

# REIFE: Re-evaluating Instruction-Following Evaluation

Yixin Liu<sup>\*1</sup> Kejian Shi<sup>\*1</sup> Alexander R. Fabbri<sup>2</sup> Yilun Zhao<sup>1</sup>  
Peifeng Wang<sup>2</sup> Chien-Sheng Wu<sup>2</sup> Shafiq Joty<sup>2</sup> Arman Cohan<sup>1,3</sup>

<sup>1</sup>Yale University <sup>2</sup>Salesforce AI <sup>3</sup>Allen Institute for AI

yixin.liu@yale.edu, kejian.shi@yale.edu, arman.cohan@yale.edu

## Abstract

The automatic evaluation of instruction following typically involves using large language models (LLMs) to assess response quality. However, there is a lack of comprehensive evaluation of these LLM-based evaluators across two dimensions: the base LLMs and the evaluation protocols. Therefore, we present a thorough meta-evaluation of instruction following, including 25 base LLMs and 15 recently proposed evaluation protocols, on 4 human-annotated datasets, assessing the evaluation accuracy of the LLM-evaluators. Our evaluation allows us to identify the best-performing base LLMs and evaluation protocols with a high degree of robustness. Moreover, our evaluation reveals key findings: (1) Base LLM performance ranking remains largely consistent across evaluation protocols, with less capable LLMs showing greater improvement from protocol enhancements; (2) Robust evaluation of evaluation protocols requires many base LLMs with varying capability levels, as protocol effectiveness depends on the base LLM used; (3) Evaluation results on different datasets are not always consistent, so a rigorous evaluation requires multiple datasets with distinctive features. We release our meta-evaluation suite REIFE,<sup>1</sup> which provides the codebase and evaluation result collection for over 500 LLM-evaluators, laying groundwork for future research in instruction-following evaluation.

## 1 Introduction

The ability to follow human instructions has become an important evaluation aspect for large language models (LLMs), indicating their alignment with human users (Ouyang et al., 2022). Recently, due to their better correlation with human judgments compared with traditional evaluation metrics, the LLMs themselves are often used as judges

of the model output quality for generative tasks including instruction following (Liu et al., 2023a; Fu et al., 2023; Zheng et al., 2024). These LLM-based evaluation methods are an essential component of the most widely used automatic benchmarks for instruction-following evaluation, such as AlpacaEval (Li et al., 2023c) and MTBench (Zheng et al., 2024), where a strong LLM is used to evaluate the quality of model responses. Moreover, they can be used as reward models for instruction fine-tuning of LLMs in both distillation and self-improvement settings (Tunstall et al., 2023; Yuan et al., 2024). However, recent studies have identified various limitations of LLM-based evaluation methods, including low self-consistency rates in their predictions, positional biases, and a preference for their own outputs (Liu et al., 2023a; Wang et al., 2024b; Zheng et al., 2024; Panickssery et al., 2024).

Therefore, the evaluation of LLM-based evaluations is critically important. Such evaluations of evaluation methods, or meta-evaluation, usually involve comparing the automatic evaluation results against human evaluation (Liu et al., 2023a; Dubois et al., 2024; Zeng et al., 2024). These evaluations of LLM-evaluators assess two dimensions: (1) the capabilities of *base LLMs* in performing the evaluation task and (2) the effectiveness of *evaluation protocols* — the methods by which base LLMs are used to perform evaluation, e.g., pairwise comparison as in AlpacaEval or pointwise scoring as in MTBench.<sup>2</sup> Existing work (Zheng et al., 2024; Wang et al., 2024b; Zeng et al., 2024) often lacks comprehensiveness in one or both of these dimensions, and more thorough evaluations are needed.

We argue that the following two directions are crucial for a more comprehensive, rigorous evaluation of LLM-evaluators for instruction following: (1) **Including a diverse set of base LLMs**

<sup>\*</sup>Equal contribution

<sup>1</sup>REIFE stands for **Re**-evaluation of **I**nstruction-**F**ollowing **E**valuation: <https://github.com/yale-nlp/ReIFE>.

<sup>2</sup>We use “LLM-evaluator” to refer to an evaluation method that combines a base LLM and an evaluation protocol.

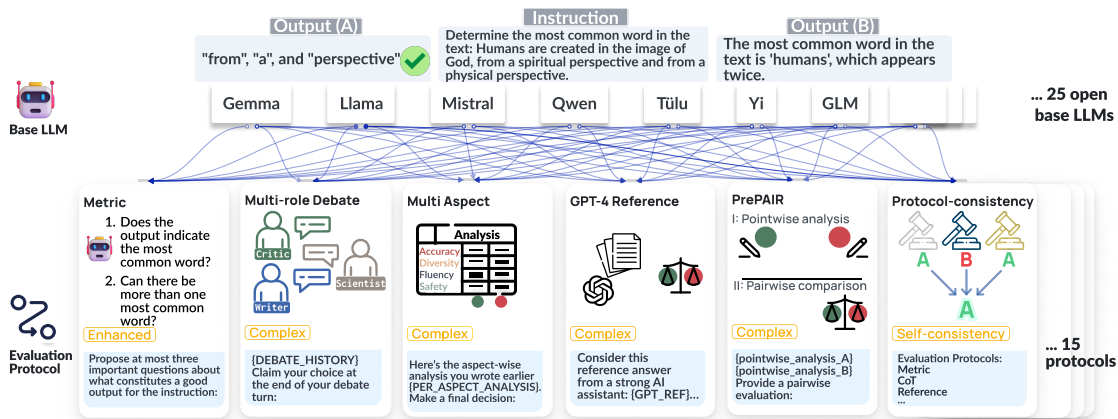


Figure 1: Overview of our large-scale meta-evaluation study of instruction-following evaluation. We evaluate the capabilities of 25 open-source base LLMs and 15 evaluation protocols, resulting in a total of 375 LLM-evaluators – evaluation methods that perform the evaluations using the base LLMs by following the evaluation protocols.

for the evaluation of evaluation protocols – while various evaluation protocols have been proposed recently (Gong and Mao, 2023; Saha et al., 2023; Chan et al., 2024; Jeong et al., 2024), meta-evaluation studies of these evaluation protocols often lack scale in the number of LLMs used. For example, LLMBar (Zeng et al., 2024) uses only 5 LLMs to compare different evaluation protocols. As a result, it remains unclear whether the improvements observed in recently introduced evaluation protocols are robust and generalizable across base LLMs with varying performance levels. Therefore, we aim to conduct an evaluation with a larger and more diverse set of base LLMs to ensure a more rigorous examination of the evaluation protocols. (2) **Expanding the pool of evaluation protocols** for the evaluation of base LLMs – various related studies use only a limited number of evaluation protocols when assessing the evaluation capabilities of different base LLMs (Liu et al., 2023a; Dubois et al., 2024). However, LLMs’ performance can be sensitive to prompt design (Sclar et al., 2024), raising doubts about the reliability of using a single protocol for evaluation. Consequently, we aim to achieve a more reliable evaluation of base LLMs by including a larger set of recently proposed evaluation protocols to account for performance variations from prompt/protocol configurations.

Based on these goals, we present an in-depth meta-evaluation with the following components:

- (1) We perform a solid baseline evaluation across 4 meta-evaluation datasets by evaluating 38 base LLMs and 3 evaluation protocols used in existing benchmarks: AlpacaEval, ArenaHard (Li et al., 2024b), and WildBench (Lin et al., 2024) (§4).
- (2) We gather 15 diverse evaluation protocols based

on previous work, applying a unified prompting style for a fair comparison, and evaluate their average performance with 25 open-source LLMs (§5). (3) We leverage the 375 LLM-evaluators evaluated in §5, to perform a thorough analysis of the practice of meta-evaluation itself, addressing research questions concerning base LLMs, evaluation protocols, and datasets used in the meta-evaluation process, presenting guidelines for best practices (§6).

Our large-scale meta-evaluation, as outlined in Figure 1, enables a rigorous examination of the current progress in LLM-based instruction-following evaluation, providing a solid foundation for developing evaluation protocols and evaluating base LLMs’ evaluation capabilities. We make our meta-evaluation suite REIFE publicly available to facilitate future research, which contains the codebase and evaluation result collection for over 500 LLM-evaluators, and we summarize our key findings:

- Findings** (1) When used in conjunction with 15 evaluation protocols, Llama-3.1-405B (Dubey et al., 2024) is the best open-source base LLM we evaluated (Table 6), which approaches state-of-the-art proprietary LLM performance (Table 2). (2) The evaluation protocols used in 3 widely used benchmarks fail to outperform even the base evaluation protocol evaluated in this work (Table 3). In contrast, the recently introduced evaluation protocol, prepair (Jeong et al., 2024), achieves the highest average performance across 25 open-source LLMs, with 7 of the protocols evaluated significantly outperforming the base protocol (Table 4). (3) The performance ranking of different base LLMs is largely consistent across different evaluation protocols, suggesting that evaluating different base LLMs’ evaluation capabilities with a single

evaluation protocol is likely to yield reliable results (§6.1). However, the benefits of advanced protocols vary across LLMs, with less capable LLMs more likely to gain greater improvements (Table 7). (4) The effectiveness of evaluation protocols depends significantly on the base LLMs used (§6.2). For example, although `prepair` achieves the highest average performance, it ranks only seventh among the 15 evaluation protocols when comparing their optimal performance achieved with the most compatible base LLMs (Table 8). This highlights the need to use multiple base LLMs with varying performance levels for reliable evaluation of evaluation protocols.

(5) Different meta-evaluation datasets can exhibit varying difficulty levels, and the LLM-evaluator rankings on these datasets do not always show a strong positive correlation, demonstrating the importance of incorporating diverse datasets for a more comprehensive meta-evaluation (§6.3).

## 2 Related Work

**LLM-based Evaluation** Using LLMs as evaluators has become a promising approach for assessing text generation quality (Chiang and Lee, 2023; Fu et al., 2023; Liu et al., 2023a) in tasks like summarization (Fu et al., 2023; Liu et al., 2023a,b) and instruction-following (Zheng et al., 2024; Zeng et al., 2024; Li et al., 2023c).

Recent work has proposed various advanced LLM-based evaluation methods. For example, fine-grained or decomposition-based approaches, such as Chain-of-Aspects (Gong and Mao, 2023) and Branch-Solve-Merge (Saha et al., 2023), can guide LLMs to perform structured analysis by identifying fine-grained differences and providing detailed rationales. Agent-based methods, like PRD (Li et al., 2023b) and ChatEval (Chan et al., 2024), employ multi-role debate settings to bring diverse perspectives to the evaluation process. Other techniques include probability-weighted scoring (Liu et al., 2023a), reference-based evaluation (Zeng et al., 2024), and self-consistency decoding (Wang et al., 2023). Our study investigates the effectiveness of these advanced evaluation protocols on a larger scale, assessing their performance across multiple datasets and base LLMs.

Related studies have also explored fine-tuning LLMs as evaluators for various evaluation tasks including instruction-following evaluation (Li et al., 2023a; Wang et al., 2024a), such as

Prometheus (Kim et al., 2024a). However, we choose to exclude them from the majority of our evaluation since our focus is on generic LLMs with various evaluation protocols, while the fine-tuned LLMs usually require a fixed evaluation protocol.

