,008 | 12 | 894 | 13 | 963 | 13 | 868 |
| StableLM-Tuned-Alpha-7B | 12 | 946 | 10 | 958 | 10 | 961 | 14 | 857 | 14 | 915 | 14 | 840 |
| Dolly-v2-12B | 15 | 902 | 15 | 914 | 14 | 921 | 15 | 824 | 15 | 902 | 15 | 822 |

rely on fixed, manually curated test sets, our method dynamically identifies a minimal set of highly informative instructions that maximize semantic discrepancy between model outputs. By concentrating human judgments on these targeted "hard" examples, we achieve reliable global rankings with dramatically fewer annotations. Moreover, the resulting counterexample corpus not only supports accurate model comparison but also serves as rich adversarial data for future model fine-tuning. Our approach is readily extensible to multimodal settings without modifying the core selection and aggregation procedures. Looking ahead, we plan to broaden our coverage by incorporating more LLMs, diversifying evaluation scenarios, and ultimately publishing an open, comprehensive leaderboard that can adapt as new models emerge.

## Limitations

Our current implementation performs an exhaustive (brute-force) search over the entire instruction pool to identify the top-$K$ discrepant examples for each model pair. While effective for pools on the order of $10^5$ instructions, this approach may become computationally prohibitive as pools scale to millions or when comparing large numbers of models. Future work could explore gradient-based or heuristic optimization techniques (*e.g.*, proxy models for discrepancy gradients, Bayesian optimization) to reduce search costs without sacrificing selection quality.

Although we optimize for semantic discrepancy and instruction diversity, we do not explicitly account for the variable cognitive load that different comparisons impose on human evaluators. Certain pairs of responses may be inherently harder to judge—due to subtle semantic differences, open-ended prompts, or required domain expertise—leading to increased annotation time or lower inter-rater agreement. Incorporating a computational measure of human difficulty, for example, estimating decision uncertainty via a small pilot annotation pass, could inform more balanced sample selection and adaptive budgeting of human effort.

Extending our method to dozens or hundreds of competing LLMs still entails $O(N^2K)$ human judgments, which may strain annotation budgets despite our efficiency gains. Coarse-to-fine strategies, such as seeding rankings with a strong LLM judge and then only collecting human judgments for closely ranked subsets, can partially mitigate this cost, but developing fully automated pipelines remains an open challenge.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (62301480 and 62302433), Zhejiang Provincial "Jianbing" "Lingyan" Research and Development Program of China (2025C01097 and 2024C01135), and Hangzhou West Lake Pearl Project Leading Innovative Youth Team Project (TD2023017).

## References

Syeda Nahida Akter, Zichun Yu, Aashiq Muhamed, Tianyue Ou, Alex Bäuerle, Ángel Alexander Cabrera, Krish Dholakia, Chenyan Xiong, and Graham Neubig. 2023. An in-depth look at Gemini's language abilities. *arXiv:2312.11444*.

Shir Ashury-Tahan, Ariel Gera, Benjamin Sznajder, Leshem Choshen, Liat Ein-Dor, and Eyal Shnarch. 2024. Label-efficient model selection for text generation. *arXiv:2402.07891*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas

Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv:2112.00861*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv:2108.07732*.

Amos Azaria, Rina Azoulay, and Shulamit Reches. 2024. ChatGPT is a remarkable tool—for experts. *Data Intelligence*, 6(1):240–296.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv:2309.16609*.

Meriem Boubdir, Edward Kim, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. 2023. Which prompts make the difference? Data prioritization for efficient human LLM evaluation. *arXiv:2310.14424*.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. ChatEval: Towards better LLM-based evaluators through multi-agent debate. *arXiv:2308.07201*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Sahil Chaudhary. 2023. Code Alpaca: An instruction-following LLaMA model for code generation. GitHub repository https://github.com/sahil280114/codealpaca.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? A study on judgement biases. *arXiv:2402.10669*.

Huajun Chen. 2023. Large knowledge model: Perspectives and challenges. *arXiv:2312.02706*.

Yew Ken Chia, Pengfei Hong, Lidong Bing, and Soujanya Poria. 2023. InstructEval: Towards holistic evaluation of instruction-tuned large language models. *arXiv:2306.04757*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. Blog post https://lmsys.org/blog/2023-03-30-vicuna/.

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. 2024. Chatbot Arena: An open platform for evaluating LLMs by human preference. *arXiv:2403.04132*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv:2110.14168*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. GLM: General language model pretraining with autoregressive blank infilling. *arXiv:2103.10360*.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv:2305.14387*.

Arpad E Elo and Sam Sloan. 1978. *The Rating of Chessplayers: Past and Present*. Ishi Press International.

Lewis Elton. 2004. Goodhart's law and performance indicators in higher education. *Evaluation & Research in Education*, 18(1-2):120–128.

Kehua Feng, Keyan Ding, Jing Yu, Yiwen Qu, Zhiwen Chen, Gang Yu, Qiang Zhang, Huajun Chen, et al. 2025. SaMer: A scenario-aware multi-dimensional evaluator for large language models. In *The Thirteenth International Conference on Learning Representations*.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios N Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. How to evaluate reward models for RLHF. *arXiv:2410.14872*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The LLaMA 3 herd of models. *arXiv:2407.21783*.

Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv:2310.19736*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv:2009.03300*.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Jiayi Lei, et al. 2023. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv:2305.08322*.

Hamel Husain, Ho-Hsiang Wu, Tiferet Gazit, Miltiadis Allamanis, and Marc Brockschmidt. 2019. Codesearchnet challenge: Evaluating the state of semantic code search. *arXiv:1909.09436*.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *arXiv:2011.00362*.

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. 2023. Mistral 7B. *arXiv:2310.06825*.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, et al. 2021. Dynabench: Rethinking benchmarking in NLP. *arXiv:2104.14337*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Y Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv:2405.01535*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody H Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Symposium on Operating Systems Principles*, page 611–626.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. CAMEL: Communicative agents for “mind" exploration of large language model society. *arXiv:2303.17760*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023b. Generative judge for evaluating alignment. *arXiv:2310.05470*.

Linhan Li, Huaping Zhang, Chunjin Li, Haowen You, and Wenyao Cui. 2023c. Evaluation on ChatGPT for Chinese language understanding. *Data Intelligence*, 5(4):885–903.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023d. AlpacaEval: An automatic evaluator of instruction-following models. GitHub repository `https://github.com/tatsu-lab/alpaca_eval`.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2021. Cross-task generalization via natural language crowdsourcing instructions. *arXiv:2104.08773*.

Yutao Mou, Shikun Zhang, and Wei Ye. 2024. SG-Bench: Evaluating LLM safety generalization across diverse tasks and prompt types. In *Advances in Neural Information Processing Systems*, pages 123032–123054.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Empirical Methods in Natural Language Processing*, pages 2241–2252.

OpenAI. 2023. GPT-4 technical report. *arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv:2311.12022*.

Rylan Schaeffer. 2023. Pretraining on the test set is all you need. *arXiv:2309.08632*.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. 2019. Variational adversarial active learning. In *International Conference on Computer Vision*, pages 5972–5981.

