

REAL-JUDGE: When Should We Trust LLM Judges? An Exploration of Uncertainty Measurement

Anonymous ACL submission

Abstract

The widespread use of LLM-as-a-judge raises concerns about their reliability and limitations in real-world applications. Existing studies have explored LLM-as-a-judge in both subjective and objective scenarios, but challenges still exist due to limited benchmark diversity, inherent data biases (e.g., length and style biases), and the lack of metrics to assess whether LLM-as-a-judge truly understands their own judgement boundaries. To address these shortcomings, we propose REAL-JUDGE, which contains 1,280 samples spanning 7 task types, specifically designed to minimize common evaluation biases. We also adopted more comprehensive evaluation methods, which enable us to effectively assess the calibration of LLM-as-a-judge. Our results reveal that even state-of-the-art models exhibit poor calibration and that different types of LLM-as-a-judge excel in distinct task categories, underscoring the need for context-specific model selection. In conclusion, we provide a bias-free dataset and a reliable method for evaluating LLM-as-a-judge.

1 Introduction

The rapid advancement of large language models (OpenAI et al., 2024; Anthropic, 2024; Team et al., 2023; Touvron et al., 2023; DeepSeek-AI et al., 2024; Yang et al., 2024) (LLMs) has led to their increasing adoption as automated evaluators, a paradigm known as LLM-as-a-judge (Zheng et al., 2023). By leveraging LLMs to assess outputs, this approach offers a scalable alternative to costly and time-consuming human judgments. LLM-as-a-judge has been widely integrated into key areas of LLM development, including reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022) and the construction of evaluation benchmarks (Bai et al., 2022; Ou et al., 2023; Bai et al., 2024). However, as the use of LLM-as-a-judge becomes more prevalent, questions about its reliability and limitations grow increasingly pressing

(Doddapaneni et al., 2024). It is therefore critical to investigate the conditions under which LLM-as-a-judge can be reliably applied and to identify the specific domains where it excels.

To address these issues, several studies have systematically evaluated the performance of LLM-as-a-judge across diverse scenarios. For instance, studies such as FairEval (Wang et al., 2023) have focused on open-ended tasks, where subjective judgment and nuanced understanding are required. More recently, efforts such as JudgeBench (Tan et al., 2024) have begun to explore objective scenarios, where precise, fact-based assessments are critical, expanding the scope of investigation into tasks with clear ground truths. Despite these advancements, many benchmarks are limited in scale and diversity, failing to fully capture the complexity of real-world applications. Additionally, inherent biases in the data—such as length bias (Dubois et al., 2024) and style bias (Gallegos et al., 2024; Panickssery et al., 2024)—can skew evaluation outcomes and undermine the generalizability of the findings. These shortcomings highlight the need for more rigorous and inclusive benchmarking frameworks to ensure a fair and accurate assessment of LLM-as-a-judge’s capabilities.

Beyond these technical limitations, a deeper challenge lies in understanding whether language models truly *know what they know* (Kadavath et al., 2022; Panickssery et al., 2024)—a question that existing metrics fail to address. Although the widespread use of LLM-as-a-judge reduces labor costs in data construction, human oversight remains essential to ensure quality. This necessity stems from the fundamental uncertainty surrounding the self-awareness of language models, as traditional metrics like accuracy do not capture their ability to gauge their own knowledge and uncertainty. To address this gap, we introduce calibration to assess whether LLM-as-a-judge can accurately align their confidence with their actual performance. Cal-