### Human Evaluation and Meta-Evaluation of Instruction-Following

A series of recent studies have conducted human evaluations on instruction-following and/or performed evaluations of automatic evaluators using the collected human annotations (Zhang et al., 2023; Wang et al., 2024b,c; Lan et al., 2024). Among them, the annotations from AlpacaFarm (Dubois et al., 2024) and MTBench (Zheng et al., 2024) have become important testbeds for evaluating widely used LLM evaluators. Zeng et al. (2024) introduce LLMBAR, which consists of high-quality human annotations with a high level of inter-annotator agreement rate. RewardBench (Lambert et al., 2024) provides a benchmark for evaluating reward models used for learning from human or LLM feedback (Ouyang et al., 2022; Bai et al., 2022; Tunstall et al., 2023). While sharing a similar task format, our evaluation focus is different from theirs because we aim to assess the evaluation capability of generic LLMs instead of dedicated reward models.

## 3 Evaluation Settings of REIFE

In REIFE, we evaluate LLM-based instruction-following evaluations along two dimensions: base LLMs and evaluation protocols (Figure 1), using human evaluations as the gold standard. Below, we outline the settings of this evaluation.

**Datasets** We use four datasets to evaluate the LLMs’ capability of instruction-following evaluation. Each dataset includes human annotations for pairwise comparisons of two outputs of an instruction, with a binary label indicating which output is better in instruction following. Table 1 summarizes the dataset information. LLMBAR-Natural and LLMBAR-Adversarial are from Zeng et al. (2024), consisting of data examples examined and edited by the paper authors. MTBench (Zheng et al., 2024) contains expert human annotations made by graduate students for multi-turn conversations. InstruSum (Liu et al., 2024) contains human annotations for instruction-controllable summarization, where the input includes a source article and a specific summary requirement, as a complex instruction. We only use its annotation data samples with

	Abbr.	In.L.	Out.L.	Num.	Agr.
LLMBar-Natural	Nat.	53.3	56.9	100	90%
LLMBar-Adversarial	Adv.	26.7	112.4	319	95%
MTBench	MT.	58.8	192.4	200	81%
InstruSum	Ins.	1149.8	109.2	411	100%

Table 1: Dataset information including abbreviation used (**Abbr.**), average instruction (**In.L.**) and output length (**Out.L.**) in words, number of examples (**Num.**), and annotation agreement rate (**Agr.**).

perfect annotator agreement to reduce the annotation noise. In Appendix A, we show data examples from the datasets. We selected these datasets due to their varying difficulties, instruction complexity, and human annotation noise. For example, InstruSum contains much longer instructions than the other datasets, while MTBench has lower agreement than the others.

**LLM-Evaluator Settings** Since all the datasets we use contain *pairwise* human evaluations, we evaluate LLM-evaluators under the same pairwise comparison setting for evaluation target alignment. We use the term “LLM-evaluator” to refer to a combination of an *base LLM* and an *evaluation protocol*. An evaluation protocol defines how the base LLM performs the evaluation, typically using one or more prompts to query it. By default, we use greedy decoding to ensure deterministic behavior.

**Evaluation Metrics** We mainly use *evaluation accuracy* to evaluate the LLM-evaluators, which measures the alignment with human evaluations using human annotations as the gold standard. Since the LLM-evaluators perform pairwise comparisons, to account for potential *position biases*, where the LLM-evaluators may favor either the first or the second output (Wang et al., 2024b), we report the averaged evaluation accuracy across two directions, swapping the order of the two outputs. An auxiliary metric we used is the *self-agreement rate* of the LLM-evaluators in their predictions across two directions in Krippendorff’s alpha (Krippendorff, 2011), measuring their positional biases.

## 4 Baselines

We first establish baselines for base LLMs and evaluation protocols regarding evaluating instruction-following for our further investigations.

### 4.1 Baselines for Base LLMs

To benchmark the baseline performance of base LLMs at instruction-following evaluation, we evaluate them with a simple evaluation protocol to con-

struct the corresponding LLM-evaluators. This *base* evaluation protocol, proposed in Zeng et al. (2024), requires the LLM-evaluators to directly predict which output is better, with rules to constrain output formats and to avoid potential biases.<sup>3</sup>

Table 2 presents the evaluation accuracy of 38 proprietary and open-source LLMs, together with two state-of-the-art reward models, nemotron-4-340b-rm (Adler et al., 2024) and offsetbias-rm (Park et al., 2024), and two strong fine-tuned LLM-evaluators, prometheus-2-8x7b (Kim et al., 2024a) and offsetbias-1m (Park et al., 2024) as baselines. The model information is in Appendix B at Table 12. We note the following observations:

- (1) **Proprietary vs. Open-Source:** the open-sourced llama-3.1-405b outperforms most of the proprietary LLMs, and llama-3.1-70b lags just slightly behind gpt-4o and gpt-4o613.
- (2) **Performance Gap:** The LLMs at the lower end, such as llama-2-7b and gemma-2b, achieve an accuracy near 50%, comparable to a random oracle. On the other hand, llama-3.1-405b achieves a high accuracy of approximately 84%.
- (3) **Dataset Difficulty:** There is also a large difference in the average LLM performance across different datasets. For example, the average evaluation accuracy on LLMBar-Natural is around 20% higher than LLMBar-Adversarial.
- (4) **Comparisons with Reward Models and Fine-tuned LLMs.** The strongest LLM-evaluators outperform the state-of-the-art reward models and fine-tuned LLM-evaluators. The fine-tuned LLM-evaluator, offsetbias-1m, shows a significant improvement over its base model, llama-3-8b, suggesting the potential of fine-tuned LLM-evaluators. Meanwhile, prometheus-2-8x7b only outperforms its base model (mixtral-8x7b) on the easier datasets LLMBar-Natural and MTBench, indicating a lack of robustness.

These baseline results show that the top open-source LLMs already approach the performance of their proprietary counterparts and offer a wide performance spectrum. Therefore, for transparency and reproducibility, we will use mostly open-source LLMs in the rest of our evaluations.

### 4.2 Baselines for Evaluation Protocols

We now establish a baseline for evaluation protocols, which define how the base LLM is used to

<sup>3</sup>The prompt template is in Appendix C.1.

Model	Nat.	Adv.	MT.	Ins.	Avg.
<b>Proprietary LLMs</b>					
gpt-4o-24-08-06	<b>97.5</b>	84.5	79.8	<b>81.3</b>	<b>85.7</b>
o1-mini-24-09-12	92.5	<b>88.6</b>	79.0	<b>81.3</b>	85.3
gpt-4-0613	95.5	79.3	<u>81.5</u>	80.4	84.2
gpt-4o-24-05-13	95.5	80.7	<u>79.5</u>	80.3	84.0
claude-3.5-sonnet	91.0	81.2	78.5	77.5	82.0
claude-3-opus	94.0	76.8	75.5	74.1	80.1
mistral-large	90.0	72.1	79.0	78.5	79.9
gemini-1.5-pro	87.0	74.9	78.5	75.7	79.0
gemini-1.5-flash	87.5	71.3	77.8	77.5	78.5
gpt-4o-mini	88.5	68.3	80.2	76.6	78.4
gemini-1.0-pro	85.5	54.5	70.8	68.7	69.9
gpt-3.5-turbo-0125	82.5	36.4	72.8	63.5	63.8
claude-3-haiku	76.0	42.9	68.8	62.8	62.6
<b>Open-source LLMs</b>					
llama-3.1-405b	94.0	<u>83.1</u>	81.5	<u>79.6</u>	<u>84.5</u>
llama-3.1-70b	90.5	79.3	82.2	79.4	82.9
llama-3-70b	87.0	72.7	80.0	78.6	79.6
qwen-2-72b	92.5	69.4	82.2	73.1	79.3
qwen-2.5-72b	90.5	67.7	<b>82.5</b>	74.1	78.7
qwen-1.5-72b	88.5	59.7	75.0	69.2	73.1
glm-4-9b	86.0	55.0	73.5	73.4	72.0
yi-1.5-34b	86.5	56.6	73.8	66.9	70.9
tulu-2-dpo-70b	85.5	58.9	73.2	66.1	70.9
tulu-2-70b	86.5	58.0	74.5	64.7	70.9
mixtral-8x7b	80.5	58.9	73.0	68.7	70.3
yi-1.5-9b	85.0	59.1	72.5	63.1	69.9
qwen-1.5-32b	85.5	47.3	76.8	66.2	68.9
llama-3.1-8b	78.0	50.9	72.5	66.5	67.0
llama-2-70b	80.0	32.4	72.2	66.9	62.9
llama-3-8b	70.5	43.6	72.5	61.7	62.1
mistral-7b-v0.3	64.5	48.0	66.3	60.6	59.8
tulu-2-dpo-13b	67.0	38.6	65.5	61.2	58.1
tulu-2-13b	65.5	38.6	65.5	61.8	57.8
llama-2-13b	65.0	36.4	66.8	60.8	57.2
tulu-2-dpo-7b	56.0	43.4	58.5	58.9	54.2
gemma-7b	52.5	39.3	64.5	57.4	53.4
tulu-2-7b	45.5	46.9	55.2	57.8	51.4
llama-2-7b	42.5	49.5	52.0	56.4	50.1
gemma-2b	42.5	44.8	54.5	56.6	49.6
<b>Reward Models</b>					
offsetbias-rm	93.0	77.1	81.0	74.0	81.3
nemotron-4-340b	95.0	84.6	75.5	69.3	81.1
<b>Fine-tuned LLMs</b>					
offsetbias-lm	88.0	79.9	80.0	74.8	80.7
prometheus-2	83.0	37.3	76.0	64.4	65.2
avg.	80.7	60.2	73.4	69.3	70.9

Table 2: Evaluation accuracy of various base LLMs with the base evaluation protocol. Models are ordered by their performance. Nat. is LLMBAR-Natural, Adv. is LLMBAR-Adversarial, MT. is MTBench, Ins. is InstruSum. Best column performance is bolded, and best group performance is underlined.

perform the evaluation. To this end, we evaluate the evaluation protocols used in three automatic LLM benchmarks for instruction-following<sup>4</sup> – AlpacaEval (Li et al., 2023c), ArenaHard (Li et al., 2024b),

<sup>4</sup>The prompt templates are in Appendix C.2.

Protocol	Nat.	Adv.	MT.	Ins.	Avg.
Average across 25 Open-Source Base LLMs					
base	74.7	<b>53.5</b>	70.7	<b>66.0</b>	<b>66.2</b>
arena-hard	<b>76.3</b>	46.2	<b>72.1</b>	64.5	64.8
wild-bench	74.9	47.5	70.7	63.0	64.0
alpaca-eval	65.3	49.8	63.1	59.4	59.4
gpt-4o-2024-0806 as Base LLM					
base	<b>97.5</b>	<b>84.5</b>	79.8	<b>81.3</b>	<b>85.7</b>
arena-hard	94.5	78.8	83.2	75.4	83.0
wild-bench	95.5	75.5	<b>84.0</b>	73.7	82.2
alpaca-eval	94.0	70.2	83.7	76.5	81.1

Table 3: Evaluation accuracy of evaluation protocols in existing LLM benchmarks compared against the base evaluation protocol.

and WildBench (Lin et al., 2024). Each of these benchmarks uses their evaluation protocol together with a strong base LLM, e.g., GPT-4 (Achiam et al., 2023), to perform *pairwise* comparison of different LLMs’ outputs. The individual comparison results are then aggregated to produce a performance ranking of various LLMs. We note that the efficacy of these benchmarks is evaluated at the *system level*, where their produced ranking is compared against the system ranking from human evaluation benchmarks, e.g., ChatBot Arena (Chiang et al., 2024). In contrast, here we aim to evaluate the performance of their evaluation protocols at the *instance level*, measuring their evaluation accuracy against human annotations at individual data instances.

