An Empirical Study of LLM-as-a-Judge: How Design Choices Impact Evaluation Reliability

Yusuke Yamauchi[*†]
The University of Tokyo
y_yamauchi@is.s.u-tokyo.ac.jp

Taro Yano[*]
NEC Corporation
taro_yano@nec.com

Masafumi Oyamada
NEC Corporation
oyamada@nec.com

Abstract

As large language models (LLMs) continue to advance, reliable evaluation methods are essential—particularly for open-ended, instruction-following tasks. LLM-as-a-Judge enables automatic evaluation using LLMs as evaluators, but its reliability remains uncertain. In this work, we analyze key factors affecting its trustworthiness, focusing on alignment with human judgments and evaluation consistency. Using BIGGENBench and EvalBiasBench, we study the effects of evaluation design, decoding strategies, and Chain-of-Thought (CoT) reasoning in evaluation. Our results show that evaluation criteria are critical for reliability, non-deterministic sampling improves alignment with human preferences over deterministic evaluation, and CoT reasoning offers minimal gains when clear evaluation criteria are present.

1 Introduction

In recent years, large language models (LLMs) have been evolving rapidly, demonstrating high performance across various tasks (OpenAI, 2023; Anthropic, 2024; Google, 2024) and exerting significant influence. In addition to the high-performing proprietary models, there have been active efforts to develop open, small and high-performance LLMs (Dubey et al., 2024; Yang et al., 2024; DeepSeek-AI et al., 2024; Abdin et al., 2024).

To compare these LLMs, it is necessary to evaluate their performance on various tasks. Open-ended evaluation is particularly required to measure response capabilities and instruction-following ability as chat assistants. LLM-as-a-Judge (Zheng et al., 2023) is a technique developed for open-ended evaluation, where an evaluator LLM measures the performance of benchmarked LLMs. This approach has the advantage of being lower-cost and faster than manual evaluation (Gu et al., 2024a).

However, despite its growing adoption, there remain open questions about the *reliability* of LLM-as-a-Judge. In particular, we investigate two essential properties to ensure its trustworthiness: 1. **Alignment with human judgments** (Li et al., 2024a), and 2. **Consistency of evaluation results** (Schroeder and Wood-Doughty, 2024; Wei et al., 2024). Without these properties, automatic evaluation using LLMs risks producing misleading conclusions about model performance.

In this work, we aim to identify key factors that affect the reliability of LLM-as-a-Judge. To this end, we conduct a series of empirical analyses using two public benchmarks—**BIGGENBench** (Kim et al., 2024) and **EvalBiasBench** (Park et al., 2024a)—which provide a diverse set of open-ended tasks. Through systematic experiments, we investigate the impact of 1. the presence or absence of **reference answers** and **score descriptions** in evaluation prompts, 2. the choice of **decoding strategy** (greedy vs. sampling) used by the evaluator model, and 3. the role of **CoT** in the evaluator's response.

Our findings reveal that:

1. **Evaluation design:** Providing both reference answers and score descriptions is crucial for reliable evaluation. Omitting either significantly degrades alignment with human judgments, especially for weaker evaluator models. Furthermore, providing descriptions only for the highest and lowest scores yields the most reliable results, suggesting that the necessity of descriptions for intermediate scores should be reconsidered.

2. **Decoding strategy:** Greedy decoding ensures zero score variance, but it tends to show lower correlation with human judgments compared

[*]Equal contribution.
[†]Work done during an internship at NEC Corporation.

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 167–176
July 4, 2026 ©2026 Association for Computational Linguistics

to sampling-based decoding. Sampling introduces variability in scores, but it better captures the nuances of human preferences. Furthermore, averaging scores aligns with human judgments the most among compared three aggregation methods.

3. **Use of CoT reasoning:** When well-defined score descriptions are available, including CoT reasoning in evaluator responses has little effect on alignment with human judgments. From both a cost and performance perspective, CoT-free scoring combined with score averaging provides strong alignment with human evaluations while maintaining low computational cost.