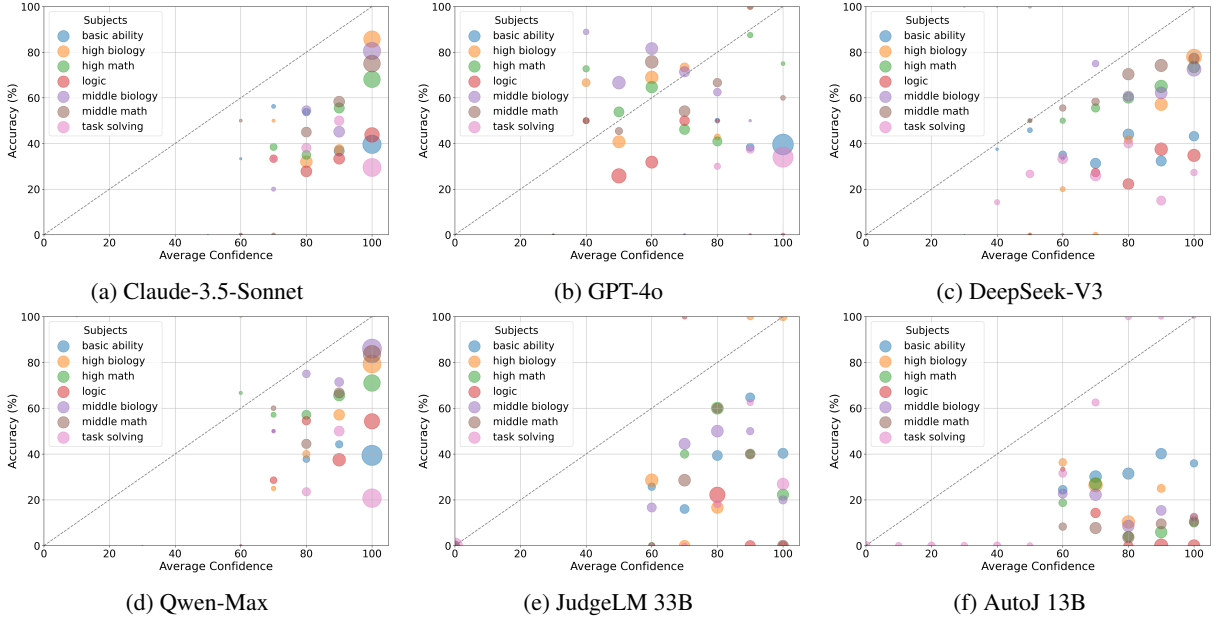


Figure 1: Reliability Diagram (MP setting)

ibration provides a deeper insight into the trustworthiness of LLM-as-a-judge, ultimately bridging the gap between automated evaluation and human-level judgment.

Our work advances the understanding and application of LLM-as-a-judge through the following key contributions:

- We introduce **REAL-JUDGE** (Reliable Exam Assessment for LLM-as-a-Judge), a comprehensive dataset of 1,280 pairs across 7 task types, designed to mitigate biases like length and style bias, enabling robust evaluation across diverse scenarios.
- We rigorously evaluate widely-used proprietary and open-source models, including finetuned variants, on accuracy (Acc) and calibration (Section 3), providing a holistic view of LLM-as-a-judge capabilities and limitations.
- Our analysis reveals that even the strongest models exhibit poor calibration as judges, highlighting reliability gaps. We also find that different LLM-as-a-judge types excel in distinct task categories, emphasizing the need for context-specific model selection.

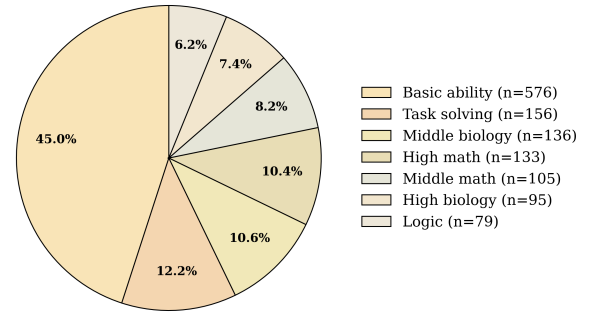


Figure 2: Distribution of topics in the dataset.

2 Data Construction

2.1 Data Composition and Sources

Our dataset comprises 1,280 positive-negative pairs, categorized into **objective category** and **subjective category**. Objective category is further divided into five subcategories: **logic**, **high school biology**, **middle school biology**, **middle school mathematics**, and **high school mathematics**. Subjective category includes **task solving** and **basic ability**. The distribution of categories is shown in Fig. 2.

The objective category is adapted from Chinese examinations, including the Gaokao (National College Entrance Examination), Zhongkao (Senior High School Entrance Examination), and the National Civil Service Examination (Guokao). The subjective category is derived from professional LLM evaluation team, with both types of questions

modified for use in this dataset.