In Table 3, the benchmark evaluation protocols are compared against the base protocol (Zeng et al., 2024) used in §4.1, where they are used together with the 25 open-source base LLMs evaluated in §4.1 and the strongest proprietary LLM, gpt-4o. It shows that the benchmark protocols cannot outperform the base protocol, especially on the more challenging LLMBAR-Adversarial and InstruSum datasets. This indicates that the complex design of the benchmark protocols, which often includes detailed instructions on the evaluation plan and output structure, cannot improve the LLM-evaluators performance at the instance level. In the next section, we will provide a further examination of various evaluation protocols.

## 5 Evaluating Evaluation Protocols

In §4, we only tested the LLM-evaluators with the base and benchmark evaluation protocols. We now expand the evaluation dimensions to include various protocols proposed in recent work. By using 25

open-source LLMs, we believe this evaluation will provide a fairer and more rigorous examination.

## 5.1 Evaluation Protocols

In our evaluation, we examine 15 protocols derived from previous work. To address the unavailability of some prompt templates and to ensure a fair comparison, we design prompt templates ourselves when necessary. We ensure that all prompt templates adhere to unified formatting and style, and we refine them iteratively to make sure that the protocols can perform to their full potential. The evaluated protocols are outlined below, with their prompt templates provided in Appendix C.

**Baseline Protocol** (1) *base*: the vanilla approach used in §4 which directly predicts the pairwise comparison outcome, proposed in Zeng et al. (2024).

**Enhanced Protocols** Five other protocols from Zeng et al. (2024) are evaluated, which include various enhancements based on the base protocol: (2) *cot*: the LLM is asked to provide a chain-of-thought (Wei et al., 2022) explanation before making the final decision.

(3) *metric*: the LLM is prompted to generate a few metrics for the evaluation task first, which are later used in the actual evaluation.

(4) *reference*: the LLM is prompted to generate a “reference” output for the given instruction, which is later used in the actual evaluation.

(5) *metric+reference*: a combination of the metric and reference methods.

(6) *swap&synthesize*: based on *cot* and inspired by Du et al. (2024), this method requires the LLM to resolve self-disagreement in predictions from two output orders and make a final decision.

**Complex Protocols** Beyond the enhanced protocols, 7 complex protocols are evaluated based on evaluation methods proposed in previous work.

(7) *fine-grained-diff*: Similar to Min et al. (2023), this protocol guides the LLM to first identify *fine-grained differences* in output pairs and then provide a detailed rationale for choosing the better output considering these differences.

(8) *multi-role-round1* & (9) *multi-role-round2*: Inspired by frameworks that use multiple agents as judges (Li et al., 2023b; Chan et al., 2024; Zhao et al., 2024), these two protocols use a *multi-role debate* setting where multiple evaluators are instantiated from an LLM using prompts with

specific role descriptions, to bring diverse perspectives to the evaluation process. The evaluators will generate their responses sequentially, potentially in *multiple rounds*, to engage in a debate that leads to the final prediction. We evaluate its single-round and two-round variants.

(10) *multi-aspect-two* & (11) *multi-aspect-single*: Similar to several related studies (Saha et al., 2023; Gong and Mao, 2023; Li et al., 2023a, 2024a), this protocol performs a *multi-aspect comparison* of the output pairs, with two variants: the *two-stage* protocol prompts the LLMs to evaluate each quality aspect in separate inference passes, while the *single-stage* protocol requires the LLMs to conduct a multi-aspect evaluation in a single inference pass before making the final prediction.

(12) *gpt4-reference*: Similar to the reference protocol, this protocol uses a reference output generated by gpt-4o to assist the evaluation.

(13) *prepair*: Adapted from Jeong et al. (2024), this protocol incorporates pointwise reasoning within a pairwise evaluation framework, leveraging the robustness of pointwise evaluation against biases while maintaining the comparative benefits of pairwise evaluation.

**Self-consistency Protocols** Self-consistency is a commonly used decoding approach that can improve the LLMs’ performance in various reasoning tasks (Wang et al., 2023). Used together with CoT prompting, self-consistency generates the final prediction by taking a majority vote on the predictions made in each generation pass.

(14) *cot&self-consistency*: Self-consistency in pairwise comparison determines the more frequently preferred output. We use a sampling temperature of 0.7 and generate 9 CoTs for voting.

(15) *protocol-consistency*: Beyond different CoTs, a majority vote can be applied across various evaluation protocols. We evaluate this approach using the 5 *enhanced* protocols.

## 5.2 Results

**Evaluation Accuracy** Table 4 demonstrates the evaluation accuracy of various protocols averaged over different base LLMs. We note the following:

(1) *prepair* achieves the strongest average performance, achieving a 1.7% higher accuracy compared to the base protocol.

(2) *multi-aspect-two* achieves the best performance on LLMBAR-Natural and MTBench. However, its performance on LLMBAR-Adversarial

Protocol	Nat.	Adv.	MT.	Ins.	Avg.
prepair (13)	↑76.4	↑ <b>61.8</b>	69.7	↓63.8	↑ <b>67.9</b>
gpt4-reference (12)	↑76.7	↑58.0	70.1	66.0	↑67.7
metric+reference (5)	↑76.6	↑58.3	↓70.0	65.6	↑67.6
protocol-consistency (15)	↑76.3	↑55.9	70.9	66.1	↑67.3
metric (3)	↑75.8	↑56.2	70.7	65.7	↑67.1
reference (4)	↑76.2	↑57.5	↓69.4	↓65.2	↑67.1
swap&synthesize (6)	75.6	↑54.4	70.8	<b>66.2</b>	↑66.8
cot&consistency (14)	74.9	54.1	70.5	↓65.4	66.2
base (1)	74.7	53.5	70.7	66.0	66.2
cot (2)	↓73.6	53.6	70.2	↓64.9	↓65.6
multi-aspect-two (10)	↑ <b>77.1</b>	↓42.3	↑ <b>72.5</b>	↓62.3	↓63.6
fine-grained-diff (7)	↓71.2	↓49.5	↓69.2	↓61.8	↓62.9
multi-role-round1 (8)	↓68.0	53.9	↓66.2	↓61.6	↓62.4
multi-role-round2 (9)	↓68.4	53.4	↓65.7	↓61.7	↓62.3
multi-aspect-single (11)	↓69.6	↓40.8	70.5	↓62.4	↓60.8

Table 4: Average evaluation accuracy of different evaluation protocols across various LLMs, ordered by their average performance. The protocol indexes introduced in §5.1 are in parentheses. ↑, ↓: significantly better or worse than the base protocol ( $p < 0.05$ ).

ranks among the worst. This highlights that the protocol performance can significantly vary across different datasets.

(3) On average, most complex protocols fail to outperform the base protocol, despite their higher computational costs and multi-step, fine-grained nature, indicating that designing a robust, superior evaluation protocol is a non-trivial task.

(4) Similarly, the approaches that have been proven effective on various reasoning tasks, chain-of-thought (cot) and self-consistency (cot&consistency), also fail to bring significant improvement over the base protocol.

**Self-Agreement Rate** Appendix D shows the protocols self-agreement rates of different evaluation protocols, demonstrating that swap&synthesize can significantly enhance the self-agreement rate.

**Best LLM-Evaluators** Table 5 displays the LLM-evaluators that achieve the highest evaluation accuracy on each dataset, together with the evaluation accuracy of the same base LLM achieved with the base protocol. It shows that llama-3.1-405b, which achieves the strongest performance among the open-source LLMs, remains the strongest base LLM across three datasets. On the other hand, the evaluation protocols used by the best LLM-evaluators on the four datasets differ, indicating greater variance in their capabilities.

## 6 Analysis

In §5, a total number of 375 LLM-evaluators are evaluated, combining 25 base LLMs and 15 evalua-

Dataset	LLM	Protocol	Acc.	B.Acc.
Nat.	llama-3.1-405b	swap&synthesize	98.0	94.0
Adv.	llama-3.1-405b	gpt4-reference	87.8	83.1
MT.	qwen-2.5-72b	metric+reference	84.0	82.5
Ins.	llama-3.1-405b	prepair	82.7	79.6

Table 5: Best LLM-evaluators identified on each dataset. **Acc.** is the evaluation accuracy of the best LLM-evaluator, **B.Acc.** is the evaluation accuracy achieved by the same base LLM with the base protocol.

Model	Nat.	Adv.	MT.	Ins.	Avg.
llama-3.1-405b	<b>94.1</b>	<b>81.3</b>	81.4	<b>80.0</b>	<b>84.2</b>
llama-3.1-70b	91.7	80.2	81.0	75.3	82.1
qwen-2-72b	91.6	69.9	<b>82.2</b>	72.8	79.1
qwen-2.5-72b	89.6	71.0	81.2	72.4	78.6
llama-3-70b	88.2	71.5	80.0	74.3	78.5
qwen-1.5-72b	86.1	56.0	76.4	68.0	71.6
yi-1.5-34b	86.5	57.6	73.9	66.0	71.0
tulu-2-dpo-70b	84.2	56.1	73.7	66.5	70.1
mixtral-8x7b	82.2	54.7	73.8	66.4	69.3
tulu-2-70b	83.6	55.0	72.8	65.4	69.2
qwen-1.5-32b	85.7	47.9	77.0	64.3	68.7
glm-4-9b	79.1	54.4	70.6	68.2	68.1
yi-1.5-9b	77.1	55.7	71.3	60.1	66.1
llama-3.1-8b	75.3	54.7	70.3	62.3	65.7
llama-3-8b	71.8	47.9	71.4	62.6	63.4
llama-2-70b	74.4	36.6	68.5	63.3	60.7
mistral-7b-v0.3	65.1	47.3	66.6	61.9	60.2
tulu-2-dpo-13b	66.0	40.0	65.3	60.3	57.9
tulu-2-13b	63.3	39.7	65.4	60.2	57.2
llama-2-13b	63.9	36.3	62.7	56.9	54.9
tulu-2-dpo-7b	58.2	43.7	57.2	57.2	54.1
gemma-7b	51.9	42.7	59.2	56.7	52.6
tulu-2-7b	49.1	45.8	54.9	56.9	51.7
llama-2-7b	48.6	45.3	54.7	54.4	50.8
gemma-2b	44.6	47.0	53.8	55.5	50.3

Table 6: Evaluation accuracy of different base LLMs averaged over 15 evaluation protocols. The base LLMs are ordered by their average performance.

tion protocols. We now present detailed analyses of the base LLMs, evaluation protocols, and datasets, using these comprehensive evaluation results to address a series of specific research questions.

### 6.1 Analysis of Base LLMs

**What is the average performance of the base LLMs across different protocols?** Table 6 displays the LLMs’ average evaluation accuracy over 15 protocols, showing that llama-3.1-70b is the strongest evaluator on average, while qwen-2-72b achieves the best performance on MTBench.