Ved Sirdeshmukh, Kaustubh Deshpande, Johannes Mols, Lifeng Jin, Ed-Yeremai Cardona, Dean Lee, Jeremy Kritz, Willow Primack, Summer Yue, and Chen Xing. 2025. MultiChallenge: A realistic multi-turn conversation evaluation benchmark challenging to frontier LLMs. *arXiv:2501.17399*.

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. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv:2206.04615*.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. JudgeBench: A benchmark for evaluating LLM-based judges. *arXiv:2410.12784*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. GitHub repository https://github.com/tatsu-lab/stanford_alpaca.

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

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. 2023. Anchor points: Benchmarking models with much fewer examples. *arXiv:2309.08638*.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023a. DecodingTrust: A comprehensive assessment of trustworthiness in GPT models. *arXiv:2306.11698*.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multitask benchmark for robustness evaluation of language models. *arXiv:2111.02840*.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023b. OpenChat: Advancing open-source language models with mixed-quality data. *arXiv:2309.11235*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023c. Large language models are not fair evaluators. *arXiv:2305.17926*.

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. 2023d. Shepherd: A critic for language model generation. *arXiv:2308.04592*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023e. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. *arXiv:2306.05087*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-Instruct: Aligning language models with self-generated instructions. *arXiv:2212.10560*.

Zhou Wang and Eero P Simoncelli. 2008. Maximum differentiation (MAD) competition: A methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):1–13.

Fangyun Wei, Xi Chen, and Lin Luo. 2024. Rethinking generative large language model evaluation for semantic comprehension. *arXiv:2403.07872*.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in LLM-as-a-judge via reinforcement learning. *arXiv:2505.10320*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. LongMemEval: Benchmarking chat assistants on long-term interactive memory. *arXiv:2410.10813*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. WizardLM: Empowering large language models to follow complex instructions. *arXiv:2304.12244*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *arXiv:2310.07641*.

Chen Zhang, Luis F. D'Haro, Chengguang Tang, Ke Shi, Guohua Tang, and Haizhou Li. 2023a. xDial-Eval: A multilingual open-domain dialogue evaluation benchmark. *arXiv:2310.08958*.

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

Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023b. Wider and deeper LLM networks are fairer LLM evaluators. *arXiv:2308.01862*.

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

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. AGIEval: A human-centric benchmark for evaluating foundation models. *arXiv:2304.06364*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023a. LIMA: Less is more for alignment. *arXiv:2305.11206*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023b. Instruction-following evaluation for large language models. *arXiv:2311.07911*.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. 2023c. Don't make your LLM an evaluation benchmark cheater. *arXiv:2311.01964*.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv:2310.17631*.

# A Elo Rating System

The Elo rating system (Elo and Sloan, 1978), devised by Arpad Elo in the 1960s, quantifies the relative skill of two competitors in games like chess or tennis. Each player holds a numerical rating that is revised after every match based on the actual result (win, loss, or tie) versus the expected outcome computed from the rating difference. Upsets—when a lower-rated player defeats a stronger opponent—yield larger point gains, whereas favored players earn fewer points for expected victories. The size of each update is governed by two parameters: the scaling factor $\tau$, which determines how rating differences translate into expected scores, and the K-factor $\eta$, which caps the maximum change per match. In our experiments, we set $\tau = 400$ and $\eta = 4$, consistent with the Chatbot Arena protocol.

# B Instruction Pool

We construct a large-scale instruction pool $\mathcal{X}$ to serve as unbiased test data for comparing LLMs. Our construction comprises three stages:

- **Task definition**. We identify four capability dimensions—scientific knowledge understanding, mathematical reasoning, creative and functional writing, and code generation and explanation—and include corresponding subtasks (see Figure 2).
- **Seed instruction collection**. For each task, we sample 3K prompts from established benchmarks:
  - *Scientific knowledge understanding* is a task to evaluate the scientific knowledge comprehension and application abilities of LLMs. Questions from CAMEL (Li et al., 2023a) cover mathematics, physics, chemistry, biology, and computer science.
  - *Mathematical reasoning* is a commonly used task to assess the math problem-solving abilities of LLMs. Problems are sourced from GSM8K (Cobbe et al., 2021).
  - *Creative and functional writing* engages in open-ended writing, aiming to satisfy user requirements. Tasks are from AlpacaEval (Li et al., 2023d) and IMPACT (Chia et al., 2023).
  - *Code generation and explanation* aims to generate high-quality code snippets based on given instructions. Coding tasks are from CodeSearchNet (Husain et al., 2019), MBPP (Austin et al., 2021), and CodeAlpaca (Chaudhary, 2023).
- **Instruction evolution**. Leveraging three strong closed-source models (GPT-4-Turbo, GPT-3.5-Turbo, and Gemini Pro), we perform ten iterative evolutions per seed (Xu et al., 2023). In each iteration, we prompt models to (i) adapt seed instructions toward more practical or nuanced scenarios (*e.g.*, transforming "a mundane text abbreviation" into "design a mnemonic to aid in memorizing a complex algorithm") and (ii) impose varied constraints (*e.g.*, "limit code to 15 lines," "compose a $1,500$-word article," or "write in Shakespearean style"). For interpretability in human evaluation, evolution prompts for understanding, reasoning, and coding also request exemplar answers.

This process yields 30K evolved instructions per task (120K in total), ensuring both diversity and real-world relevance. Figure 2 illustrates the final task distribution, and Tables 10 to 13 list our default evolution prompts.

# C Competing LLMs

We apply our sample-efficient evaluation method to eight representative LLMs:

- **GPT-4-Turbo** (*i.e.*, GPT-4-1106-preview) and **GPT-3.5-Turbo** (*i.e.*, GPT-3.5-Turbo-1106) are OpenAI's most advanced proprietary models at the time.
- **Gemini-Pro** (*i.e.*, Gemini-1.0-Pro) (Team et al., 2023) is Google's multimodal closed-source model, trained jointly on diverse, high-quality multimodal data, demonstrating strong understanding and reasoning capabilities across a variety of specialized domains.
- **OpenChat-3.5** (Ouyang et al., 2022) is a 7B-parameter derivative of Mistral-7B (Jiang et al., 2023) that applies C-RLFT (Wang et al., 2023b) to mixed-quality ShareGPT dialogues (70K in total, including 6K generated by GPT-4) for fine-tuning.
- **WizardLM-13B** (*i.e.*, WizardLM-13B-V1.2) (Xu et al., 2023) is built on LLaMA2-13B (Touvron et al., 2023) and refined via the instruction evolution method *Evol-Instruct*, which expands the 52K seed instructions in Alpaca (Taori et al., 2023) to a 520K-instruction corpus.
- **Vicuna-13B** (*i.e.*, Vicuna-13B-V1.5) (Chiang et al., 2023) is a 13B-parameter model, fine-tuned from LLaMA2-13B (Touvron et al., 2023) on real human-machine dialogue data from ShareGPT.
- **Qwen-14B-Chat** (Bai et al., 2023) is a 14B-parameter model, fine-tuned from Qwen-14B-base that is pretrained on a large-scale, diverse dataset of over three trillion tokens, covering multiple languages such as Chinese and English.
- **ChatGLM3-6B** (Du et al., 2021) is a 6B-parameter model originated from ChatGLM3-6B-Base, pretrained on over one trillion data.