2 Related Work

Evaluation of LLMs. Evaluating LLMs for generative tasks involves significant manual costs, leading to autonomous evaluation methods. Traditional approaches measure similarity between model outputs and references using lexical features (BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015)) or semantic features (BERTScore (Zhang et al., 2020), COMET (Rei et al., 2020)). However, these methods struggle with tasks allowing diverse valid responses. LLM-as-a-Judge (Zheng et al., 2023) addresses this by using capable models like GPT-4 as evaluators, employing Single Answer Grading (1-10 scoring) or Pairwise Evaluation (ranking multiple outputs) (Doddapaneni et al., 2024). MT-Bench (Zheng et al., 2023) assesses multi-turn capabilities using Single Answer Grading with reference answers. AlpacaEval 2.0 (Dubois et al., 2024) uses Pairwise Evaluation to mitigate length bias. Arena-Hard (Li et al., 2024b) filters ChatbotArena prompts for quality and diversity. BIGGEN-Bench (Kim et al., 2024) provides instance-specific criteria improving human judgment correlation.

Alignment with human judgments in LLM-as-a-Judge. Various approaches improve alignment with human judgments, including CoT reasoning, self-generated criteria, and multiple evaluations (Zheng et al., 2023; Zeng et al., 2024). Other methods optimize prompts using human annotation correlation (Liu et al., 2023b, 2024b) or employ ensemble voting (Liu et al., 2023a). Gu et al. (2024b) proposed metacognitive re-evaluation for consistency. Our study utilizes simple methodologies from a neutral standpoint to analyze the impact of evaluation design, decoding strategies, and CoT reasoning on alignment with human judgments.

Consistency of evaluation results in LLM-as-a-Judge. Existing studies have identified various biases where semantically unchanged modifications affect evaluation results. Chen et al. (2024) examined gender, authority, and aesthetic biases. Ye et al. (2024) identified 12 major latent biases including positional and self-enhancement bias. Park et al. (2024b) highlighted challenges with response length variations and content continuity. This paper extensively investigates how much evaluation results can fluctuate depending on the design of the evaluation tasks and the evaluation strategies.

3 Experiments

In this section, we examine what factors affect the alignment with human judgments and consistency of evaluation results. Research Questions (RQs) we aim to investigate are as follows:

1. Which components of evaluation design facilitate improved alignment with human judgments and enhance the consistency of evaluation results?

2. What are the advantages and disadvantages of deterministic versus non-deterministic decoding strategies?

3. Does CoT improve alignment with human judgments and the consistency of evaluation results?

3.1 Experimental Method

We describe the experimental methods to investigate the RQs.

Alignment with human judgments. To measure the degree of alignment with human judgments, we compute the correlation coefficient between the scores provided by humans and those generated by an evaluator LLM.

Consistency of evaluation results. We use Krippendorff's alpha coefficient to evaluate consistency of evaluation results, denoted as α. The α value, which indicates the consistency and reliability of evaluations, is 1 for perfect agreement, 0 for random annotations, and negative for systematic disagreement (see Appendix A for details).

Datasets. We adopt BIGGEN-Bench (Kim et al., 2024), which includes nine tasks such as instruction following, tool use, and reasoning, each with

Template for Evaluation Prompt

```
###Task Description:
An instruction (which may include an Input), a response to evaluate, a reference answer scoring 5, and
    a score rubric representing evaluation criteria are provided.
1. Write detailed feedback assessing the response strictly based on the score rubric.
2. After the feedback, provide an integer score from 1 to 5, referring to the rubric.
3. The output format should be: "(write feedback for criteria) [RESULT] (an integer between 1 and 5)"
4. Do not include any additional introductions, conclusions, or explanations.
###The instruction to evaluate:
{instruction}
###Response to evaluate:
{response}
###Reference Answer (Score 5):
{reference answer}
###Score Rubrics:
[{evaluation axes}]
Score 1: {score1_description}
Score 2: {score2_description}
Score 3: {score3_description}
Score 4: {score4_description}
Score 5: {score5_description}
###Feedback:
```

Figure 1: Prompt template used in our experiments to evaluate responses based on provided reference answers and evaluation criteria. The evaluation criteria consist of evaluation axes, which define general evaluation principles, and score descriptions, which provide rubrics for each of the five scores (1 through 5).

detailed, hand-crafted evaluation criteria. The evaluation template used in our experiments is shown in Figure 1. We also use EvalBiasBench (Park et al., 2024a), an instruction-following benchmark with both correct and biased answers. Since this benchmark does not include evaluation criteria or reference answers by default, we generated them using GPT-4o-2024-08-06. Regarding the score descriptions in the evaluation criteria, we designed them to encourage lower scores for biased responses. We utilized 765 instances from BIGGEN-Bench and 80 instances from EvalBiasBench for our experiments.