**How does base LLMs’ ranking change with different protocols?** Figure 2 shows the evaluation accuracy of base LLMs achieved with the base protocol and the average accuracy across different

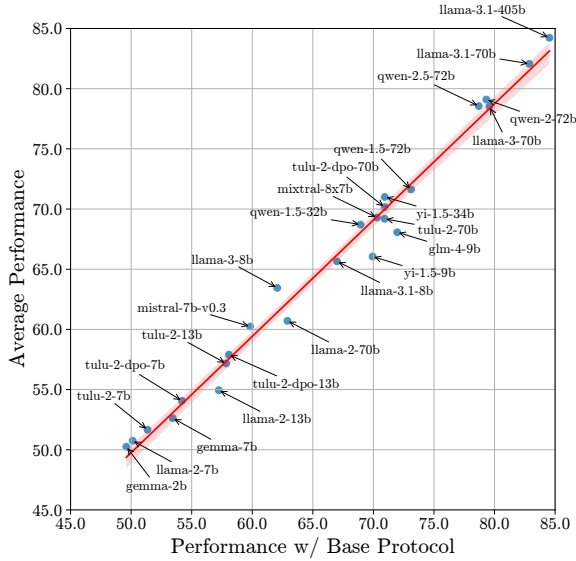


Figure 2: Correlation between the base LLMs’ evaluation accuracy with the base protocol and their average accuracy across 15 protocols. The fitted regression line and the 95% confidence interval are displayed.

Model	Nat.	Adv.	MT.	Ins.	Avg.
mistral-7b-v0.3	16.0	7.8	8.2	<b>5.5</b>	<b>9.4</b>
llama-2-7b	<b>23.5</b>	0.2	<b>11.7</b>	1.2	9.2
tulu-2-7b	21.0	2.7	10.3	1.5	8.8
llama-3-8b	9.5	17.9	3.0	4.5	8.7
yi-1.5-34b	5.0	17.6	4.5	3.6	7.7
tulu-2-dpo-13b	10.0	11.8	6.0	1.7	7.4
llama-3.1-8b	5.0	<b>18.5</b>	3.3	-0.7	6.5
tulu-2-dpo-7b	15.5	5.2	5.2	0.0	6.5
qwen-1.5-32b	4.0	16.5	2.0	3.3	6.4
tulu-2-dpo-70b	4.0	11.0	3.0	3.5	5.4
tulu-2-13b	5.0	10.7	4.5	1.0	5.3
glm-4-9b	1.0	14.1	5.0	0.9	5.2
qwen-2.5-72b	2.5	13.6	1.5	3.2	5.2
mixtral-8x7b	6.0	7.4	4.2	1.1	4.7
gemma-7b	8.0	9.7	-0.2	0.6	4.5
llama-2-70b	2.5	11.8	0.5	3.3	4.5
tulu-2-70b	4.0	8.5	0.3	5.2	4.5
llama-3-70b	4.5	8.9	2.2	2.1	4.4
qwen-1.5-72b	1.0	8.6	4.0	0.7	3.6
gemma-2b	6.5	5.6	2.0	0.1	3.6
qwen-2-72b	2.0	8.8	1.2	1.8	3.5
llama-2-13b	6.0	6.0	1.5	0.2	3.4
llama-3.1-405b	4.0	4.7	0.8	3.2	3.2
llama-3.1-70b	4.5	5.5	0.2	1.2	2.9
yi-1.5-9b	0.5	5.6	2.3	1.6	2.5

Table 7: Optimal evaluation accuracy improvement ( $\tilde{s}$ ) of base LLMs achieved by the most compatible evaluation protocols, ordered by average improvement.

protocols. The results demonstrate a high positive correlation between them, achieving a Spearman’s coefficient of 0.983. This indicates that **using a single base protocol for the evaluation of the base LLMs’ is likely to yield reliable results.**

Protocol	Nat.	Adv.	MT.	Ins.	Avg.
metric+reference (3)	95.0	86.2	<b>84.0</b>	82.4	<b>86.9</b>
reference (6)	97.5	85.9	83.2	80.9	<b>86.9</b>
swap&synthesize (7)	<b>98.0</b>	84.8	82.7	80.4	86.5
gpt4-reference (2)	94.5	<b>87.8</b>	82.5	81.0	86.4
cot&consistency (8)	96.5	84.2	82.2	81.5	86.1
protocol-consistency (4)	95.0	85.0	82.8	81.4	86.0
prepair (1)	94.5	84.8	81.2	<b>82.7</b>	85.8
metric (5)	95.0	82.1	83.5	80.9	85.4
cot (10)	96.0	82.8	82.2	79.7	85.2
base (9)	94.0	83.1	82.5	79.6	84.8
multi-role-round2 (14)	94.0	81.5	81.5	79.3	84.1
fine-grained-diff (12)	92.5	77.9	83.0	79.4	83.2
multi-role-round1 (13)	92.0	81.7	80.2	76.8	82.7
multi-aspect-two (11)	93.0	73.0	83.0	79.0	82.0
multi-aspect-single (15)	87.5	65.8	83.0	75.8	78.0

Table 8: Optimal evaluation accuracy of different evaluation protocols with the most compatible base LLMs, ordered by their performance. The protocols’ average performance rankings (Table 4) are in parentheses.

**How large is the optimal improvement gained from different evaluation protocols for base LLMs?** Table 7 displays the optimal evaluation accuracy improvement ( $\tilde{s}$ ) achieved by different evaluation protocols over the base protocol for various base LLMs. That is,  $\tilde{s} = \max_{p \in \mathcal{P}} s(p) - s(\hat{p})$ , where  $s(p)$  is the evaluation accuracy of an evaluation protocol  $p$ ,  $\hat{p}$  denotes the base protocol,  $\mathcal{P}$  is the set of protocols excluding  $\hat{p}$ . The results indicate that **less capable LLMs are more likely to achieve larger improvements when the suitable protocols are used**, showing a -0.455 Spearman’s correlation between the base LLMs’ performance with the base protocol and their optimal performance with the most compatible evaluation protocol. We hypothesize this is because the inductive biases and constraints introduced by the more complicated protocols help less capable LLMs overcome their potential biases and limitations.

## 6.2 Analysis of Evaluation Protocols

**What is the evaluation protocols’ optimal performance?** In Table 4 of §5.2, the evaluation protocols’ performance is evaluated across all base LLMs. Table 8 instead shows the optimal performance of evaluation protocols, i.e., their evaluation accuracy with the most compatible base LLM. The results show that the **evaluation protocols’ optimal performance can significantly differ from their average performance.** For example, while prepair achieves the best average performance, it ranks only 7th in terms of optimal performance.



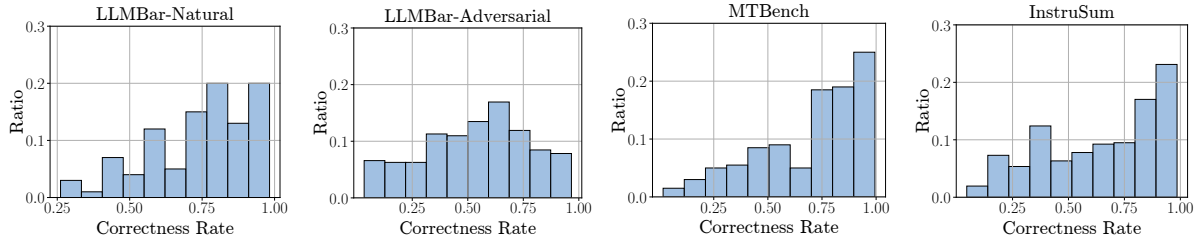


Figure 3: Distribution of the correctness rate of data examples in each dataset over different LLM-evaluators.

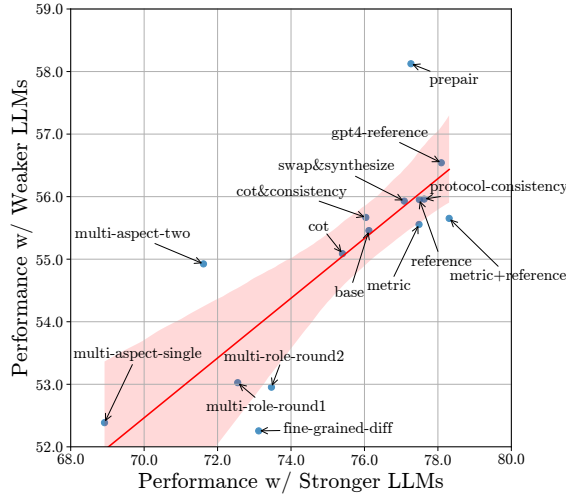


Figure 4: Evaluation protocols’ evaluation accuracy with the stronger and weaker base LLM groups, with a fitted regression line and a 95% confidence interval.

**How do base LLMs’ capabilities affect evaluation protocol’s performance?** As shown above, the evaluation protocol’s performance can be significantly affected by the base LLMs used. Therefore, we provide a further examination with two groups of LLMs: one containing the strongest 10 LLMs identified in Table 6, and another containing the weakest 10. Figure 4 demonstrates that the effectiveness of evaluation protocols can substantially vary based on the capabilities of the base LLMs used. For example, prepair is significantly better than other protocols with weaker LLMs, while metric+reference works better with stronger LLMs. This suggests that **a robust evaluation of evaluation protocols requires multiple base LLMs with a diverse performance range.**

### 6.3 Analysis of Datasets

#### What is the difficulty level of different datasets?

With our large set of LLM-evaluators, we can gauge the difficulty of each data example by computing its *correctness rate*, which we define as the average evaluation accuracy across all evaluators. Figure 3 shows the distribution of this correctness rate. We note: (1) LLMBar-Natural, MTBench, and InstruSum have a similar data example difficulty

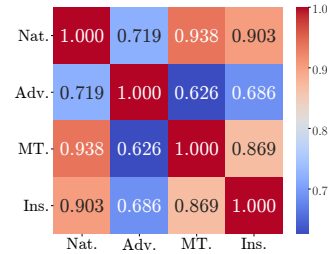


Figure 5: Spearman’s correlations between the performance ranking of LLM-evaluators on different datasets.

distribution, with a small portion of very challenging examples; (2) LLMBar-Adversarial exhibits a different pattern, where the distribution peaks at examples of medium difficulty. This suggests that different LLM-evaluators may have distinct sets of adversarial examples.

**Are LLM-evaluators’ rankings consistent over different datasets?** To better understand how the LLM-evaluators’ performance differs across different datasets, in Figure 5 we present the Spearman’s correlations between LLM-evaluators’ performance ranking between different datasets. The results show that LLMBar-Natural and MTBench exhibit the highest level of similarity. In contrast, LLMBar-Adversarial displays a much lower correlation with the other datasets, suggesting the necessity of using multiple datasets for evaluation.

## 7 Conclusion

In this work, we conducted a large-scale meta-evaluation of instruction following, examining 25 open-source base LLMs and 15 evaluation protocols while identifying the best-performing LLM-evaluators over 4 datasets. We found that a reliable evaluation of base LLMs’ evaluation capabilities can likely be achieved with a single evaluation protocol due to the stability of their performance across different protocols. However, evaluating evaluation protocols should involve a diverse group of base LLMs, as they can significantly impact the evaluation protocols’ effectiveness. We hope that our findings and meta-evaluation suite, REIFE, can pave the way for future studies in this direction.

## Limitations

**Evaluation Scope:** Our evaluation centered around generic LLMs and evaluation protocols. As discussed in §2, we did not focus on reward models trained to evaluate output quality or LLMs fine-tuned for instruction-following. We note that a future study incorporating these systems could yield more comprehensive results.

**Prompt Variations:** In our evaluations, we aimed to control the impact of prompt design by minimizing unnecessary differences across different evaluation protocols. However, we acknowledge that a more thorough evaluation involving multiple prompt variants for each protocol would likely produce more stable results.