**Implementation** We conduct all experiments in a zero-shot setting. Proprietary models (GPT-4-Turbo, GPT-3.5-Turbo, and Gemini-Pro) are accessed via their official APIs with the temperature set to $0.7$, top-$p$ to

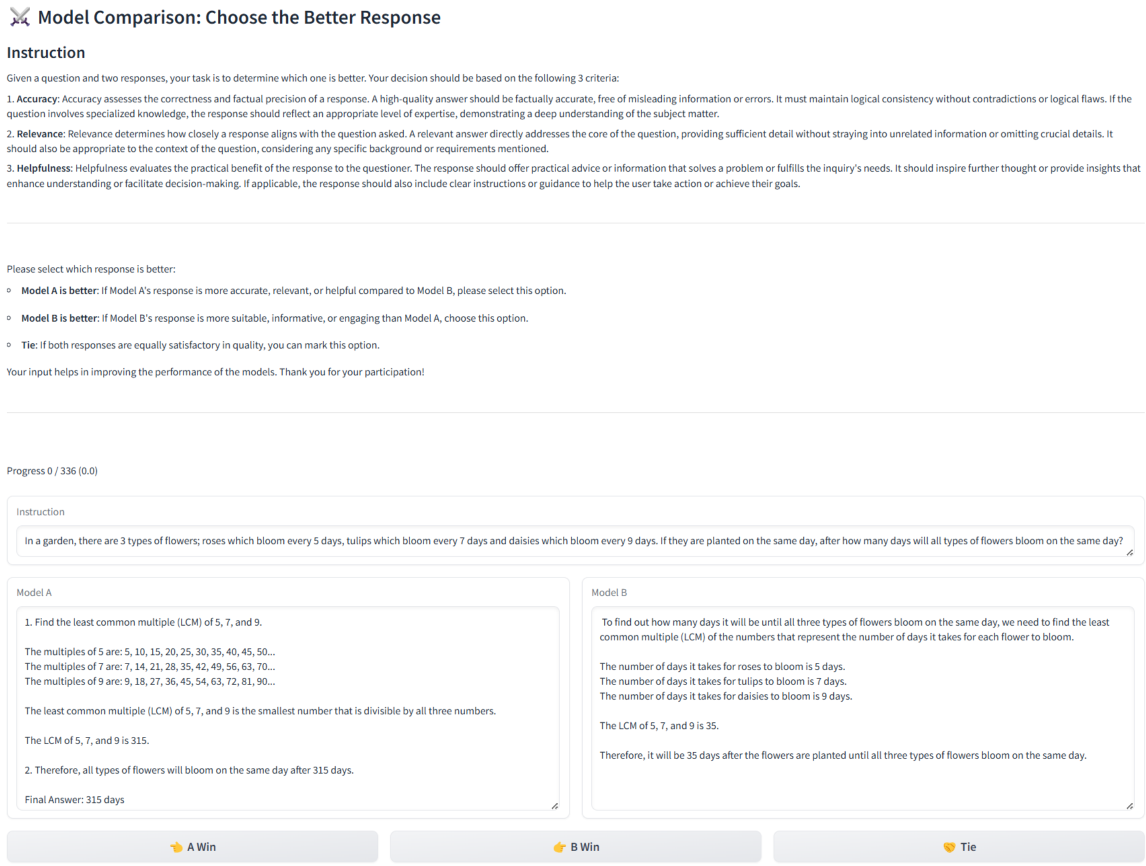



Figure 5: Graphical user interface for collecting human preferences.

1.0, and a maximum sequence length of $2,048$, leaving all other parameters at their defaults. The five open-source models are deployed locally on two NVIDIA GeForce RTX 4090 GPUs and accelerated using the vLLM method (Kwon et al., 2023), with the same inference settings, except that Qwen-14B-Chat is limited to a maximum length of $1,024$ tokens.

# D Details of Human Evaluation

This section provides a detailed overview of our human evaluation studies.

## D.1 Evaluator Selection Criteria

All evaluators have graduate-level STEM training and meet the following academic and technical proficiency criteria:

- **Language proficiency**. Each evaluator meets at least one of the following requirements:
  - Native-level English proficiency.
  - National College Entrance Examination (NCEE) English score $\geq$ 125 and College English Test (CET-6) score $\geq$ 500.
- **Disciplinary foundation**. They shall possess:
  - High-school–level mastery of mathematics, physics, chemistry, biology, and formal logic.
  - General knowledge of computer science.
  - Proficiency in Python at a professional level.
- **Sustained concentration**. Evaluators commit to offline sessions of at least two hours (with breaks in between), ensuring the accuracy and efficiency of their annotations.

## D.2 Pre-Experiment Briefing

Prior to participation, each annotator receives a comprehensive briefing on the study's objectives and procedures. Participants are explicitly informed that their annotation outputs would form part of the research dataset and that their continued involvement constitutes voluntary consent.

All annotation data are treated as strictly confidential and used exclusively for scientific analysis. No personally identifiable information is collected, stored, or disclosed, ensuring that participation entails no risk or adverse consequences for the annotators.

## D.3 Graphical User Interface

Figure 5 illustrates the interface employed to elicit human preference judgments. Initially, annotators review the task instructions, which (1) describe the core task—choosing the better of two model-generated responses—and (2) define the evaluation criteria (accuracy, relevance, and helpfulness). During the experiment, each

Table 6: Comparison of global ranking results between LLM-based and human evaluation across four tasks.

| Model | Understanding | | Reasoning | | Writing | | Coding | | Overall | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Human Rank | GPT-4o Rank | Human Rank | GPT-4o Rank | Human Rank | GPT-4o Rank | Human Rank | GPT-4o Rank | Human Rank | GPT-4o Rank |
| GPT-4-Turbo | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Gemini-Pro | 1 | 1 | 2 | 3 | 2 | 2 | 3 | 3 | 2 | 2 |
| OpenChat-3.5 | 3 | 3 | 3 | 2 | 3 | 3 | 4 | 5 | 3 | 3 |
| GPT-3.5-Turbo | 4 | 4 | 4 | 4 | 5 | 4 | 2 | 2 | 4 | 4 |
| WizardLM-13B | 5 | 5 | 8 | 8 | 4 | 5 | 6 | 6 | 5 | 5 |
| Qwen-14B-Chat | 6 | 6 | 6 | 6 | 7 | 7 | 5 | 4 | 6 | 6 |
| ChatGLM3-6B | 8 | 7 | 5 | 5 | 8 | 8 | 7 | 8 | 7 | 8 |
| Vicuna-13B | 7 | 8 | 7 | 7 | 6 | 6 | 8 | 7 | 8 | 7 |
| Spearman's $\rho$ | 0.9762 | | 0.9762 | | 0.9762 | | 0.9524 | | 0.9762 | |

annotator examines the two candidate responses alongside the instruction and then records her/his decision by clicking one of three buttons at the bottom of the page: "A win," "Tie," or "B win."