Models. We use GPT-4o-2024-08-06 as the evaluator LLM. Furthermore, considering recent studies on self-improvement (Yuan et al., 2024; Madaan et al., 2023) that use local LLMs as evaluators (Song et al., 2024; Kamoi et al., 2024), we also use LLaMA-3.1-70B-Instruct [1] (Dubey et al., 2024) as the evaluator LLM.

3.2 Results

RQ1. Which components of evaluation design facilitate improved alignment with human judgments and enhance the consistency of evaluation results? As shown in Table 1, removing either the evaluation criteria or the reference answer leads to a decrease in correlation with human judgments. For GPT-4o, the correlation drops from 0.666 to 0.591 and 0.638, respectively, while for LLaMA3.1-70B-Instruct, it drops from 0.641 to 0.555 and 0.581. This indicates that, regardless of the evaluator LLM used, the evaluation criteria have a greater impact than the reference answer. Furthermore, the degradation in correlation is more pronounced for LLaMA3.1 than for GPT-4o. When both the evaluation criteria and the reference answer are removed, the correlation with human judgment declines significantly and reaches its minimum.

Regarding evaluation consistency, in the BIGGEN-Bench dataset, removing the evaluation criteria or the reference answer does not substan-

Table 1: Experimental results for RQ1 report Krippendorff's alpha coefficients across five sampled scores, with values in parentheses indicating the correlation with human evaluation. Removing evaluation criteria (*w/o crt*) or reference answers (*w/o ref*) reduces human correlation. Eliminating both (*w/o ref&crt*) increases score fluctuation and significantly lowers human correlation.

	BIGGEN-Bench	
Method	GPT-4o	LLaMA3.1
Default	0.908 (0.666)	0.806 (0.641)
w/o crt	0.909 (0.591)	0.807 (0.555)
w/o ref	0.921 (0.638)	0.824 (0.581)
w/o ref&crt	0.896 (0.487)	0.758 (0.346)
	EvalBiasBench	
Method	GPT-4o	LLaMA3.1
Default	0.865	0.768
w/o crt	0.839	0.725
w/o ref	0.869	0.787
w/o ref&crt	0.811	0.753

[1] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

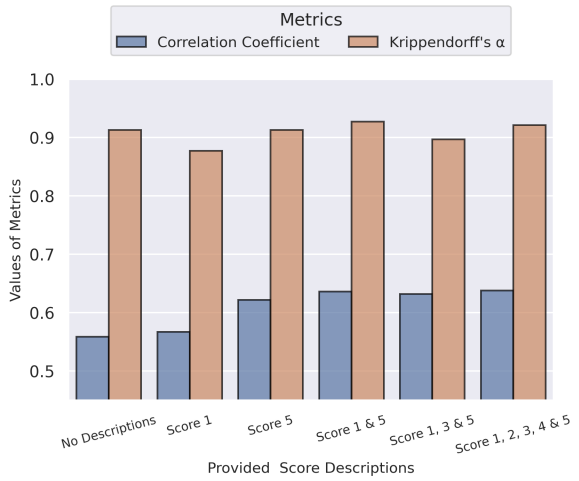

Figure 2: Additional experimental results for RQ1, showing the correlation coefficient and Krippendorff's α when parts of the score descriptions are removed from the evaluation criteria. When only the descriptions for scores 1 and 5 are provided (*Score 1 & 5*), the results exhibit the highest correlation with human evaluations while maintaining high evaluation consistency. This suggests that the role of score descriptions for intermediate scores (2, 3, and 4) should be reconsidered.

tially affect consistency. However, in EvalBias-Bench, removing the evaluation criteria leads to a noticeable drop in consistency. This suggests that, in EvalBiasBench, the absence of explicit criteria for penalizing biased responses may result in inconsistent scoring—potentially depending on random factors. Therefore, clearly defining scoring criteria for biased responses is crucial to ensure consistent evaluation.