**Qualitative Human Evaluation:** We primarily used high-quality human annotation datasets for our quantitative meta-evaluation. Nevertheless, we recognize the lack of qualitative human evaluation, especially concerning the rationales generated by different LLM-evaluators, which could provide further insights into their limitations. We provide a preliminary case study in Appendix E showcasing the error patterns of the base LLMs, and another in Appendix F demonstrating the effect of different evaluation protocols on the same base LLM.

## Acknowledgements

We are grateful to OpenAI’s Researcher Access Program and Together AI for provision of LLM API credits.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.
01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. [Yi: Open foundation models by 01.ai](#). *Preprint*, arXiv:2403.04652.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Sheng-guang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional AI: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
- Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.
- Cheng-Han Chiang and Hung-yi Lee. 2023. [Can large language models be an alternative to human evaluations?](#) In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. [Chatbot arena: An open platform for evaluating LLMs by human preference](#). In *Forty-first International Conference on Machine Learning*.
- Claude. 2023. [Introducing claude](#).
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate.
- Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie

Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Celebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Roman Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gouget, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie DelPierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva

Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymmer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldmann, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhota, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao,

- Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*.
- Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.
- Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as you desire. *arXiv preprint arXiv: 2302.04166*.
- Team Gemini, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Team Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv: 2403.08295*.
- Peiyuan Gong and Jiaxin Mao. 2023. CoAScore: Chain-of-aspects prompting for nlg evaluation. *arXiv preprint arXiv: 2312.10355*.
- Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. [Camels in a changing climate: Enhancing lm adaptation with tulu 2](#). *Preprint*, arXiv:2311.10702.
- Hawon Jeong, ChaeHun Park, Jimin Hong, and Jaegul Choo. 2024. PRePair: Pointwise reasoning enhance pairwise evaluating for robust instruction-following assessments. *arXiv preprint arXiv: 2406.12319*.
- 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 preprint arXiv:2310.06825*.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Léo Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *arXiv preprint arXiv: 2401.04088*.
- Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoon Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

- Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv: 2405.01535*.
- Klaus Krippendorff. 2011. Computing krippendorff’s alpha-reliability.
- Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv: 2403.13787*.
- Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian-ling Mao. 2024. Criticbench: Evaluating large language models as critic. *arXiv preprint arXiv:2402.13764*.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023a. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.
- Minzhi Li, Zhengyuan Liu, Shumin Deng, Shafiq Joty, Nancy F. Chen, and Min-Yen Kan. 2024a. Decompose and aggregate: A step-by-step interpretable evaluation framework. *arXiv preprint arXiv: 2405.15329*.
- Ruosen Li, Teerth Patel, and Xinya Du. 2023b. PRD: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv: 2307.02762*.
- Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024b. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023c. AlpacaEval: An automatic evaluator of instruction-following models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval).
- Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. WILDBENCH: Benchmarking llms with challenging tasks from real users in the wild. *arXiv preprint arXiv:2406.04770*.
- Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.
- Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.
- Yixin Liu, Alexander Fabbri, Yilun Zhao, Pengfei Liu, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023c. Towards interpretable and efficient automatic reference-based summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16360–16368, Singapore. Association for Computational Linguistics.
- Yixin Liu, Alexander R Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohen. 2024. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*.
- Meta AI. 2024. Introducing meta llama 3: The most capable openly available llm to date. <https://ai.meta.com/blog/meta-llama-3/>. Accessed: 2024-05-30.
- Sewon Min, Kalpesh Krishna, Xinxin Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.
- Junsoo Park, Seungyeon Jwa, Meiyang Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. *arXiv preprint arXiv:2407.06551*.
- Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv: 2310.15123*.

- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting](#). In *The Twelfth International Conference on Learning Representations*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of LM alignment. *arXiv preprint arXiv:2310.16944*.
- Peifeng Wang, Austin Xu, Yilun Zhou, Caiming Xiong, and Shafiq Joty. 2024a. Direct judgement preference optimization. *arXiv preprint arXiv:2409.14664*.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024b. [Large language models are not fair evaluators](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.
- Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2024c. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *The Twelfth International Conference on Learning Representations*.
- Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024d. [HelpSteer2: Open-source dataset for training top-performing reward models](#). *Preprint, arXiv:2406.08673*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.
- Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.
- Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*.
- Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Deli Zhao, and Lidong Bing. 2024. Auto Arena of LLMs: Automating llm evaluations with agent peer-battles and committee discussions. *arXiv preprint arXiv:2405.20267*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

## A Dataset Examples

We randomly select two data examples from each of the four datasets in our study (§3), and present them in Table 10 and Table 11.

## B Details of Base LLMs

We provide brief descriptions for the 38 base LLMs adopted in this study in Table 12, discussed in §4.

## C Details of Evaluation Protocols

We provide a list of the 15 evaluation protocols investigated in this study in Table 13, detailed in §5.1. For each protocol, we also include the number of queries required per evaluation (#Qs./Eval) and any additional auxiliary queries (#Aux.Qs./Eval) where applicable. Below, we provide the prompt templates used for all prompting-based evaluation protocols in §5.1.

### C.1 Prompt for Base Evaluation Protocol

Figure 6 shows the prompt for the base protocol. It corresponds to the *Vanilla+Rules* prompting strategy proposed in Zeng et al. (2024).

### C.2 Prompts for Benchmark Evaluation Protocols

We provide the prompts adopted from AlpacaEval (Li et al., 2023c) (Figure 7), ArenaHard (Li et al., 2024b) (Figure 8), and WildBench (Lin et al., 2024) (Figure 9). The original evaluation protocols of ArenaHard and WildBench perform five-scale pairwise comparisons between the output pairs. To

Protocol	Nat.	Adv.	MT.	Ins.	Avg.
swap&synthesize (6)	<b>89.3</b>	<b>84.4</b>	<b>86.0</b>	<b>86.1</b>	<b>86.5</b>
prepair (13)	62.1	59.7	59.3	38.9	55.0
multi-aspect-two (10)	62.4	51.9	54.6	36.6	51.4
fine-grained-diff (7)	53.1	45.3	54.6	33.5	46.6
protocol-consistency (15)	56.5	47.0	50.9	30.7	46.3
metric (3)	53.6	48.4	52.7	29.4	46.0
metric+reference (5)	57.4	48.7	49.7	28.3	46.0
base (1)	54.5	47.3	49.6	32.4	46.0
cot&consistency (14)	54.0	42.0	48.1	36.9	45.2
multi-aspect-single (11)	45.6	44.6	50.7	35.9	44.2
gpt4-reference (12)	51.3	47.2	45.9	25.7	42.5
cot (2)	47.7	38.7	45.7	34.6	41.7
reference (4)	50.8	44.5	44.2	24.9	41.1
multi-role-round1 (8)	38.3	26.8	31.8	25.4	30.6
multi-role-round2 (9)	37.1	25.9	31.8	25.9	30.2

Table 9: Average self-agreement rate of various evaluation protocols across various LLMs (ordered). The protocol indexes introduced in §5.1 are in parentheses.

better suit our task format, we modify their evaluation task to a binary pairwise comparison. The evaluation protocol of WildBench requires a task-specific checklist of output quality to aid the evaluation. In WildBench, these checklists were created using GPT-4-Turbo and Claude-3-Opus and manually reviewed. Following a similar approach, we use GPT-4o to generate these checklists for the 4 datasets used in this work.

### C.3 Prompts for Enhanced Evaluation Protocols

We list the prompts for cot (Figure 10), metric (Figure 11 & Figure 12), reference (Figure 13), metric+reference (Figure 14), and swap&synthesize (Figure 15). These prompt templates are proposed by Zeng et al. (2024).

### C.4 Prompts for Complex Evaluation Protocols

We present our prompts for fine-grained-diff (Figure 16), multi-role-round2 (Figure 17), multi-aspect-single (Figure 18), multi-aspect-two (Figure 19 and Figure 20), gpt4-reference (Figure 21), and prepair (Figure 22 and Figure 23).

## D Self-Agreement Rate of Different Evaluation Protocols

Table 9 shows the protocols self-agreement rates, demonstrating that (1) swap&synthesize can significantly enhance the self-agreement rate; (2) The multi-role debate protocols,

multi-role-round1&2, yield a significantly lower self-agreement rate, indicating that introducing more complex evaluation processes can lead to larger self-inconsistency.

## E Case Study

We perform a qualitative analysis of the evaluation performance of llama-3-70B and identify three main error patterns that impact its performance in various instances. The following paragraphs outline these error patterns, and we present a specific case study in Table 14.

**Surface-level deception (SURFACE)** The model tends to favor outputs that appear more positive or have more structured presentations like numbered lists or professional layout despite clear disadvantages in addressing the instruction task compared to less structured but more appropriate and accurate responses. This failure mode is a recognized pattern across LLM-evaluators (Zheng et al., 2024; Wang et al., 2024b).

**Overlooking crucial context (OVERLOOK)** The model sometimes fails to identify important context in the question when it is not explicitly emphasized. This can lead to generic responses that miss key elements specific to the task at hand.

**Counting issues (COUNTING)** This applies to both numerical counting (for example, “generate an 8-letter password”) and text-based context-relevant counting. The model often fails to count letters correctly or prefers longer outputs that exceed the instructed requirements, contrary to human annotators’ preference for accurate answers.

## F Generation Samples

We present generation samples across different evaluation protocols of llama-3-70B model for an example from LLMBench-Natural, as shown in Figure 24. Model evaluations of llama-3-70B under selected evaluation protocols are presented in Figure 25 (cot), Figure 26 (fine-grained-diff), Figure 27 (multi-role-round2), and Figure 28 (prepair). Our analysis of the evaluation protocols reveals interesting disparities in their effectiveness. For example, the cot fails because it overlooks the factual error regarding the frequency of the word “humans” in one of the candidate outputs. In contrast, multi-role-round2’s success can be attributed to its multi-perspective debate approach,

which allows for a more thorough examination and fact-checking of the claims made in each output.



<b>LLMBar-Natural</b>	
Explain: Pathos, Ethos, Logos.	
<b>Output 1</b> Ethos is the ethical appeal, which is used to convince an audience of the author's credibility or character. Pathos is the emotional appeal, which is used to persuade an audience by appealing to their emotions. Logos is the appeal to logic and is used to convince an audience by the use of logic and reason.	<b>Output 2</b> Pathos is a tool of persuasion that is used to appeal to readers' emotions by arousing positive or negative feelings. It can be used in rhetoric, literature, film, and other forms of expression. While pathos is used to draw an emotional response, the other rhetorical appeals appeal to credibility and logic, respectively. Ethos is the distinguishing character, sentiment, moral nature, or guiding beliefs of a person, group, or institution. It is often evident as facts, numbers, or statistics and is used to convince an audience of an argument.
<b>LLMBar-Adversarial</b>	
Name an animal that is commonly kept as a pet.	
<b>Output 1</b> Dogs are commonly kept as pets.	<b>Output 2</b> Dogs, cats, and fish.
<b>MTBench</b>	
A is the father of B. B is the father of C. What is the relationship between A and C?	
<b>Output 1</b> A is the grandfather of C.	<b>Output 2</b> * A is the father of B * B is the father of C * Therefore, A is the grandfather of C So the relationship between A and C is: Grandfather - Grandson
<b>InstruSum</b>	
Here is an article: It is a book with 277 authors but with one shared experience - becoming a new mother in 2020. The mothers speak about their isolation, ... Please summarize the above article based on the given requirement:	
<b>Output 1</b> Born in Lockdown is a book written by 277 new mothers in the UK about their experiences of becoming a mother during the pandemic. The mothers share their feelings of isolation, uncertainty, and pressure on their mental health, but also highlight the silver linings and positives that have come out of this joint experience. The project was set up by Bristol novelist Emylia Hall to encourage writing for pleasure and purpose. The book is an unflinchingly honest and moving account of new motherhood during the pandemic, where resilience, love, and hope shine through. The book is available for free download from the Mothership Writers website, with voluntary donations going to the stillbirth and neonatal death charity Sands.	<b>Output 2</b> Mothers in 2020 have experienced a unique set of challenges due to the pandemic, including isolation, uncertainty, and pressure on their mental health. Despite the hardships, many have found silver linings and positives that have come out of the shared experience. Through the writing project Born in Lockdown, 277 mothers from across the UK have shared their stories, expressing a range of emotions including resilience, love, and hope.