### D.4 Human Evaluation Process

To evaluate model performance, we first identify the ten most discriminative instructions for pairwise comparisons among eight models across four distinct tasks. This selection yields $4 \times \binom{8}{2} \times 10 = 1,120$ comparisons. Thirteen graduate students with strong STEM backgrounds are recruited to annotate these pairs, each of which receives annotations from at least five different students, so that, on average, each annotator assesses approximately 345 pairs. The annotation phase lasts about one week. We measure the annotation agreement and observe an average inter-annotator agreement of $83.39\%$. In cases of divergent labels, the final consensus is determined by majority vote.

## E More Experimental Results

### E.1 LLM-based Evaluation

The proposed sample-efficient evaluation method for LLMs substantially reduces human effort but does not eliminate it, which hampers scalability as the number of models grows. To overcome this bottleneck, we substitute human judges with specialized LLM-based evaluators guided by carefully crafted prompts (Tables 14 to 17). As shown in Table 6, the Spearman's $\rho$ between LLM (in particular, GPT-4o-2024-08-26) and human preferences exceeds $0.95$, confirming that top-performing LLMs (provided that they are excluded from MAD Competition) can reliably emulate human judgments. Exploiting this finding, we then apply our LLM-based method to twenty state-of-the-art LLMs (see Table 7). The resulting ranking exhibits strong agreement with that obtained via the labor-intensive Chatbot Arena, demonstrating that our approach can scale to large model sets with minimal additional labor and significantly reduced time costs.

### E.2 Sampling Algorithm Comparison

In Section 4.4, we evaluate our adaptive sampling method based on MAD Competition against three alternatives in the *reasoning* (Table 2) and *writing* (Table 8) tasks. Below is a concise summary of each baseline and its configuration:

- **DiffUse** (Ashury-Tahan et al., 2024)
  - **Method**: Cluster the embedding-difference vectors between each pair of model responses; estimate the expected ranking by drawing instructions from each cluster.
  - **Setting**: 10 clusters per model pair; 3 prompts per cluster $\rightarrow \binom{7}{2} \times 10 \times 3 = 630$ instructions.
- **Anchor Points** (Vivek et al., 2023)
  - **Method**: Apply $K$-Medoids to select a small set of "anchor" instructions.
  - **Setting**: $K = 10$ anchors per model pair $\rightarrow \binom{7}{2} \times 10 = 210$ instructions.
- **KL and Cross-Entropy Sampling** (Boubdir et al., 2023)
  - **Method**: Identical to MAD Competition's protocol except that semantic discrepancy is measured by KL divergence or cross-entropy on token log probabilities.
  - **Setting**: 10 prompts per model pair $\rightarrow \binom{7}{2} \times 10 = 210$ instructions.

Despite their differing sample budgets, all alternatives fall short of MAD Competition's accuracy in recovering the golden ranking (Tables 2 and 8). We attribute their failures to two key limitations: inefficient sampling and biased/uninformative discrepancy measures. DiffUse demands a relatively large sample count simply to approximate its cluster-based expectation, and performance degrades sharply if fewer samples are used. The Anchor Points method likewise needs far more than 10 anchors per pair to represent complex response distributions adequately. Moreover, its medoid selection biases toward dataset geometry rather than evaluation

Table 7: Global ranking results of twenty LLMs by our LLM-based evaluation method. The gray column denotes the Chatbot Arena leaderboard positions, and the two rankings exhibit strong agreement (Spearman's $\rho = 0.965$).

| Model | Understanding | | Reasoning | | Writing | | Coding | | Overall | | Chatbot Arena |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank |
| GPT-4o-2024-05-13 | 1 | 1,166 | 7 | 1,065 | 1 | 1,106 | 3 | 1,097 | 1 | 1,151 | 1 |
| GPT-4o-mini-2024-07-18 | 5 | 1,126 | 6 | 1,078 | 3 | 1,096 | 6 | 1,077 | 3 | 1,144 | 2 |
| Claude-3.5-Sonnet-20240620 | 4 | 1,139 | 1 | 1,105 | 6 | 1,054 | 2 | 1,110 | 4 | 1,142 | 3 |
| Gemini-1.5-pro-latest | 6 | 1,110 | 4 | 1,088 | 7 | 1,052 | 9 | 1,044 | 5 | 1,112 | 4 |
| GPT-4-Turbo-2024-04-09 | 2 | 1,164 | 5 | 1,078 | 2 | 1,104 | 1 | 1,110 | 2 | 1,147 | 5 |
| GPT-4-1106-preview | 3 | 1,144 | 2 | 1,103 | 4 | 1,088 | 5 | 1,080 | 6 | 1,100 | 6 |
| Claude-3-Sonnet-20240229 | 7 | 1,055 | 3 | 1,103 | 12 | 1,022 | 4 | 1,097 | 8 | 1,077 | 7 |
| Gemma2-9B-it | 8 | 1,035 | 8 | 1,030 | 8 | 1,049 | 7 | 1,058 | 9 | 1,069 | 8 |
| Llama3.1-8B-it | 10 | 978 | 10 | 1,009 | 5 | 1,070 | 11 | 1,014 | 7 | 1,079 | 9 |
| Llama3-8B-it | 18 | 878 | 15 | 951 | 11 | 1,023 | 14 | 964 | 13 | 1,001 | 10 |
| Gemini-pro | 15 | 910 | 11 | 1,006 | 13 | 1,014 | 8 | 1,048 | 11 | 1,021 | 11 |
| Qwen1.5-14B-Chat | 11 | 973 | 12 | 1,000 | 10 | 1,026 | 12 | 998 | 10 | 1,030 | 12 |
| OpenChat-3.5 | 12 | 959 | 9 | 1,016 | 16 | 930 | 16 | 911 | 16 | 939 | 13 |
| Mistral-7B-it | 16 | 910 | 16 | 941 | 9 | 1,032 | 15 | 930 | 12 | 1,005 | 14 |
| Qwen1.5-7B-Chat | 9 | 1,016 | 13 | 983 | 14 | 975 | 13 | 977 | 14 | 969 | 15 |
| GPT-3.5-Turbo-1106 | 13 | 959 | 14 | 977 | 15 | 952 | 10 | 1,017 | 15 | 942 | 16 |
| Wizardlm-13B | 17 | 909 | 20 | 832 | 17 | 906 | 18 | 881 | 17 | 815 | 17 |
| Vicuna-13B | 19 | 871 | 19 | 851 | 18 | 889 | 19 | 865 | 18 | 804 | 18 |
| Qwen-14B-Chat | 14 | 924 | 17 | 924 | 19 | 836 | 17 | 909 | 19 | 786 | 19 |
| ChatGLM3-6B | 20 | 773 | 18 | 859 | 20 | 777 | 20 | 814 | 20 | 670 | 20 |