Figure 2 illustrates the additional experimental results, which examines correlations and score fluctuation when removing a part of score descriptions from evaluation criteria. The figure shows there is little difference in both correlation with human judgments and score consistency between the setting where only the descriptions for scores 1 and 5 are provided and the setting where descriptions for all scores (1, 2, 3, 4, and 5) are given. These results suggest that the descriptions for intermediate scores (2, 3, and 4) have limited impact on alignment with human judgments, and their role should be reconsidered. It is also surprising that evaluation consistency remains generally high across all settings, indicating that even without detailed score descriptions, evaluations tend to remain consistent as long as general evaluation axes are provided.

RQ2. What are the advantages and disadvantages of deterministic versus non-deterministic decoding strategies?

Table 2: Experimental results for RQ2. Non-deterministic scoring methods (Majority, Median, Mean) show larger correlations with human judges compared to deterministic decoding (Greedy). Among the non-deterministic methods, score averaging (Mean) shows the largest correlations with human judges consistently across different evaluator LLMs, reasoning types, and evaluation design.

Method	Default	w/o crt	w/o ref	w/o ref&crt
GPT-4o				
Greedy	0.635	0.571	0.614	0.466
Majority	0.647	0.583	0.627	0.480
Median	0.648	0.581	0.621	0.481
Mean	**0.666**	**0.591**	**0.638**	**0.487**
GPT-4o w/o CoT				
Greedy	0.636	0.507	0.612	0.378
Majority	0.643	0.545	0.627	0.406
Median	0.651	0.546	0.629	0.399
Mean	**0.664**	**0.570**	**0.641**	**0.422**
LLaMA3.1				
Greedy	0.593	0.524	0.551	0.273
Majority	0.625	0.519	0.555	0.297
Median	0.624	0.520	0.558	0.297
Mean	**0.641**	**0.555**	**0.581**	**0.346**

We compare the correlation of scores with human judges between non-deterministic decoding and deterministic decoding on BIGGEN-Bench. For non-deterministic decoding, we sample five scores and aggregate them using majority voting (Majority), taking the median (Median), and averaging scores (Average). For deterministic decoding, we adopt greedy decoding (Greedy).

Table 2 shows the results. Non-deterministic scoring methods show larger correlations with human judges compared to deterministic decoding consistently. This finding is consistent with the fact that, in general inference tasks, multiple sampling and aggregation of results outperforms greedy decoding (Wang et al., 2023). More interestingly, among non-deterministic decoding methods, averaging scores shows the highest correlation with humans regardless of the evaluator LLM, evaluation design, or presence of CoT. This can be attributed to the fact that averaging allows for expressing fine-grained nuances, such as 4.5 when an evaluator LLM is torn between scores of 4 and 5, whereas median or majority voting methods round the score to either 4 or 5, thus failing to fully leverage the LLM's capabilities as an evaluator. Overall, employing multiple sampling yields higher accuracy than deterministic evaluation, though it leads to in-

creased computational costs. Therefore, a practical system design would involve utilizing sampling for precise evaluation specifically when dealing with complex tasks or sophisticated model responses.

RQ3. Does CoT improve alignment with human judgments and the consistency of evaluation results? To investigate the impact of CoT in LLM-as-a-Judge, we used GPT-4o to examine the correlation with human judges and the consistency of scores in two settings: one where a score was output after a Chain-of-Thought (w/ CoT), and one where only the score was output directly without any reasoning (Direct). Table 3 shows the results. In the Default setting, where evaluation criteria and reference answers are provided, both methods show similar correlation and consistency. Thus, when well-defined score descriptions are available, including explicit CoT in evaluator responses has little effect. From both a cost and performance perspective, direct scoring combined with score averaging provides strong alignment with human evaluations while maintaining low computational cost.

Table 3: Experimental results for RQ3. When given evaluation criteria and a reference answer (Default), scoring with CoT reasoning (w/ CoT) achieves comparable alignment with human judgments and evaluation consistency to Direct scoring (Direct).