Table 10: Selected examples from our datasets (§3). The green label indicates the output is the preferred output.

---

**LLMBar-Natural**

---

Given the following input statement, provide the type of sentiment in the statement and the extent of sentiment in terms of sentiment scores. This restaurant has the best service I've ever seen. You can design the score scale by yourself.

---

**Output 1**

This statement has a sentiment score of 9 / 10

**Output 2**

Positive sentiment, sentiment score: 9.5 out of 10

---

**LLMBar-Adversarial**

---

Analyze the results of the following test scores. 90, 86, 75, 63

---

**Output 1**

The average grade is 78.75.

**Output 2**

The test scores show a decreasing trend, with the first score being the highest and the last score being the lowest. This could indicate that the test taker's knowledge and skills were increasingly weak as the test progressed.

---

**MTBench**

---

Identify the countries, their capitals, and the languages spoken in the following sentences. Output in JSON format.

- a) Amidst the idyllic vistas, Copenhagen, Denmark's capital, captivates visitors with its thriving art scene and the enchanting Danish language spoken by its inhabitants.  
b) Within the enchanting realm of Eldoria, one discovers Avalore, a grandiose city that emanates an ethereal aura. Lumina, a melodious language, serves as the principal mode of communication within this mystical abode.  
c) Nestled amidst a harmonious blend of age-old customs and contemporary wonders, Buenos Aires, the capital of Argentina, stands as a bustling metropolis. It is a vibrant hub where the expressive Spanish language holds sway over the city's inhabitants.
- 

**Output 1**

```
{ "sentences": [
  { "text": "Amidst the idyllic vistas, Copenhagen, Denmark's capital, captivates visitors with its thriving art scene and the enchanting Danish language spoken by its inhabitants.",
    "country": "Denmark",
    "capital": "Copenhagen",
    "language": "Danish" },
  { "text": "Within the enchanting realm of Eldoria, one discovers Avalore, a grandiose city that emanates an ethereal aura. Lumina, a melodious language, serves as the principal mode of communication within this mystical abode.",
    "country": "Eldoria (fictional)",
    "capital": "Avalore (fictional)",
    "language": "Lumina (fictional)" },
  { "text": "Nestled amidst a harmonious blend of age-old customs and contemporary wonders, Buenos Aires, the capital of Argentina, stands as a bustling metropolis. It is a vibrant hub where the expressive Spanish language holds sway over the city's inhabitants.",
    "country": "Argentina",
    "capital": "Buenos Aires",
    "language": "Spanish" }
]
```

**Output 2**

- a) The capital of Denmark is Copenhagen. The language spoken in Denmark is Danish.  
b) The capital of Eldoria is Avalore. The language spoken in Eldoria is Lumina.  
c) The capital of Argentina is Buenos Aires. The language spoken in Argentina is Spanish.
- 

**InstruSum**

---

Here is an article:

As pressure to tackle London's housing crisis grows, a number of controversial views are being offered on how the problem...

Please summarize the above article based on the given requirement:

---

**Output 1**

Research director at The Adam Smith Institute, Sam Bowman, suggests that building on London's green belt lands could provide bigger and cheaper homes for everyone. However, Ann Goddard of Elstree & Borehamwood Greenbelt Society argues that green belt lands are important for recreation purposes and to maintain a separation between settlements. Bowman proposes a compromise by differentiating between areas of beauty and intensive farmland within the green belt.

**Output 2**

One solution to London's housing crisis is to build more properties on green belt lands. This would mean bigger and cheaper homes for everyone, but there is opposition to this idea as green belt land is seen as important for recreation and to make a demarcation between settlements. A compromise could be reached by differentiating between areas of beauty and intensive farmland.

---

Table 11: Selected examples from our datasets (§3). The green label indicates the output is the preferred output.

Name	Size	License	Description
gemma-2b	2b	Gemma	Gemma is a family of open models from Google (Gemma et al., 2024)
gemma-7b	7b	Gemma	
glm-4-9b	9b	GLM-4	GLM-4-9B is an open-source version of the latest generation of pre-trained models launched by Zhipu AI (Du et al., 2022).
yi-1.5-9b	9b	Yi	Yi series are bilingual language models trained on a 3T multilingual corpus by 01.AI (AI et al., 2024)
yi-1.5-34b	34b	Yi	
llama-2-7b	7b	llama 2 Community	llama 2 models are the latest generation developed by Meta AI (Touvron et al., 2023), pretrained on 2.2T tokens.
llama-2-13b	13b	llama 2 Community	
llama-2-70b	70b	llama 2 Community	
llama-3-8b	8b	llama 3 Community	llama 3 are the latest open models from Meta AI (Meta AI, 2024), pretrained on 15T tokens.
llama-3-70b	70b	llama 3 Community	
llama-3.1-8b	8b	llama 3.1 Community	llama 3.1 collection offers a series of multilingual models that outperform many open and closed chat models on industry benchmarks (Dubey et al., 2024).
llama-3.1-70b	70b	llama 3.1 Community	
llama-3.1-405b	405b	llama 3.1 Community	
TÜLU-2-7b	7b	AI2 ImpACT Low-risk	
TÜLU-2-dpo-7b	7b	AI2 ImpACT Low-risk	TÜLU V2 (Iverson et al., 2023) is a series of llama 2 based models that are instruction-tuned on TÜLU V2 MIX.
TÜLU-2-13b	13b	AI2 ImpACT Low-risk	
TÜLU-2-dpo-13b	13b	AI2 ImpACT Low-risk	
TÜLU-2-70b	70b	AI2 ImpACT Low-risk	
TÜLU-2-dpo-70b	70b	AI2 ImpACT Low-risk	
gemini-1.0-pro	-	Proprietary	Gemini models are the most capable multimodal models from Google featuring long context lengths (Gemini et al., 2023).
gemini-1.5-flash	-	Proprietary	
gemini-1.5-pro	-	Proprietary	
qwen-1.5-32b	32b	Qianwen	Qwen is a family of models built by Alibaba Cloud (Bai et al., 2023). Qwen1.5 and Qwen2 have recently surpassed most open models on common benchmarks.
qwen-1.5-72b	72b	Qianwen	
qwen-2-72b	72b	Qianwen	
qwen-2.5-72b	72b	Qianwen	
mistral-7b-v0.3	7b	Apache 2.0	Instruction-tuned versions of Mistral models (Jiang et al., 2023) from Mistral AI.
mistral-large	-	Proprietary	
mixtral-8x7b	8x7b	Apache 2.0	Mixtral-8x22B is a pretrained generative Sparse Mixture of Experts (MoE) from Mistral AI (Jiang et al., 2024)
claude-3-haiku	-	Proprietary	Claude-3-Haiku and Claude 3 Opus, and Claude 3.5 Sonnet are top proprietary models trained by Anthropic PBC (Claude, 2023).
claude-3-opus	-	Proprietary	
claude-3.5-sonnet	-	Proprietary	
gpt-3.5-turbo-0125	-	Proprietary	GPT models are strong proprietary models (Achiam et al., 2023) from OpenAI. “o1” model was released in September 2024 with strong reasoning capability.
gpt-4-0613	-	Proprietary	
gpt-4o-2024-05-13	-	Proprietary	
gpt-4o-2024-08-06	-	Proprietary	
o1-mini-2024-0912	-	Proprietary	
prometheus-2-8x7b	8x7b	Apache 2.0	Prometheus 2 is an alternative to GPT-4 for fine-grained evaluation of LLMs and reward models for RLHF, based on Mistral-Instruct (Kim et al., 2024b)
offsetbias-lm	8b	llama 3 Community	OffsetBias is a generative judge model for pairwise preference evaluation, designed to be robust against various evaluation biases (Park et al., 2024)
nemotron-4-340b-reward	340b	NVIDIA Open Model	A multi-aspect reward model for synthetic data generation and RLAIIF, based on Nemotron-4-340B-Base (Wang et al., 2024d)
offsetbias-rm	8b	llama 3 Community	Reward model trained on OffsetBias dataset, designed to be robust against various evaluation biases (Park et al., 2024)

Table 12: Model registry and metadata in our study used in §4.

Full Name	Code	Related Work	# Qs./Eval	# Aux.Qs./Eval
Base Pairwise Evaluation	base	Zeng et al. (2024)	1	N/A
Chain-of-Thought	cot	Zeng et al. (2024); Wei et al. (2022)	1	N/A
Metric Generation	metric	Zeng et al. (2024)	1	1
Reference Generation	reference	Zeng et al. (2024)	1	1
Metric and Reference	metric+reference	Zeng et al. (2024)	1	2
Swap and Synthesize	swap&synthesize	Zeng et al. (2024)	2	N/A
Fine-grained Differences	fine-grained-diff	Inspired by Liu et al. (2023b,c); Min et al. (2023)	1	N/A
Multi-Role Debate (Single Round)	multi-role-round1	Chan et al. (2024)	3	N/A
Multi-Role Debate (Two Rounds)	multi-role-round2	Chan et al. (2024)	6	N/A
Multi-Aspect Comparison (Two-Stage)	multi-aspect-two	Li et al. (2024a)	1	5
Multi-Aspect Comparison (Single-Stage)	multi-aspect-single	Li et al. (2024a)	1	N/A
GPT-4 Reference	gpt4-reference	Modified from Zeng et al. (2024)	1	N/A
Pointwise Reasoning Enhanced Pairwise	prepair	Jeong et al. (2024)	1	2
Self-Consistency over CoTs	cot&consistency	Wang et al. (2023)	9	N/A
Self-Consistency over Different Protocols	protocol-consistency	Wang et al. (2023)	5	N/A

Table 13: Method registry for evaluation protocols in §5.1. #Qs./Eval indicates the number of queries required per evaluation, and #Aux.Qs./Eval refers to any additional auxiliary queries.

Base

**[System Message]**

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

**[User Message]**

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 6: Prompt for base protocol described in §5.1.

## AlpacaEval

### [System Message]

You are a highly efficient assistant, who evaluates and selects the best large language model (LLMs) based on the quality of their responses to a given instruction. This process will be used to create a leaderboard reflecting the most accurate and human-preferred answers.

### [User Message]

I require a leaderboard for various large language models. I'll provide you with prompts given to these models and their corresponding outputs. Your task is to assess these responses, and select the model that produces the best output from a human perspective.

#### ## Instruction

```
{  
  "instruction": """"{INSTRUCTION}""",  
}
```

#### ## Model Outputs

Here are the unordered outputs from the models. Each output is associated with a specific model, identified by a unique model identifier.

```
{  
  { "model_identifier": "m",  
    "output": """"{OUTPUT_1}""",  
  },  
  {  
    "model_identifier": "M",  
    "output": """"{OUTPUT_2}""",  
  }  
}
```

#### ## Task

Evaluate the models based on the quality and relevance of their outputs, and select the model that generated the best output. Answer by providing the model identifier of the best model. We will use your output as the name of the best model, so make sure your output only contains one of the following model identifiers and nothing else (no quotes, no spaces, no new lines, ...): Model (m) or Model (M).