Table 8: Comparison of global ranking results using different sampling algorithms in the *writing* task.

| Model | Random | | KL Divergence | | Cross-Entropy | | Anchor Points | | DiffUse | | MAD Competition | | "Golden" ranking (Chatbot Arena) | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Elo | Rank | Accuracy |
| GPT-4-Turbo | 1 | 1,080 | 1 | 1,075 | 1 | 1,046 | 1 | 1,127 | 1 | 1,129 | 1 | 1,086 | 1 | 1,250 |
| OpenChat-3.5 | 3 | 993 | 4 | 1,005 | 4 | 1,026 | 3 | 1,011 | 3 | 1,028 | 2 | 1,028 | 2 | 1,091 |
| WizardLM-13B | 5 | 985 | 5 | 988 | 5 | 995 | 2 | 1,037 | 4 | 989 | 3 | 1,022 | 3 | 1,068 |
| GPT-3.5-Turbo | 4 | 988 | 2 | 1,033 | 2 | 1,035 | 7 | 941 | 5 | 942 | 4 | 1,010 | 4 | 1,059 |
| Vicuna-13B | 6 | 983 | 6 | 974 | 6 | 942 | 6 | 944 | 7 | 931 | 5 | 990 | 5 | 1,042 |
| Qwen-14B-Chat | 2 | 1,038 | 3 | 1,007 | 3 | 1,030 | 4 | 993 | 2 | 1,044 | 6 | 954 | 6 | 1,035 |
| ChatGLM3-6B | 7 | 932 | 7 | 919 | 7 | 925 | 5 | 946 | 6 | 938 | 7 | 910 | 7 | 955 |

relevance. KL divergence and cross-entropy on token log probabilities do not reliably reflect substantive differences in response quality and thus yield rankings no better than random sampling. In contrast, MAD Competition achieves high-fidelity ranking estimates with a minimal, unbiased sample set, demonstrating both statistical efficiency and robustness across tasks.

### E.3 Pairwise Comparison Results

Figure 6 presents the pairwise comparison results. Across both the overall assessment and each individual task, GPT-4-Turbo and Gemini-Pro emerge as the top two models, outperforming all other competitors by a substantial margin.

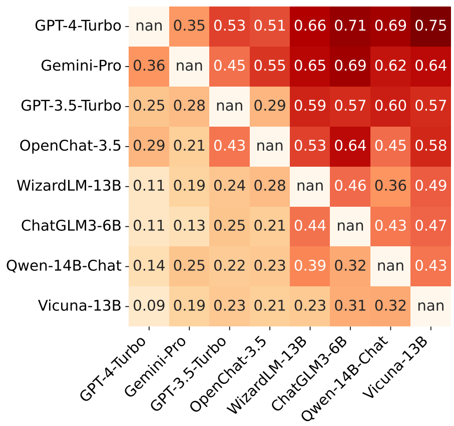
### E.4 Case Studies

In this subsection, we present illustrative cases that corroborate the trends summarized in Table 9.

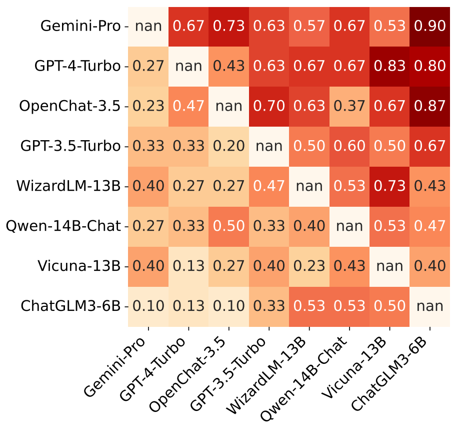
**Scientific knowledge understanding** Table 21 compares proprietary and open-source models on domain-specific questions. Proprietary systems consistently outperform open-source counterparts, reflecting their larger knowledge repositories and more effective retrieval mechanisms. In Table 22, both OpenChat-3.5 and GPT-3.5-Turbo correctly identify the key scientific facts, yet humans prefer OpenChat-3.5 because it delivers richer and deeper analytical commentary.

**Mathematical reasoning** Table 23 highlights how divergent reasoning paths between WizardLM-13B and OpenChat-3.5 can yield different answers on simple arithmetic tasks. This variability underscores WizardLM's relative deficit in systematic, step-by-step deduction. Table 24 further reveals that minor lapses in arithmetic precision by WizardLM-13B lead to incorrect final results, indicating a need to strengthen its core computational routines.

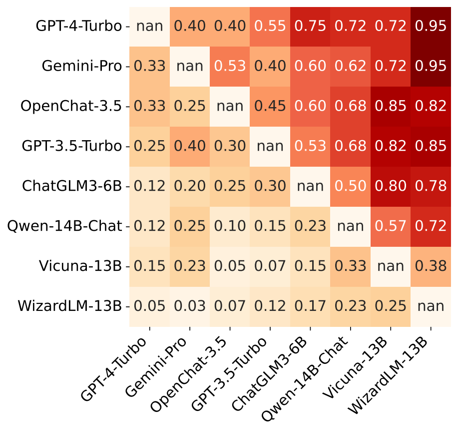
**Creative and functional writing** Table 25 demonstrates clear human preferences for responses that incorporate substantive content (*e.g.*, illustrative examples



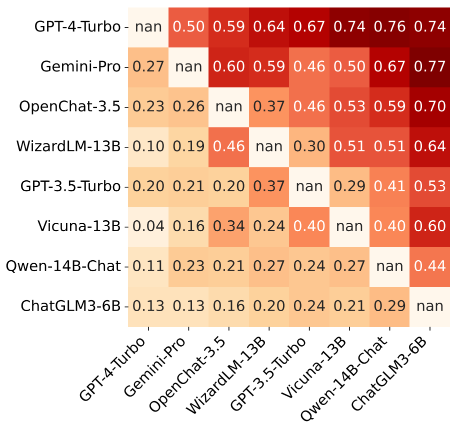

(a) Overall



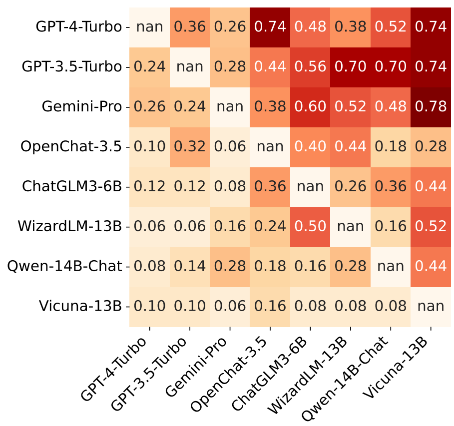

(b) Scientific knowledge understanding




(c) Mathematical reasoning




(d) Creative and functional writing




(e) Code generation and explanation

Figure 6: Heatmaps of pairwise win rates between model pairs for the overall evaluation and four individual tasks. Each cell shows the proportion of head-to-head victories for the model on the vertical axis against the model on the horizontal axis; darker red indicates a higher win rate. Note that tied outcomes do not contribute to either direction.

or nuanced argumentations), suggesting that response length correlates positively with perceived quality.