2.1.1 Objective Category Annotation

For the objective category, which consists of multiple-choice questions (MCQs) selected from real examinations, we adopted two approaches to rewrite them to reduce the risk of data leakage and ensure the uniqueness of our dataset:

- **Question Rewriting:** We modified specific elements of the question, such as numerical values or operational relationships, to shift the correct answer to a different option while maintaining the core logic of the question.
- **Option Rewriting:** We introduced a new correct option and a new distractor to replace the original correct option, ensuring the original answer key is no longer valid.

After rewriting, we prompted models to generate answers and their corresponding analyses. Since the models lack access to the correct information, their outputs may fall into two scenarios:

- Both the answer and the analysis are incorrect.
- The answer is correct, but the analysis is incorrect.

These scenarios are particularly suitable for evaluation using the LLM-as-judge approach, as they present challenges that traditional rule-based methods struggle to handle effectively. For more details about the annotation guidelines, please refer to Appendix A.4.

2.1.2 Subjective Category Annotation

For the subjective category, each question was answered by over 50 models, and each response was labeled by three annotators as either “qualified” or “unqualified.” The final label was determined by majority voting. To construct positive-negative pairs, we paired a qualified response with an unqualified response of similar length for each question. This approach ensures a balanced comparison while mitigating potential biases related to response length.

2.2 Practical Considerations and Bias Mitigation

In constructing our dataset, we focused on mitigating biases that could affect judge model evaluation. Key measures include:

- **Length Bias Mitigation:** We enforced a strict criterion where the token length difference between positive and negative examples is less than 20%. Token counts were calculated using the tiktoken library¹.
- **Style Bias Mitigation:** We employed more than 50 models to generate negative examples, avoiding reliance on a single model’s output style. This prevents models from favoring specific patterns or phrasing, promoting a balanced evaluation based on reasoning quality.

3 Experiments

3.1 Metrics

To comprehensively evaluate model calibration, we use the Expected Calibration Error (ECE) metric (Guo et al., 2017), a widely used measure of model consistency. The ECE is calculated as follows:

$$ECE = \sum_{i=1}^{10} \frac{|B_i|}{n} |\text{acc}(B_i) - \text{conf}(B_i)|$$

where B_i represents the i -th confidence interval (divided into 10 bins from 0 to 1), $|B_i|$ is the number of samples in B_i , n is the total number of samples, $\text{acc}(B_i)$ is the accuracy of B_i , and $\text{conf}(B_i)$ is the average confidence of B_i . The final ECE is the weighted average across all intervals.

3.2 Confidence Calculating Details and Results

We employed two distinct methods for calculating confidence: Self-Confidence (SC) and Multiple-Prompting (MP) confidence.

- **SC setting:** We prompt the model to output both the result and its confidence. Model’s temperature is set to 0 to ensure the reproducibility of the setting.
- **MP setting:** We adopt a method similar to SimpleQA (Wei et al., 2024), but reduce the number of requests from 100 to 10 for efficiency, while keeping the temperature at 1. The final reply is determined by majority voting, and the confidence score is the count of the chosen response over 10. This balances computational efficiency with reliable confidence estimation.

¹<https://github.com/openai/tiktoken>

	Objective				Subjective				Overall			
	SC		MP		SC		MP		SC		MP	
	Acc	ECE	Acc	ECE	Acc	ECE	Acc	ECE	Acc	ECE	Acc	ECE
Claude-3.5-Sonnet	58.58	33.41	57.48	34.23	40.67	48.31	39.67	53.93	52.24	38.68	51.18	41.20
Gemini-Pro-1.5	58.94	35.39	60.58	22.24	38.00	56.14	36.67	51.80	51.53	42.73	52.12	32.58
Gemini-Flash-2.0	65.88	26.10	66.42	25.38	41.33	50.06	37.00	57.97	57.19	34.58	56.01	36.77
DeepSeek-V3	61.50	31.80	59.67	26.90	38.33	53.64	35.67	39.23	53.30	39.53	51.18	31.12
Llama-3.1-405B-Instruct	57.30	31.83	56.75	25.26	40.33	50.94	37.00	43.67	51.30	38.33	49.76	31.77
GPT-4o-mini	47.81	39.24	50.18	44.56	44.00	43.36	40.67	52.80	46.46	40.70	46.82	47.48
GPT4-turbo	59.12	30.95	61.50	34.40	41.33	48.50	37.33	59.07	52.83	37.16	52.83	43.13
GPT-4o	56.20	36.01	59.49	11.35	41.33	51.49	38.67	58.93	50.94	41.49	52.12	28.18
Qwen-Max	70.62	21.31	68.61	24.47	39.67	52.99	37.00	58.53	59.67	32.52	57.43	36.52
AutoJ 7B	-	-	12.98	67.26	-	-	31.46	47.76	-	-	23.52	56.11
JudgeLM 7B	-	-	19.12	27.92	-	-	28.96	56.61	-	-	24.32	35.20
JudgeLM 13B	-	-	27.53	47.57	-	-	32.92	25.72	-	-	30.35	23.48
JudgeLM 33B	-	-	32.89	46.84	-	-	39.35	17.92	-	-	36.31	19.46