## Which is better, Model (m) or Model (M)? Your response should be either "Model (m)" or "Model (M)":

Figure 7: Prompt for AlpacaEval baseline described in §4.2

ArenaHard

**[System Message]**

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user prompt displayed below. You will be given assistant A's answer and assistant B's answer. Your job is to evaluate which assistant's answer is better.

**[User Message]**

Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers.

When evaluating the assistants' answers, compare both assistants' answers with your answer. You must identify and correct any mistakes or inaccurate information.

Then consider if the assistant's answers are helpful, relevant, and concise. Helpful means the answer correctly responds to the prompt or follows the instructions. Note when user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. Relevant means all parts of the response closely connect or are appropriate to what is being asked. Concise means the response is clear and not verbose or excessive.

Then consider the creativity and novelty of the assistant's answers when needed. Finally, identify any missing important information in the assistants' answers that would be beneficial to include when responding to the user prompt.

After providing your explanation, you must always end your response with either "Therefore, Answer (a) is better." or "Therefore, Answer (b) is better." verbatim.

<{User Prompt}>  
{INSTRUCTION}

<{The Start of Answer (a)}>  
{OUTPUT\_1}  
<{The End of Answer (a)}>

<{The Start of Answer (b)}>  
{OUTPUT\_2}  
<{The End of Answer(b)}>

# Decision (Give an explanation of your evaluation followed by either "Therefore, Answer (a) is better." or "Therefore, Answer (b) is better." verbatim. Always claim which is better at the end.):

Figure 8: Prompt for ArenaHard described in §4.2

**[System Message]**

You are an expert evaluator. Your task is to evaluate the quality of the responses generated by two AI models.

**[User Message]**

# Instruction

We will provide you with the user query and a pair of AI-generated responses (Response A and Response B).

You should first read the user query and the conversation history carefully for analyzing the task, and then evaluate the quality of the responses based on and rules provided below.

# Conversation between User and AI

## User Query

<|begin\_of\_query|>

{INSTRUCTION}

<|end\_of\_query|>

## Response A

<|begin\_of\_response\_A|>

{OUTPUT\_1}

<|end\_of\_response\_A|>

## Response B

<|begin\_of\_response\_B|>

{OUTPUT\_2}

<|end\_of\_response\_B|>

# Evaluation

## Checklist

<|begin\_of\_checklist|>

{CHECKLIST}

<|end\_of\_checklist|>

Please use this checklist to guide your evaluation, but do not limit your assessment to the checklist.

## Rules

You should compare the above two responses based on your analysis of the user query.

You should first write down your analysis and the checklist that you used for the evaluation, and then provide your assessment according to the checklist.

You should always end your response by selecting the better response.

## Output Format

First, please output your analysis for each model response, and then summarize your assessment to three aspects: "reason A=B", "reason A>B", and "reason B>A", and finally make your choice for the final assessment by selecting the better response (ties are NOT allowed).

Please provide your evaluation results in the following json format by filling in the placeholders in []:

```

““
{
  "analysis of A": "[analysis of Response A]",
  "analysis of B": "[analysis of Response B]",
  "reason of A=B": "[where Response A and B perform equally well]",
  "reason of A>B": "[where Response A is better than Response B]",
  "reason of B>A": "[where Response B is better than Response A]",
  "choice": "[Response (A) or Response (B)]",
}
““

```

Figure 9: Prompt for WildBench baseline described in §4.2



CoT

**[System Message]**

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

**[User Message]**

After giving a brief explanation, select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

You should first provide a brief explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning. You should do reasoning and thinking before claiming which is better.

# Instruction:  
{INSTRUCTION}

# Output (a):  
{OUTPUT\_1}

# Output (b):  
{OUTPUT\_2}

# Decision (Give a brief explanation of your evaluation followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Always claim which is better at the end. In your explanation, you should always use "Output (a)" or "Output (b)" to refer to the two outputs respectively.):

Figure 10: Prompt for cot protocol described in §5.1.

Metric (metric generation prompt)

**[System Message]**

You are a helpful assistant.

**[User Message]**

Please propose at most three concise questions about whether a potential output is a good output for a given instruction. Another assistant will evaluate different aspects of the output by answering all the questions.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.

# Instruction:

{INSTRUCTION}

# Requirements for Your Output:

- (1) The questions should **\*\*specifically\*\*** target the given instruction instead of some general standards, so the questions may revolve around key points of the instruction.
- (2) You should directly give the questions without any other words.
- (3) Questions are presented from most important to least important.

# Please give your questions here:

Figure 11: Prompt for metric generation stage of the metric protocol described in §5.1.

## Metric

### [System Message]

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

### [User Message]

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:  
{INSTRUCTION}

# Output (a):  
{OUTPUT\_1}

# Output (b):  
{OUTPUT\_2}

# Questions about Outputs:

Here are at most three questions about the outputs, which are presented from most important to least important. You can do the evaluation based on thinking about all the questions.

{QUESTIONS}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 12: Prompt for metric protocol described in §5.1.

## Reference

### **[System Message]**

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

### **[User Message]**

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# A reference output generated by a strong AI assistant:

{REFERENCE}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 13: Prompt for reference protocol described in §5.1.

## Metric + Reference

### [System Message]

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

### [User Message]

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:  
{INSTRUCTION}

# Output (a):  
{OUTPUT\_1}

# Output (b):  
{OUTPUT\_2}

# Questions about Outputs:

Here are at most three questions about the outputs, which are presented from most important to least important. You can do the evaluation based on thinking about all the questions.

{QUESTIONS}

# A reference output generated by a strong AI assistant:  
{REFERENCE}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 14: Prompt for metric + reference protocol described in §5.1.

## Swap&Synthesize

### [System Message]

You are a helpful assistant who reviews a debate between two other assistants in evaluating the quality of the outputs for a given instruction.

### [User Message]

The two assistants, Assistant (a) and Assistant (b), are given an instruction, Output (a) and Output (b). They are asked to select the Output (a) or Output (b) that is better for the given instruction. Output (a) and Output (b) are generated by two different AI chatbots respectively.

Assistant (a) and Assistant (b) have conflicting evaluations. Your goal is to review their evaluations and give your final decision on which output is better.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

Now carefully review the instruction, Output (a), Output (b), and the debate between Assistant (a) and Assistant (b). Select the Output (a) or Output (b) that is better for the given instruction.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:

{INSTRUCTION}

# Output (b):

{OUTPUT\_2}

# Output (a):

{OUTPUT\_1}

# Debate between Assistant (a) and Assistant (b):

## Evaluation given by Assistant (a), who thinks Output (a) is better:

{EXPLANATION\_1}

## Evaluation given by Assistant (b), who thinks Output (b) is better:

{EXPLANATION\_2}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 15: Prompt for swap&synthesize protocol described in §5.1.

## Fine-grained-diff

### [System Message]

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

### [User Message]

After giving a detailed explanation, select either Output (a) or Output (b) as the better response for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

You should first provide a detailed explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning. You should do reasoning and thinking before claiming which is better.

Here is the evaluation plan:

1. Differences Identification: Enumerate the key fine-grained content differences observed between Output (a) and Output (b).
2. Explanation and Rationale: Provide explanations and rationale for which output better addresses the instruction by considering these differences, as well as other relevant factors such as relevance, completeness, coherence, and clarity.
3. Final Decision: End your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Provide your response in the following format:

""""

Differences Identification:

1. [Difference 1]
2. [Difference 2]
- ...
- N. [Difference N]

Explanation and Rationale: [Detailed explanation and rationale for your decision]

Final Decision: Therefore, Output (a)/Output (b) is better.

""""

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# Your Response (Give a detailed explanation of your evaluation followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Always claim which is better at the end. In your explanation, you should always use "Output (a)" or "Output (b)" to refer to the two outputs respectively.):

Figure 16: Prompt for fine-grained-diff protocol described in §5.1.

## Multi-role-round2

### [System Message]

You are a helpful assistant who evaluates the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

### [User Message]

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

You should first provide a brief explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning. You should do reasoning and thinking before claiming which is better.

There are a few other referees assigned the same task, it's your responsibility to discuss with them and think critically before you make your final judgement. You should avoid any potential bias and ensure that the order in which the responses were presented does not affect your judgment. Debate with others.

Always end your response with "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Make sure to make the claim to end your response.

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# Previous referees' arguments:

{CHAT\_HISTORY}

# Your role:

{ROLE\_DESCRIPTION}

# Decision (Give a brief explanation of your evaluation followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Always claim which is better at the end. In your explanation, you should always use "Output (a)" or "Output (b)" to refer to the two outputs respectively.):

Figure 17: Prompt for Multi-role-round2 protocol described in §5.1.



## Multi-aspect-single

### [System Message]

You are a helpful assistant who analyzes and evaluates the quality of two candidate outputs for a given instruction task based on a list of criteria, and makes a final decision on which output is better.

### [User Message]

Given an instruction and two responses, Output (a) and Output (b), each aiming to fulfill the task, your task is to carefully analyze and evaluate each output based on a list of criteria.

Here are some general rules for the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more or less than what the instruction asks for, as such outputs do NOT precisely execute the instruction. (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better one.

You should first provide an explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning. You should do reasoning and thinking before claiming which output is better.

Here are some criteria to consider:

1. Text Quality: The response should be fluent, well-structured, and free of spelling and grammatical errors. It should also be coherent, with a clear and logical flow of ideas.
2. Information Richness: The response is encouraged to provide rich, detailed and professional information, e.g. by providing examples, explanations, citations, and additional information. This criterion is not applicable if the user asks for a short or direct answer without additional information.
3. User Intention Inference: If the user's intention is not clearly expressed by the query, the response should provide some relevant information, do some reasonable inference and ask more information for clarification. This criterion is not applicable if the user's intention is clearly expressed by the query.
4. Accuracy: All contents provided or mentioned in the response should be accurate and correct.
5. Completeness of Instruction Following: For all key instructions (e.g., answer multiple questions or perform multiple tasks) and explicit constraints (e.g. word count, response length limit, word usage, output format, etc.) provided by the user, the response should be complete in following all of them without any omission.

Consider how well each output meets the list of criteria and provide a comparative analysis. After your analysis, make a final decision on which output is better overall. Provide a brief explanation of your evaluation, weighing the importance of each aspect, and make a final decision. Be mindful of the importance of each aspect in the context of the given instruction task, as some aspects may significantly influence the output's quality and relevance to the instruction, while others might be less critical.

Provide your response in the following format:

""

1. Text Quality:

[Your analysis]

2. Information Richness:

[Your analysis]

3. User Intention Inference:

[Your analysis]

4. Accuracy:

[Your analysis]

5. Completeness of Instruction Following:

[Your analysis]

[Your overall evaluation and explanation, followed by the final decision]

""

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# Decision (Give an explanation of your evaluation followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Always claim which is better at the end. In your explanation, you should always use "Output (a)" or "Output (b)" to refer to the two outputs respectively.):

Figure 18: Prompt for multi-aspect-single protocol described in §5.1.

```
Multi-aspect-two (aspect-wise analysis stage)

[System Message]
You are a helpful assistant who analyzes and evaluates the quality of the outputs for a given instruction based on specific criteria.

[User Message]
Please provide a detailed analysis and evaluation of the two outputs based on the following criteria:
{CRITERIA_TEXT}

Consider how well each output meets the criteria and provide a comparative analysis.

# Instruction:
{INSTRUCTION}

# Output (a):
{OUTPUT_1}

# Output (b):
{OUTPUT_2}

# Your analysis:
```

Figure 19: Prompt for multi-aspect-two protocol described in §5.1. This is the prompt for aspect-wise analysis (the first stage) within the method.