	BIGGEN-Bench	
Method	Direct	w/ CoT
Default	0.912 (0.664)	0.908 (0.666)
w/o crt	0.818 (0.570)	0.909 (0.591)
w/o ref	0.910 (0.641)	0.921 (0.638)
w/o ref&crt	0.833 (0.422)	0.896 (0.487)
	EvalBiasBench	
Method	Direct	w/ CoT
Default	0.855	0.865
w/o crt	0.856	0.839
w/o ref	0.647	0.869
w/o ref&crt	0.650	0.811

4 Conclusion

In this work, we conducted a comprehensive empirical analysis to identify key factors affecting the reliability of LLM-as-a-Judge. Through systematic experiments on BIGGENBench and EvalBiasBench, we found that: (1) comprehensive evaluation design with both reference answers and score descriptions is essential for human alignment; (2) sampling-based scoring with mean aggregation outperforms scoring with greedy decoding; and (3) CoT reasoning provides diminishing returns when detailed evaluation criteria are present. These findings help establish best practices for reliable automatic evaluation and provide a principled framework for LLM-as-a-Judge deployment.

5 Limitations

While our study provides valuable insights, we acknowledge several limitations. First, we used GPT-4o and LLaMA-3.1-70B-Instruct as evaluator LLMs, representing a closed model and an open model, respectively, and obtained consistent results. However, it remains unclear whether consistent results can be obtained when using other LLMs as evaluators.

Additionally, the benchmarks used in this study employed different evaluation criteria: BIGGEN-Bench used human-crafted criteria, while EvalBiasBench relied on criteria generated by an LLM. Benchmarks accompanied by detailed evaluation criteria and reference answers are relatively rare; consequently, we utilized only two benchmarks in our experiments, which may be insufficient to ensure the generalizability of our results. Conducting experiments with evaluation criteria developed through more diverse methods and across a wider range of datasets is an important future direction. Furthermore, since the data used in our experiments were exclusively in English, verification in other languages would also be an essential area for future research.

6 Ethics Statement

This study primarily utilizes publicly available and properly licensed datasets and models. We have verified compliance with the terms of use for all datasets to ensure there are no legal or ethical conflicts. All models were used exclusively for inference without additional training, and their outputs were used solely for analysis. While certain models (e.g., GPT-4o) may present reproducibility challenges, we ensured the robustness of our results by employing multiple models for cross-validation. Finally, AI tools were used for proofreading and translation assistance; however, all content has been thoroughly reviewed and verified by the human authors.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Anthropic. 2024. Claude 3.5 sonnet.

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34:555–596.

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2024. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36.

Guiming Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? A study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 8301–8327. Association for Computational Linguistics.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

Sumanth Doddapaneni, Mohammed Safi Ur Rahman Khan, Sshubam Verma, and Mitesh M. Khapra. 2024. Finding blind spots in evaluator llms with interpretable checklists. *Preprint*, arXiv:2406.13439.

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, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, 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, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. 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, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *CoRR*, abs/2404.04475.

Google. 2024. Our next-generation model: Gemini 1.5.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and

Jian Guo. 2024a. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024b. A survey on llm-as-a-judge. *ArXiv*, abs/2411.15594.

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, Lélio 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. *Preprint*, arXiv:2401.04088.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can llms actually correct their own mistakes? A critical survey of self-correction of llms. *CoRR*, abs/2406.01297.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, Se June Joo, Miyoung Ko, Yoonjoo Lee, Hyungjoo Chae, Jamin Shin, Joel Jang, Seonghyeon Ye, Bill Yuchen Lin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models. *Preprint*, arXiv:2406.05761.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024a. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *CoRR*, abs/2412.05579.

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. *CoRR*, abs/2406.11939.

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

Chengyuan Liu, Fubang Zhao, Lizhi Qing, Yangyang Kang, Changlong Sun, Kun Kuang, and Fei Wu. 2023a. Goal-oriented prompt attack and safety evaluation for llms. *Preprint*, arXiv:2309.11830.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024a. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023b. Calibrating llm-based evaluator. *Preprint*, arXiv:2309.13308.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024b. HD-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7641–7660, Bangkok, Thailand. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024a. Offsetbias: Leveraging debiased data for tuning evaluators. *Preprint*, arXiv:2407.06551.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024b. Offsetbias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1043–1067. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *Preprint*, arXiv:2009.09025.

Kayla Schroeder and Zach Wood-Doughty. 2024. Can you trust LLM judgments? reliability of llm-as-a-judge. *CoRR*, abs/2412.12509.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham M. Kakade, Dean P. Foster, and Udaya Ghai. 2024. Mind the gap: Examining the self-improvement

capabilities of large language models. *CoRR*, abs/2412.02674.