Table 1: Model Performance of Prompted Judges and Fine-tuned Judges

We also observe that judge models may exhibit positional bias. To mitigate this, we systematically exchanged the order of inputs in the aforementioned settings. For further details, please refer to Appendix A.2.

It is worth noting that for fine-tuned judges (including AutoJ (Li et al., 2023) and JudgeLM (Zhu et al., 2023)), due to their limited instruction-following capabilities, we could only employ the MP setting. The experiment results can be found in Table 1. To have a better analysis of our experiments, we divide the model’s confidence into 10 intervals based on the confidence of the model. Then we plot the reliability diagram (Figure 1).

4 Analysis

4.1 Powerful Models Perform Better on Objective Category

As shown in Table 1, on the objective category, prompted judges significantly outperform fine-tuned judges in both Acc and ECE. Among the fine-tuned judges, even the best-performing model, JudgeLM 33B, performs worse than all prompted judges. This is likely because fine-tuned judges, as smaller models, lack the knowledge to answer objective questions. This suggests that a model’s intrinsic problem-solving capability is crucial for evaluating such tasks.

4.2 Small Fine-tuned Models are Competitive on Subjective Questions

On our subjective category, the best-performing fine-tuned judge, JudgeLM 33B, achieves comparable accuracy to most prompted judges and even demonstrates a clear advantage in ECE over the strongest prompted judges. This is likely be-

cause fine-tuned judges, trained on high-quality datasets, demonstrate improved calibration performance. This gives them an edge in assessing subjective tasks, where evaluation relies on the model’s internal criteria.

4.3 Almost All Models are Overly Confident

According to Figure 1, except for GPT-4, both fine-tuned judges and prompted judges exhibit significant overconfidence. This indicates that LLM-as-a-judge still faces significant challenges in calibration. The case of GPT-4 is unique, as it shows overconfidence in some scenarios and underconfidence in others.

4.4 Poor Performance on Subjective Category

We conducted a case study on the three strongest proprietary models. Specifically, we calculated the proportions of cases where the models consistently answered correctly, consistently answered incorrectly, or provided inconsistent responses across two attempts. The results are shown in Appendix A.3.

5 Conclusion

We introduced REAL-JUDGE, a comprehensive dataset designed to evaluate the reliability of LLM-as-a-judge. Our experiments revealed significant gaps in the calibration of even the strongest models. We found that prompted judges and fine-tuned judges each have their own advantages on subjective and objective evaluations. These findings provide practical insights for selecting appropriate LLM-as-a-judge models based on task requirements. Future work will focus on expanding the dataset and exploring methods to improve model calibration.

6 Limitation

This study has several limitations that highlight opportunities for future research. First, although it has been observed that fine-tuned models perform well on subjective questions compared to proprietary models, the study did not further attempt to train a fine-tuned model or propose a solution to better address it. Second, while the study covers a variety of topics, it could further refine the disciplines of objective questions and the categories of subjective questions to achieve a more granular evaluation and uncover more insights. Additionally, the dataset in this study is limited to Chinese, which may introduce biases for different models, a multilingual dataset would provide a more comprehensive analysis. Finally, the data construction process in this study remains heavily reliant on manual effort, resulting in poor scalability. Future work could explore the development of an automated pipeline to continuously update the data.