**Code generation and explanation** Although evaluators consider readability and adherence to instructions, accuracy remains paramount. In Table 26, Vicuna-13B annotates its code thoroughly, but human raters prefer Gemini-Pro because it alone delivers reliably correct, executable solutions within the specified constraints.

**Counterexamples of GPT-4-Turbo** Despite its top ranking overall, GPT-4-Turbo exhibits notable shortcomings in certain contexts:

- **Algorithm explanation** (Table 27). OpenChat-

Table 9: Some strengths and weaknesses of LLMs discovered by our sample-efficient evaluation method.

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

3.5's inclusion of a complete Dijkstra implementation renders its explanation more intuitive and actionable for human readers, suggesting that model performance should be judged not only by correctness but also by pedagogical clarity.

- **Instruction comprehension** (Tables 28 and 30). GPT-4-Turbo occasionally misinterprets prompts—for example, labeling Rosalind Franklin's contributions as "underappreciated," contrary to historical consensus—indicating room to improve contextual sensitivity and factual alignment.
- **Code constraint** (Table 29). GPT-4-Turbo sometimes generates solutions that exceed prescribed length limits or contain subtle logical errors, highlighting persistent challenges in enforcing user-specified coding guidelines.
- **Instruction adherence** (Table 31). At times, GPT-4-Turbo omits direct answers or skims over critical request elements, a phenomenon we describe as "response laziness," underscoring the importance of robust instruction-following mechanisms.

These counterexamples illustrate that, although GPT-4-Turbo leads on average, future high-performance LLMs must dynamically adapt response formats to specific tasks, enhance fine-grained comprehension, and ensure both factual and procedural fidelity.

Table 10: Instruction evolution prompt for *scientific knowledge understanding*.

You are a brilliant assistant. Your goal is to draw inspiration from #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 by humans.
Your response should include 'new_prompt' and 'answer' in the 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 #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 #Created Prompt# should be similar to that of #Given Prompt#.

#Given Prompt#:
{instruction}

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

Table 11: Instruction evolution prompt for *mathematical 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 by humans.
Your response should include 'question' and 'answer' in the following format:
{{
"question": "The new question for math reasoning. Try your best to follow the same difficulty level and a 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 #Given Prompt#.
The question in #Created Prompt# should focus on grade school math problems.
The LENGTH and difficulty level of #Created Prompt# should be similar to that of #Given Prompt#.

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

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

Table 12: Instruction evolution prompt for *creative and functional writing*.

You are a brilliant assistant. Your goal is to draw inspiration from #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 by humans.
You should design a more rare scenerio or topic that are totally DIFFERENT from #Given Prompt# but has practical significance.
The LENGTH and difficulty level of #Created Prompt# should be extremely similar to that of #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 13: 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 by humans.
Your response should include 'new_prompt' and 'answer' in the following format:
{{
"new_prompt": "The rewritten Python prompt with two constraints.",
"answer": "The Python code and brief code explanation to the rewritten prompt. Ten to Thirty lines are recommended."
}}
Output in JSON.

The rewritten prompt should be different from the scenarios in #Given Prompt#. You should add two of the following constraints to the rewritten prompt:
1. Limit the lines of the code, e.g., 'Write a 10-line Python code'. Ten to thirty lines are recommended and preferred.
2. Require high efficiency (time or space) of the code, 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. Limit the complexity of the code, e.g., 'Write a Python code ... easy to read for freshman'.

The rewritten prompt can only add ten to twenty words into #Given Prompt#. You should try your best not to make the new prompt become verbose.

#Given Prompt#:
{instruction}

#Response#:
Now, choose two constraints and create a new prompt:

Table 14: Prompt used by LLM-based evaluators for the *understanding* task.

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You are given a reference answer to the user question. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the following factors:

1. **Accuracy**: Whether the answer is correct based on the reference answer.
2. **Core Knowledge Coverage**: Whether the answer covers the core knowledge points of the question.
3. **Logical Consistency**: Whether the answer is logically clear and presents the arguments coherently.
4. **Clarity of Expression**: Whether the answer uses precise language and is easy to understand.
5. **Relevance**: Whether the answer stays focused on the question without deviating from the topic.

If both assistants provide correct answers that cover the core knowledge points, it should be considered a tie (C), regardless of the length or detail of the responses. Only choose a winner if one assistant's response is clearly superior in terms of the evaluation factors.

Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses are presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible. Output "A" if assistant A is better, "B" if assistant B is better, and "C" stands for a tie. Output your final verdict by strictly following this format:
{
"explanation": "The short explanation of your evaluation.",
"winner": "A, B or C."
}
Output this format in JSON.

[The Start of User Question]
{question}
[The End of User Question]

[The Start of Reference Answer]
{answer}
[The End of Reference Answer]

[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Table 15: Prompt used by LLM-based evaluators for the *reasoning* task.

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You are given a reference answer to the user question. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the following factors:

1. **Accuracy**: Whether the answer is correct based on the reference answer.
2. **Logical Consistency**: Whether the answer is logically clear and presents the arguments coherently.
3. **Clarity of Expression**: Whether the answer uses precise language and is easy to understand.

If both assistants provide the same final answer, it should be considered a tie (C), regardless of the length or detail of the responses. Only choose a winner if one assistant's response is clearly superior in terms of the evaluation factors, e.g., one achieves the correct answer while the other fails.

Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses are presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible. Output "A" if assistant A is better, "B" if assistant B is better, and "C" stands for a tie. Output your final verdict by strictly following this format:
{
"explanation": "The short explanation of your evaluation.",
"winner": "A, B or C."
}
Output this format in JSON.

[The Start of User Question]
{question}
[The End of User Question]

[The Start of Reference Answer]:
{answer}
[The End of Reference Answer]

[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Table 16: Prompt used by LLM-based evaluators for the *writing* task.

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the following factors:

1. **Accuracy**: How well the response matches the user's instructions and stays on topic.
2. **Creativity**: The uniqueness of the perspective and the ability to engage the reader.
3. **Logic and Structure**: Clarity of structure with a clear beginning and ending, and logical flow of ideas.
4. **Language Expression**: Clarity of language, richness of vocabulary, and appropriate use of rhetorical devices.
5. **Detail and Depth**: Provision of sufficient details to support main points and depth of exploration of the topic.

Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses are presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible. Output "A" if assistant A is better, "B" if assistant B is better, and "C" stands for a tie. A tie should be determined based on the following criteria:

1. **Similar Quality**: When the two responses are close in performance without any significant difference. Performance is measured by quality, and is not related to the length of the responses.
2. **Similar Core Content Match**: When the performance of both responses is close to meeting the main requirements of the user's instructions.
3. **Complementary Strengths**: When each response excels (or fails) in some different aspects, but their overall quality is comparable.

DO NOT make the judgment too strict. You can output "C" just because they are similar.
Output your final verdict by strictly following this format:
{
"explanation": "The short explanation of your evaluation.",
"winner": "A, B or C."
}
Output this format in JSON.