Multi-aspect-two (final evaluation stage)

**[System Message]**

You are a helpful assistant who makes a final decision on which output is better based on the analysis of multiple aspects.

**[User Message]**

You have been provided with an instruction and two outputs, along with an analysis of each output based on several key aspects.

Your task is to carefully consider the analysis for each aspect and make a final decision on which output is better overall. Provide a brief explanation of your evaluation, weighing the importance of each aspect, and make a final decision. Be mindful of the importance of each aspect item in the context of the given instruction task, because some aspects may significantly influence the output's quality and relevance to the instruction, while others might be less critical.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are equally likely to be the better.

You should first provide an explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning. You should do reasoning and thinking before claiming which is better.

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# Here are the aspect-wise analyses provided by another helpful critic:

{ANALYSIS\_HISTORY}

# Decision (Give a brief explanation of your evaluation followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim. Always claim which is better at the end. In your explanation, you should always use "Output (a)" or "Output (b)" to refer to the two outputs respectively.):

Figure 20: Prompt for multi-aspect-two protocol described in §5.1. This is the final evaluation prompt (the second stage) within the method.

#### GPT4 Reference

**[System Message]**

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

**[User Message]**

Select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation: (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.

(2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.

(3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are **\*\*equally likely\*\*** to be the better.

Do NOT provide any explanation for your choice.

Do NOT say both / neither are good.

You should answer using ONLY "Output (a)" or "Output (b)". Do NOT output any other words.

# Instruction:

{INSTRUCTION}

# Output (a):

{OUTPUT\_1}

# Output (b):

{OUTPUT\_2}

# A reference output generated by a strong AI assistant: {REFERENCE}

# Which is better, Output (a) or Output (b)? Your response should be either "Output (a)" or "Output (b)":

Figure 21: Prompt for gpt4-reference protocol described in §5.1.

### Prepair (pointwise analysis)

**[System Message]**

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to evaluate the quality of output for the given instruction.

**[User Message]**

Giving a brief explanation to evaluate the quality of the response to the given instruction. The output is generated by an AI chatbot.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) The model outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible.

You should provide a brief explanation of your evaluation.

Your explanation should identify critical drawbacks in model outputs that do not meet the above evaluation rules.

# Instruction:

{INSTRUCTION}

# Output:

{OUTPUT}

# Provide your explanation:

Figure 22: Prompt for prepair protocol described in §5.1. This is the prompt for pointwise analysis (the first stage) within the protocol.

### Prepair (pairwise evaluation)

#### [System Message]

You are a helpful assistant in evaluating the quality of the outputs for a given instruction. Your goal is to select the best output for the given instruction.

#### [User Message]

After giving a brief explanation, select the Output (a) or Output (b) that is better for the given instruction. The two outputs are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:

- (1) You should prioritize evaluating whether the output honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
- (2) Outputs should NOT contain more/less than what the instruction asks for, as such outputs do NOT precisely execute the instruction.
- (3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the outputs were presented should NOT affect your judgment, as Output (a) and Output (b) are **\*\*equally likely\*\*** to be the better.

You should first provide a brief explanation of your evaluation, and then always end your response with either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim.

Do NOT say both / neither are good.

Do NOT output any other words.

Do NOT say "Output (a) is better" or "Output (b) is better" at the beginning.

You should do reasoning and thinking **\*\*before\*\*** claiming which is better. Your explanation should identify critical drawbacks in model outputs that do not meet the above evaluation rules.

# Instruction:  
{INSTRUCTION}

# Output (a):  
{OUTPUT\_1}

# Output (b):  
{OUTPUT\_2}

# Here's the analysis for each output you wrote earlier:  
{PER OUTPUT ANALYSES}

# Your Response (Provide your evaluation and reasoning, followed by either "Therefore, Output (a) is better." or "Therefore, Output (b) is better." verbatim):

Figure 23: Prompt for prepair protocol described in §5.1. This is the pairwise evaluation stage (the second stage) within the method.

---

**Case 1**

---

**Instruction:** Imagine that you are a guest speaker giving a lecture. Write a 1 to 2 sentences introduction about yourself. It should be generalizable to different fields.

**Output (a)**

Hello everyone! My name is John Smith and I'm glad to be here today as a guest speaker. I'm a software engineer with over 10 years of experience ....

**Output (b)**

Good evening everyone! My name is \_\_\_\_\_ and I'm excited to be here today to discuss the importance of \_\_\_\_\_.

**Generation:**

Output (a) provides a specific and concrete introduction about the speaker, including their name and profession, which is relevant to the lecture. On the other hand, Output (b) leaves blank spaces for the name and topic, making it unclear and incomplete. **Therefore, Output (a) is better.**

**Comment:** Error type(s): surface, overlook, counting; llama-3-70b failed to recognize that Output (a) violates both the length constraint and the generalization requirement specified in the instruction. Instead, it preferred Output (a) because it appeared more complete than Output (b).

---

**Case 2**

---

**Instruction:** "Determine the most common word in the text. Humans are created in the image of God, from a spiritual perspective and from a physical perspective."

**Output (a)**

"from", "a", and "perspective"

**Output (b)**

The most common word in the text is "humans", which appears twice.

**Generation:**

The instruction asks to determine the most common word in the given text. Output (a) provides a list of words without specifying which one is the most common, whereas Output (b) clearly identifies the most common word as "humans" and provides the frequency of its appearance. Output (b) precisely executes the instruction by providing a single most common word, whereas Output (a) does not. **Therefore, Output (b) is better.**

**Comment:** Error type(s): surface, counting; The model failed to identify the error in Output (b), where "humans" only appears once and is not the most common word. It preferred Output (b) due to its better formatting, despite the underlying error.

---

**Case 3**

---

**Instruction:** "Write a script for a 10-second conversation between two characters."

**Output (a)**

Person A: Hi, I'm John. It's nice to meet you. Person B: Hi John, I'm Jane. It's nice to meet you too. .... (7 more turns)

**Output (b)**

Person A: Hi, how are you? Person B: I'm doing well, how about you? (1 more turn)

**Generation:**

The instruction asks for a script for a 10-second conversation between two characters. Output (a) provides a conversation that is too long and detailed for a 10-second conversation, as it includes multiple topics and exchanges. Output (b), on the other hand, provides a brief and concise conversation that can fit within a 10-second time frame. However, Output (b) does not provide a complete conversation, as it only includes two exchanges and does not provide a clear conclusion or ending. **Therefore, Output (a) is better.**

**Comment:** Error type(s): surface, overlook; llama-3-70b overlooked a clear violation of the instruction in Output (a), where the conversation exceeds the 10-second limit. Despite this, it preferred the output as the more "complete" conversation, even though completeness was not a required attribute in the instruction.

---

Table 14: Case study (Appendix §E) showing error patterns in llama-3-70b generations under cot protocol (§5.1).

### Instruction

Determine the most common word in the text. Humans are created in the image of God, from a spiritual perspective and from a physical perspective.

#### Output (A)

"from", "a", and "perspective"

#### Output (B)

The most common word in the text is "humans", which appears twice.

Figure 24: An instance from LLMBAR-Natural dataset. Output (A) is the gold preferred response.

### CoT

The instruction asks to determine the most common word in the given text. Output (a) provides a list of words without specifying which one is the most common, whereas Output (b) clearly identifies the most common word as "humans" and provides the frequency of its appearance. Output (b) precisely executes the instruction by providing a single most common word, whereas Output (a) does not. Therefore, Output (b) is better. ❌

Figure 25: llama-3-70b output under cot protocol.

### Fine-grained Differences

"" Differences Identification: 1. Output (a) provides a list of three words, while Output (b) provides a single word with its frequency. 2. Output (a) does not specify the frequency of the words, while Output (b) does. 3. The words listed in Output (a) are not necessarily the most common words in the text. Explanation and Rationale: The instruction asks to determine the most common word in the text. Output (b) directly addresses this instruction by providing the most common word, "humans", and its frequency, which is twice. On the other hand, Output (a) provides a list of words without specifying their frequencies, and it is unclear whether these words are the most common in the text. In fact, "from", "a", and "perspective" may not be the most common words in the text. Therefore, Output (b) is more relevant, complete, and accurate in addressing the instruction. Final Decision: Therefore, Output (b) is better. ❌

Figure 26: llama-3-70b output under fine-grained-diff protocol.

### Multi-role Debate

General Public: Upon evaluating the outputs, I notice that Output (a) provides a list of words without specifying which one is the most common, whereas Output (b) clearly identifies the most common word as "humans" and provides the frequency of its appearance. Output (b) precisely executes the instruction by providing a single most common word, whereas Output (a) does not. Therefore, Output (b) is better. ❌

Critic: Upon reviewing the outputs, I agree with the General Public's argument that Output (b) provides a clear answer to the instruction by identifying the most common word as "humans" and its frequency. However, I noticed that Output (b) is incorrect, as "humans" appears only once in the text. On the other hand, Output (a) provides a list of common words, but it does not specify which one is the most common. Although it does not precisely execute the instruction, it provides more accurate information about the common words in the text. Considering the accuracy and helpfulness of the outputs, I think Output (a) is more useful, even though it does not directly answer the question. Therefore, Output (a) is better. ❌

Scientist: As a Scientist, I carefully evaluated the outputs and the arguments presented by the previous referees. I agree with the General Public that Output (b) attempts to provide a clear answer to the instruction, but I also concur with the Critic that Output (b) is incorrect, as "humans" appears only once in the text. On the other hand, Output (a) provides a list of common words, but it does not specify which one is the most common. Although it does not precisely execute the instruction, it provides more accurate information about the common words in the text. Upon closer examination, I noticed that Output (a) actually provides a correct list of common words, including "from", "a", and "perspective", which all appear twice in the text. This information can be useful for further analysis. Considering the accuracy and helpfulness of the outputs, I think Output (a) is more useful, even though it does not directly answer the question. Therefore, Output (a) is better. ✅

Figure 27: llama-3-70b output under multi-role-round2 protocol.



### Prepair

#### **Pointwise analysis for Output (a):**

The output does not precisely execute the instruction, as it asks for the "most common word" but the model provides multiple words ("from", "a", and "perspective"). The instruction implies a single answer, but the output gives three different words. This does not meet the rule of providing exactly what the instruction asks for. A correct output should identify a single word as the most common word in the text.

#### **Pointwise analysis for Output (b):**

The output does not precisely execute the instruction. The instruction asks to determine the most common word in the text, but the output provides additional information ("which appears twice") that is not requested. However, the output is helpful and accurate in identifying the most common word as "humans" is not correct.

#### **Pairwise evaluation:**

Both outputs have their drawbacks. Output (a) does not precisely execute the instruction as it provides multiple words instead of a single most common word. On the other hand, Output (b) also does not precisely execute the instruction as it provides additional information that is not requested. However, Output (a) is closer to the correct answer as it lists some of the most common words in the text, whereas Output (b) provides an incorrect answer. Therefore, Output (a) is better. ✓

Figure 28: llama-3-70b outputs under prepair protocol. Outputs from the pointwise analysis stage and the pairwise evaluation stage are presented.