7 Ethical Statement

In conducting this research, we have adhered to the highest ethical standards and guidelines. All data used in this study were collected and processed in compliance with relevant data protection regulations and ethical guidelines. We ensured that no personally identifiable information (PII) was included in the dataset, and all data were anonymized to protect individual privacy.

For studies involving human participants, informed consent was obtained prior to their involvement, and participants were informed of their right to withdraw at any stage without penalty. No harmful or deceptive practices were employed, and the well-being of participants was prioritized throughout the research process.

Additionally, we have considered the potential societal impacts of our work, including the risks of misuse or unintended consequences. We are committed to promoting the responsible use of our findings and encourage further research to address any ethical concerns that may arise.

References

- Anthropic. 2024. Introducing claude 3.5 sonnet. <https://www.anthropic.com/news/claude-3-5-sonnet>.
- Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su,

Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. *Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 7421–7454. Association for Computational Linguistics.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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, Miaoqun 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, Wangding Zeng, Wanbiao Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu,

- Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. [Deepseek-v3 technical report](#). *Preprint*, arXiv: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.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. [Length-controlled alpaca-eval: A simple way to debias automatic evaluators](#). *Preprint*, arXiv:2404.04475.
- Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. [On calibration of modern neural networks](#). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. [Language models \(mostly\) know what they know](#). *Preprint*, arXiv:2207.05221.
- Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. [Generative judge for evaluating alignment](#). *Preprint*, arXiv:2310.05470.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qim-

512	ing Yuan, Wojciech Zaremba, Rowan Zellers, Chong
513	Zhang, Marvin Zhang, Shengjia Zhao, Tianhao
514	Zheng, Juntang Zhuang, William Zhuk, and Bar-
515	ret Zoph. 2024. Gpt-4 technical report . <i>Preprint</i> ,
516	arXiv:2303.08774.
517	Jiao Ou, Junda Lu, Che Liu, Yihong Tang, Fuzheng
518	Zhang, Di Zhang, and Kun Gai. 2023. Dialog-
519	bench: Evaluating llms as human-like dialogue sys-
520	tems. <i>arXiv preprint arXiv:2311.01677</i> .
521	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,
522	Carroll Wainwright, Pamela Mishkin, Chong Zhang,
523	Sandhini Agarwal, Katarina Slama, Alex Ray, et al.
524	2022. Training language models to follow instruc-
525	tions with human feedback. <i>Advances in neural in-</i>
526	<i>formation processing systems</i> , 35:27730–27744.
527	Arjun Panickssery, Samuel R. Bowman, and Shi Feng.
528	2024. Llm evaluators recognize and favor their own
529	generations . <i>Preprint</i> , arXiv:2404.13076.
530	Sijun Tan, Siyuan Zhuang, Kyle Montgomery,
531	William Y Tang, Alejandro Cuadron, Chenguang
532	Wang, Raluca Ada Popa, and Ion Stoica. 2024.
533	Judgebench: A benchmark for evaluating llm-based
534	judges. <i>arXiv preprint arXiv:2410.12784</i> .
535	Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-
536	Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan
537	Schalkwyk, Andrew M Dai, Anja Hauth, Katie
538	Millican, et al. 2023. Gemini: a family of
539	highly capable multimodal models. <i>arXiv preprint</i>
540	<i>arXiv:2312.11805</i> .
541	Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier
542	Martinet, Marie-Anne Lachaux, Timothée Lacroix,
543	Baptiste Rozière, Naman Goyal, Eric Hambro,
544	Faisal Azhar, et al. 2023. Llama: Open and effi-
545	cient foundation language models. <i>arXiv preprint</i>
546	<i>arXiv:2302.13971</i> .
547	Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu,
548	Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and
549	Zhifang Sui. 2023. Large language models are not
550	fair evaluators. <i>arXiv preprint arXiv:2305.17926</i> .
551	Jason Wei, Nguyen Karina, Hyung Won Chung,
552	Yunxin Joy Jiao, Spencer Papay, Amelia Glaese,
553	John Schulman, and William Fedus. 2024. Mea-
554	suring short-form factuality in large language models .
555	<i>Preprint</i> , arXiv:2411.04368.
556	An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui,
557	Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu,
558	Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 tech-
559	nical report. <i>arXiv preprint arXiv:2412.15115</i> .
560	Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan
561	Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin,
562	Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023.
563	Judging llm-as-a-judge with mt-bench and chatbot
564	arena. <i>Advances in Neural Information Processing</i>
565	<i>Systems</i> , 36:46595–46623.
566	Lianghui Zhu, Xinggang Wang, and Xinlong Wang.
567	2023. Judgelm: Fine-tuned large language models
568	are scalable judges . <i>Preprint</i> , arXiv:2310.17631.