[The Start of User Question]
{question}
[The End of User Question]

[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Table 17: Prompt used by LLM-based evaluators for the *coding* task.

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the following factors:

1. **Accuracy**: Whether the code correctly implements the user's requirements or the explanation is accurate.
2. **Level of Detail**: Whether the code or explanation is detailed enough to meet the user's needs.
3. **Logical Consistency**: Whether the code structure or explanation logic is clear and consistent.
4. **Code Quality**: The quality of the code, including readability, efficiency, and maintainability.
5. **Creativity and Reasonableness**: Whether the code implementation or explanation shows creativity and is reasonable.

Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses are presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible. Output "A" if assistant A is better, "B" if assistant B is better, and "C" stands for a tie. A tie should be determined based on the following criteria:

- If both assistants' responses are good across all criteria without significant quality differences, it should be judged as a tie.
- If both responses have significant errors (despite that the errors may be different), it should be judged as a tie.
- If both responses have no significant differences in core accuracy and logical consistency, even if there are slight differences in detail or code quality, it should be judged as a tie.
- If both responses follow different but reasonable, effective approaches, it should be judged as a tie.

Output your final verdict by strictly following this format:
{
"explanation": "The short explanation of your evaluation.",
"winner": "A, B or C."
}
Output this format in JSON.

[The Start of User Question]
{question}
[The End of User Question]

[The Start of Assistant A's Answer]
{response_1}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response_2}
[The End of Assistant B's Answer]

Table 18: Top-10 instructions chosen by different sampling strategies.

| Order | Random | KL Divergence | Cross-Entropy | MAD Competition |
|---|---|---|---|---|
| 1 | You are asked to offer valuable **suggestions**, relevant statistics, or elucidation on the issue outlined below... | Craft a **limerick** centered around a family embarking on a submarine voyage to explore the Mariana Trench, capturing the whimsical tone of Lewis Carroll. | Compose four **essays** 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 **address the following question**... |
| 2 | Compose a compelling **essay** on the importance of learning a second language. | Can you compose **limericks** for various renewable energy sources? Begin with solar power. | Develop a Market **Analysis** Report, following APA referencing style, with six sections evaluating a different emerging technology... | ... **Suggest** a few alternative historical items that could temporarily fill this space and enhance the exhibition's narrative. |
| 3 | You have been assigned to **explore** the environmental implications of a modern dilemma... | Compose a **limerick** that uses alliteration to add humor to the described situation.<br>A clumsy chef spills the soup. | Craft a 1900-word science fiction **story** set in an underwater city. | Craft a one-page **essay** discussing the importance of preserving linguistic diversity in the age of global communication. |
| 4 | Draft a dialogue for a historical fiction **novella**. | Craft a **poem** 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 **adapting** the discovery of penicillin into an interactive escape room game. |
| 5 | Craft a 500-word **essay** on the significance of preserving historical languages that are on the verge of extinction. | Compose a **poem** in the voice of Edgar Allan Poe. | Craft a 1900-word adventure **story** set in the midst of a perilous journey across the Sahara Desert in the early 20th century. | Develop a compelling and nuanced **backstory** for my protagonist, a cunning diplomat in a fantasy realm... |
| 6 | Craft a **tale** about an ethnomusicologist who discovers a remote village where the people communicate using a complex system of rhythm and drumming. | Compose a **quatrain** 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 **narrative** situated in a secluded mountain monastery. | Draft a **villanelle** utilizing the traditional ABA ABA ABA ABA ABA ABAA rhyme scheme. |
| 7 | Your assignment is to showcase your proficiency in effective and engaging **marketing writing** as described below... | Compose a **sonnet** about a group of friends on a hiking adventure in the Swiss Alps... | Write four unique **essays** 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 **essay** examining the difficulties faced when introducing a new public healthcare system in a developing country. | Compose a **sonnet** following the Shakespearean rhyme pattern. | Compose four distinct **essays** analyzing the influence of climate change on different agricultural practices... | Greetings, could you help me formulate my introduction **speech** for the local art gallery opening? |
| 9 | You are assigned to **explore** the following environmental issue by detailing arguments for various viewpoints... | Craft a **sonnet** 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 **essay**... | You have been selected to demonstrate your expertise in critical **analysis** for the situation described below... |
| 10 | ...Would you be able to supply me with a detailed **proposal**? | Demonstrate your creative **narrative** skills with this task... | Draft a 2000 word **analysis** on the influence of community-based recycling programs on reducing municipal waste. | Compose a brief **narrative** beginning with the provided opening line... |

Table 19: Top-10 instructions chosen by MAD Competition with and without the diversity term.

| Order | **without** Diversity | **with** Diversity |
|---|---|---|
| 1 | You have been assigned to evaluate the following technological issue by considering opinions from diverse standpoints.<br>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.<br>Is the widespread implementation of autonomous vehicles on public roads justifiable? |
| 2 | Devise a catchy mnemonic to remember the key elements.<br>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.<br>An astronomical method for categorizing, observing, and explaining Variable Star Observation Parameters (VSOP). |
| 4 | Draft a villanelle utilizing the traditional ABA ABA ABA ABA ABA ABAA rhyme scheme. | Draft a villanelle utilizing the traditional ABA ABA ABA ABA ABA ABAA rhyme scheme. |
| 5 | Compose a sonnet 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 sonnets 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 haikus 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 20: Prompt used by GPT-4-Turbo as a semantic similarity measure.

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 tasks or themes are not entirely the same, analyze their 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 the language expression between the two responses, including language style, vocabulary, syntax, rhetorical devices, etc.
You should write an 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 themes 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 the same, their themes have few differences, but are 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 THE 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 appropriate 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 the following two components:
- 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 21: For *scientific knowledge understanding*, Gemini-Pro demonstrates significantly better understanding and application capabilities of scientific 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 22: For *scientific knowledge understanding*, OpenChat-3.5 provides fine-grained explanation while encompassing the required core knowledge, garnering preferences from the majority of human subjects, compared to GPT-3.5-Turbo.