Qwen Team. 2024. Introducing qwen1.5.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges. *Preprint*, arXiv:2406.12624.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. *Preprint*, arXiv:1411.5726.

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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hui Wei, Shenghua He, Tian Xia, Andy Wong, Jingyang Lin, and Mei Han. 2024. Systematic evaluation of llm-as-a-judge in LLM alignment tasks: Explainable metrics and diverse prompt templates. *CoRR*, abs/2408.13006.

Ziyou Yan. 2024. Evaluating the effectiveness of llm-evaluators (aka llm-as-judge). *eugeneyan.com*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V. Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *CoRR*, abs/2410.02736.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

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*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Krippendorff's alpha coefficient

In LLM-as-a-Judge, commonly used metrics for measuring inter-rater agreement include Pearson's correlation coefficient, Spearman's rank correlation coefficient, and Cohen's Kappa (Bai et al., 2024; Liu et al., 2024a; Thakur et al., 2025). However, these metrics have been criticized for reliability issues and their inability to handle various measurement scales or missing data (Yan, 2024; Artstein and Poesio, 2008). Krippendorff's alpha is a general statistical measure that encompasses multiple agreement metrics and can be applied to various tasks (Krippendorff, 2011). Krippendorff's alpha coefficient is defined as follows:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{1}$$

Here, D_o represents the total disagreement between pairs of ratings observed in the dataset. It is calculated using the coincidence matrices o_{ck}, the total sample size N, the total number of rating pairs n for the j-th unit, and the difference function $\delta(c,k)$, which quantifies the discrepancy between a given rating pair (c,k). The formula is expressed as:

$$D_o = \frac{1}{n} \sum_{c} \sum_{k} o_{ck} \delta_{ck} \tag{2}$$

In this study, we assume that $c, k \in \{1,2,3,4,5\}$. The coincidence matrix o_{ck} is defined as:

$$o_{ck} = \sum_{j} \frac{\text{Number of c-k pairs in unit } u}{\text{Total number of judges in unit } j-1} \tag{3}$$

The difference function $\delta(c,k)$, assuming an interval scale for the scores, is given by:

$$\delta_{ck} = (c-k)^2 \tag{4}$$

D_e represents the expected disagreement under a random distribution of ratings, computed as:

$$D_e = \frac{1}{n(n-1)} \sum_c n_c \sum_k n_k \delta_{ck} \tag{5}$$

Here, n_c and n_k denote the respective frequencies of the ratings. The alpha coefficient approaches 1 as different raters assign similar scores to the same unit.

B Experimental Details

B.1 Hyperparameters

Hyperparameters used to judge the responses are listed in Table 4. We retry the judging process while changing the seed value until a valid score is output for all samples.

Table 4: Hyperparameters used during inference time

hyperparameter	value
Seed	{0, 10, 20, 30, 40}
Max Seq Length	8192
Temperature	1.0
Repetition penalty	1.03

B.2 Response Model

We used the responses generated by Mixtral-8x7B-Instruct-v0.1 (Jiang et al., 2024), Qwen1.5-7B (Team, 2024), and GPT-3.5-turbo as inputs for the Judge.

B.3 Details of Prompting Strategies

w/o crt prompt template. When we remove elements from the evaluation criteria, we omit the corresponding items from the prompt template. In the w/o crt setting, we use the MT-Bench prompt (Zheng et al., 2023) as the base prompt.

Direct prompt template. When conducting the judging process in the Direct setting, only the score should be output without generating rationale. Therefore, we directly add **"### Feedback: [Result]"** at the end of the prompt.

MT-Bench prompt template

```
###Task Description:
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant
     to the user question displayed below.
You will be given a reference answer and the assistant's answer.
Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity
     , and level of detail of the response.
Begin your evaluation by providing a short explanation.
Be as objective as possible. After providing your explanation, please rate the response on a scale of 1
      to 5 by strictly following this format: [RESULT] (an integer number between 1 and 5)

###The instruction to evaluate:
{instruction}

###Reference Answer:
{reference_answer}

###Assistant's Answer to evaluate:
{response}

### Feedback:
```

Figure 3: Prompt template used in the w/o crt setting.