A	Appendix	569
A.1	Model Families Used in Data Construction	570
		571
	Here, we list some of the models used in our data construction process.	572
		573
	• Baichuan	574
	– Baichuan4	575
	• DeepSeek	576
	– DeepSeek-V2.5	577
	– DeepSeek-Chat	578
	• Llama	579
	– Llama3_1-70B-6M-math	580
	– Meta-Llama-3.1-405B-Instruct	581
	– Meta-Llama-3.1-70B-Instruct	582
	– Meta-Llama-3.1-8B-Instruct	583
	– Meta-Llama-3.2-1B-Instruct	584
	– Meta-Llama-3.2-3B-Instruct	585
	– Meta-Llama-3.3-70B-Instruct	586
	• Mistral	587
	– Mistral-Large-Instruct	588
	– Mistral-Small-Instruct	589
	• Phi	590
	– Phi-3.5-mini-instruct	591
	• Qwen	592
	– Qwen2.5-0.5B-Instruct	593
	– Qwen2.5-72B-Instruct	594
	– Qwen2.5-7B-Instruct	595
	– Qwen2.5-Coder-32B-Instruct	596
	– Qwen-Max	597
	• ChatGPT	598
	– ChatGPT-4o-latest	599
	– GPT-4o-2024-08-06	600
	– GPT-4o-2024-11-20	601
	– GPT-4o-mini-2024-07-18	602
	• Claude	603
	– Claude-3-5-Sonnet	604
	• Gemini	605
	– Gemini-1.5-pro-latest	606

- **Gemma**
 - Gemma-2-27b-it
 - Gemma-2-2b-it
 - Gemma-2-9b-it
 - Gemma-2-9b-0729-v32
- **O1**
 - O1-mini
 - O1-preview
 - O1-pro
- **Yi**
 - Yi-large
 - Yi-lightning

Models	Both correct	Conflict	Both wrong
GPT-4o	62.77	18.61	18.61
Claude-Sonnet3.5	56.20	25.55	18.25
Gemini-Flash	66.97	20.44	12.59

Table 2: Case study of our subjective dataset

Models	Both correct	Conflict	Both wrong
GPT-4o	48.73	27.66	23.61
Claude-3.5-Sonnet	47.79	32.86	19.34
Gemini-Flash-2.0	45.93	29.67	24.40

Table 3: Case study of our objective dataset

A.2 Positional Bias Mitigation

LLM-based judges are known to exhibit positional bias, where the order of response pairs may influence model’s decision. We take this bias into consideration. Since our data always have one response that is better than the other, we can ask the model twice. For example, if the answer is $A > B$, we only consider it correct if the model outputs $A > B$ when asked in the original order, and $A < B$ when asked in the reverse order. We do the similar in the multiple-prompting confidence method, we ask five times in the original order and five times in the reverse order, take their respective majorities as each order’s answer and deal with answers similarly to the self-confidence case. It is worth mentioning that we need to choose a number of times that, when divided by 2, results in an odd number, for we need to get a majority of answers whether we ask in forward or reverse order. Given the cost limitations and the stability of our experiment, we choose 10 as a balanced number of trials.

This helps eliminate the positional bias. It is worth noting that this bias is related to the lower accuracy in the experimental results.

A.3 Case Study of Subjective Category

As we mentioned before in Section 4.4, we conduct a case study on the three most influential models. The results are shown in the Table 3 and Table 2. As can be seen, the proportion of inconsistent responses is significantly higher for subjective questions compared to objective questions, with the Claude Sonnet model even reaching as high as 32.86%. This is likely due to the lack of a stable and robust evaluation criterion in the models.