**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 (H2) and iodine gas (I2) to form hydrogen iodide (HI). The balanced chemical equation for this reaction is:
H2(g) + I2(g) $\leftrightharpoons$ 2HI(g)
Initially, the reaction is at equilibrium, and the concentrations of H2, I2, and HI are in a specific ratio. Now, let's say we increase the concentration of hydrogen gas (H2) by adding more H2 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 (H2 + I2 $\rightarrow$ 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 H2 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 H2, again restoring the original concentration of H2.
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 (NO2) and dinitrogen tetroxide (N2O4):
2NO2(g) $\leftrightharpoons$ N2O4(g)
If we increase the concentration of NO2, according to Le Chatelier's principle, the system will respond by shifting the equilibrium to the left to counteract the increase in NO2 concentration. As a result, more N2O4 will be formed to consume the excess NO2, thereby reducing the concentration of NO2 and restoring equilibrium.
Conversely, if we decrease the concentration of NO2, the system will shift the equilibrium to the right to produce more NO2 and consume N2O4, 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 23: For *mathematical reasoning*, WizardLM-13B takes an incorrect reasoning path, leading to wrong solutions, in contrast to OpenChat-3.5.

| **Instruction:**<br>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:**<br>Let's assume the width of the garden is W feet.<br>Since the length of the garden is twice the width, the length will be 2W feet.<br>The perimeter of a rectangle is given by the formula: P = 2(length + width)<br>In this case, the perimeter is 60 feet, so we can write the equation as:<br>60 = 2(2W + W)<br>Simplifying the equation, we get: 60 = 2(3W)<br>60 = 6W<br>Dividing both sides by 6, we get:<br>W = 10<br>So, the width of the garden is 10 feet.<br>Since the length is twice the width, the length is 2 * 10 = 20 feet.<br>Therefore, each side of the garden will be 10 feet and the other side will be 20 feet. | **WizardLM-13B Response:**<br>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.<br>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:<br>$b^2 = 2 \times a^2$<br>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:<br>$c^2 = a^2 + b^2$<br>Substituting the given values, we get:<br>$c^2 = 60^2$<br>Solving for 'a', we get: $a = \sqrt{60^2}$<br>Therefore, the width of the garden is $\sqrt{60^2}$ feet. |
| **Human Preference**: OpenChat-3.5 | |

Table 24: For *mathematical reasoning*, WizardLM-13B makes an arithmetic error during the intermediate reasoning process, leading to the eventual collapse of the final result, compared to GPT-3.5-Turbo.

| **Instruction:**<br>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:**<br>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.<br>Step 1: Add the number of cupcakes and cookies.<br>Cupcakes: 40<br>Cookies: 20<br>Step 2: Calculate the total number of items sold.<br>Total items sold = Cupcakes + Cookies = 40 + 20 = 60<br>So, the bakery sold 60 items in total in one day.<br>The answer is: 60. | **GPT-3.5-Turbo Response:**<br>Sure! Let's solve the problem step by step.<br>Let's represent the number of cupcakes sold as C and the number of cookies sold as K.<br>We are given the following information:<br>1. Each cupcake costs $3, and the bakery makes $120 from selling cupcakes. This can be represented as 3C = 120.<br>2. Each cookie costs $2, and the bakery makes $80 from selling cookies. This can be represented as 2K = 80.<br>Now, let's solve the equations to find the values of C and K.<br>From the first equation, we can solve for C:<br>3C = 120<br>C = 120 / 3 = 40<br>From the second equation, we can solve for K:<br>2K = 80<br>K = 80 / 2 = 40<br>So, the bakery sold 40 cupcakes and 40 cookies in one day.<br>To find the total number of items sold, we add the number of cupcakes and cookies:<br>Total items = 40 cupcakes + 40 cookies = 80 items<br>Therefore, the bakery sold a total of 80 items in one day. |
| **Human Preference**: GPT-3.5-Turbo | |

Table 25: For *writing*, the response of GPT-4-Turbo has richer content, while ChatGLM3-6B's response not only fails to meet the instruction requirements, but is also lacking in content.

**Instruction:**
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.

**GPT-4-Turbo Response:**
**Savor the Seasons: A Gastronomic Journey**
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.
**Welcome Bite**
*Velvet Corn Whisper*
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.
**Appetizer**
*Harvest Moon Carpaccio*
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.
**Soup**
*Whispering Waves Bisque*
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.

...(multiple paragraphs omitted)

**Specialty Beverage**
*Sunset Serenade*
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!

**ChatGLM3-6B Response:**
Here's a sample menu for the upcoming food festival:
Welcome to the Taste of Passion Menu
1. **Incorrect**: Baked chicken with roasted red peppers
2. **Correct**: Grilled chicken with roasted red peppers and a zesty avocado salsa
3. **Incorrect**: Spaghetti with meatballs in marinara sauce
4. **Correct**: Hearty spaghetti with meatballs in a rich, homemade tomato sauce
5. **Incorrect**: Roasted vegetables with quinoa and feta cheese
6. **Correct**: Grilled Vegetables with Quinoa and Feta Cheese
7. **Incorrect**: Fruit salad with a honey dressing
8. **Correct**: Heavenly Fruit Salsa with a Shortbread Crumb Topping
9. **Incorrect**: Brownies with chocolate chips
10. **Correct**: 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.

**Human Preference**: GPT-4-Turbo

Table 26: For *coding*, both Gemini-Pro and Vicuna-13B meet the 10-line constraint. Although Gemini-Pro provides less detailed explanation, its correct solutions make it the preferred choice among human evaluators.

**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 27: Failure case of GPT-4-Turbo in *scientific knowledge understanding*. The response from OpenChat-3.5 includes algorithm code after introducing the algorithm itself, 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 s̈ourcen̈ode 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 28: Failure case of GPT-4-Turbo in *mathematical reasoning*. GPT-4-Turbo overlooks 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.

**GPT-4-Turbo Response:**
To solve this problem, we want to find out how many students are in each of the 10 classrooms, given that there are 250 students in total.
Step 1: Determine the total number of students.
The problem states that the total number of students in the school is 250.
Step 2: Determine the number of classrooms.
The problem states that there are 10 classrooms in the school.
Step 3: Divide the total number of students by the number of classrooms to find the number of students per classroom.
Number of students per classroom = Total number of students / Number of classrooms
Step 4: Perform the division.
Number of students per classroom = 250 / 10
Step 5: Calculate the final answer.
Number of students per classroom = 25
So, there are 25 students in each of the 10 classrooms.

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

Table 29: Failure case of GPT-4-Turbo in *coding*. The response from GPT-4-Turbo exhibits errors in the test cases while lacking 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 30: Failure case of GPT-4-Turbo in *writing*. GPT-4-Turbo fails to fully comprehend the meaning of the word "underappreciated" in this 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 toward 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.

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

Table 31: Faiulre case of GPT-4-Turbo in *writing*. GPT-4-Turbo exhibits a sense of "laziness" in its responses, failing to address the demands of the instruction, despite the instruction lacking specific details.

<table>
<tr><td colspan="2">Instruction:<br>Hello, please assist me in drafting my inaugural speech.</td></tr>
<tr><td>OpenChat-3.5 Response:<br>Ladies and Gentlemen, esteemed colleagues, and fellow citizens,<br>It is with great honor and humility that I stand before you today as your newly elected leader. 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.<br>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.<br>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.<br>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.<br>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.<br>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.</td>
<td>GPT-4-Turbo Response:<br>I'd be happy to help you draft your inaugural speech, but I'll need some additional information, such as:<br>1. The position or role you are inaugurating.<br>2. The audience you will be addressing (e.g., community members, professionals, students).<br>3. Key themes or issues you wish to address.<br>4. Your vision and goals in your new role.<br>5. Any specific achievements or acknowledgments you want to include.<br>6. The tone you want to set (e.g., hopeful, determined, conciliatory).<br>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.</td></tr>
<tr><td colspan="2">Human Preference: OpenChat-3.5</td></tr>
</table>