A.4 Task Guidelines for Objective Category Data Construction

Appendix: Task Guidelines for Question Rewriting and Analysis

Objective

In this task, your goal is to rewrite provided questions from high school entrance exams, college entrance exams, and national exams, while also providing detailed analysis and explanations. The questions primarily cover mathematics, chemistry, biology, and basic logical reasoning. For each provided question, you will see the question text and its original answer in the annotation interface. Note that all questions are single-choice; there are no multiple-choice or fill-in-the-blank questions.

Rewriting Methods

You need to rewrite the original questions in one of the following two ways. Please note that the rewriting methods are prioritized from high to low. You should prioritize higher-priority methods:

1. Rewriting the Options

First, check if the original question can be rewritten by modifying the options. Rewriting the options involves: - Changing the original correct option to an incorrect option. - Selecting one of the incorrect options and rewriting it as the new correct option. - Only these two options should be modified; the other two options should remain unchanged. - Directly delete the original correct option and write a completely new correct option. Do not simply modify numerical values, swap options, or partially edit options. The new correct option should be entirely different from the original. For numerical questions, "different" means a change in values. For knowledge-based or factual questions, "different" means the new correct option should state a completely different objective fact. - Do not use large language models to assist in rewriting. - Indicate your chosen rewriting method in the annotation interface.

If, after careful consideration, you determine that the question cannot be rewritten by modifying the options, proceed to the second rewriting method.

2. Rewriting the Question

If the question cannot be rewritten by modifying the options, rewrite the question itself. Rewriting the question involves: - Making minimal changes to the question to alter the correct option. - Ensuring that the question remains a single-choice question after modification. - Avoid modifying both the question and the options simultaneously. - There are two main ways to rewrite the question: 1. Modify numerical values or chemical equations (common in mathematics and chemistry questions). 2. Modify factual information (common in biology and chemistry questions). - Prioritize modifying numerical values for easier analysis updates. - If neither of these methods works, you may use other approaches to rewrite the question. - Indicate your chosen rewriting method in the annotation interface.

Writing the Analysis

After rewriting the question or options, you must provide a detailed analysis of the question. Questions can be categorized into three types: 1. **Numerical Calculation Questions:** - Directly write out the calculation process in the analysis. - Provide the correct option.

2. **Knowledge-Based Questions:** - Analyze each option in detail, explaining why it is correct or incorrect. - Do not only analyze the correct option.

3. **Logical Reasoning Questions:** - Provide a complete logical reasoning process. - Provide the correct option.

A.5 Prompts

For better reproducibility, here we provide the prompts we use in our experiments.

Self-Confidence(SC) Prompt

Prompt: 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 and provide a confidence score for your selection.

Here is the question:

{question}

Please evaluate the outputs and provide your best guess along with a confidence score between 0% to 100% in the following JSON format:

```
{  
  "selected_output": "Output (a)" or "Output (b)",  
  "confidence_score": number  
}
```

Instruction:

{question}

Output (a):

<| The Start of Assistant A's Answer |>

Output (b):

<| The Start of Assistant B's Answer |>

Your response should be in the JSON format as shown above. Do not output ANYTHING else. Do not provide the % symbol in the confidence score.

Multiple-Prompting(MP) Prompt

Prompt: 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.

Here is the question:

{question}

Please evaluate the outputs and provide your best guess in the following JSON format:

```
{  
  "selected_output": "Output (a)" or "Output (b)"  
}
```

Instruction:

{question}

Output (a):

<| The Start of Assistant A's Answer |>

Output (b):

<| The Start of Assistant B's Answer |>

Your response should be in the JSON format as shown above. Do not output ANYTHING else.

JudgeLM Prompt

Prompt: You are a helpful and precise assistant for checking the quality of the answer.

[Question]

{ question }

[The Start of Assistant 1's Answer]

{ answer_a }

[The End of Assistant 1's Answer]

[The Start of Assistant 2's Answer]

{ answer_b }

[The End of Assistant 2's Answer]

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

Response:

AutoJ Prompt

User Prompt: You are assessing two submitted responses on a given user's query and judging which response is better or they are tied. Here is the data:

[BEGIN DATA]

[Query]: { question }

[Response 1]: { response_a }

[Response 2]: { response_b }

[END DATA]

Here are the instructions to assess and compare the two responses:

1. Pinpoint the key factors to distinguish these two responses.
2. Conclude your comparison by providing a final decision on which response is better, or they are tied. Begin your final decision statement with "So, the final decision is Response 1 / Response 2 / Tie". Ensure that your decision aligns coherently with the comprehensive evaluation and comparison you've provided.

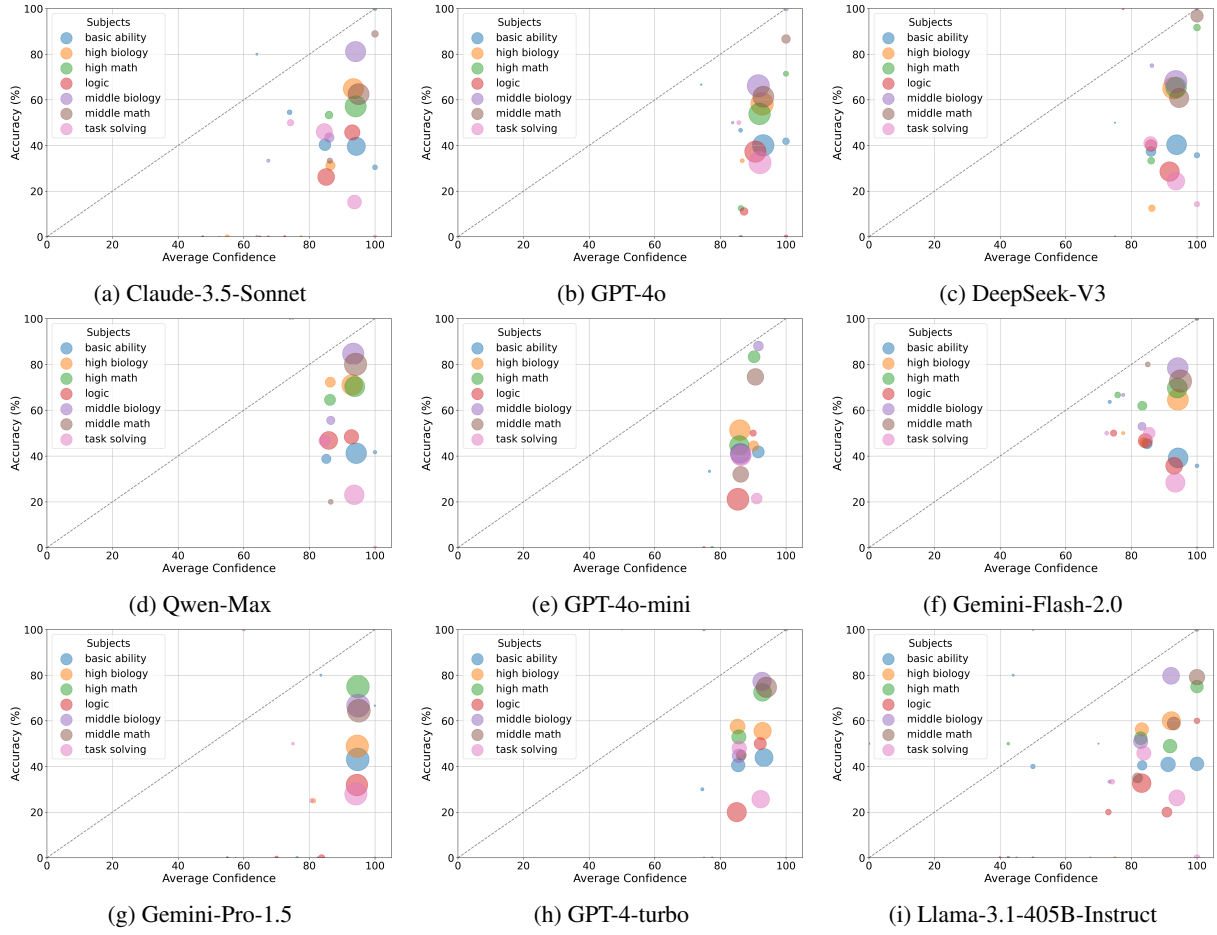


Figure 3: Reliability Diagram (Self-Confidence Setting)

A.6 Detailed Results

Here we provide experimental results of different subjects for more detailed analysis.

A.6.1 Self-Confidence Setting

A.6.2 Multiple-Prompting Setting

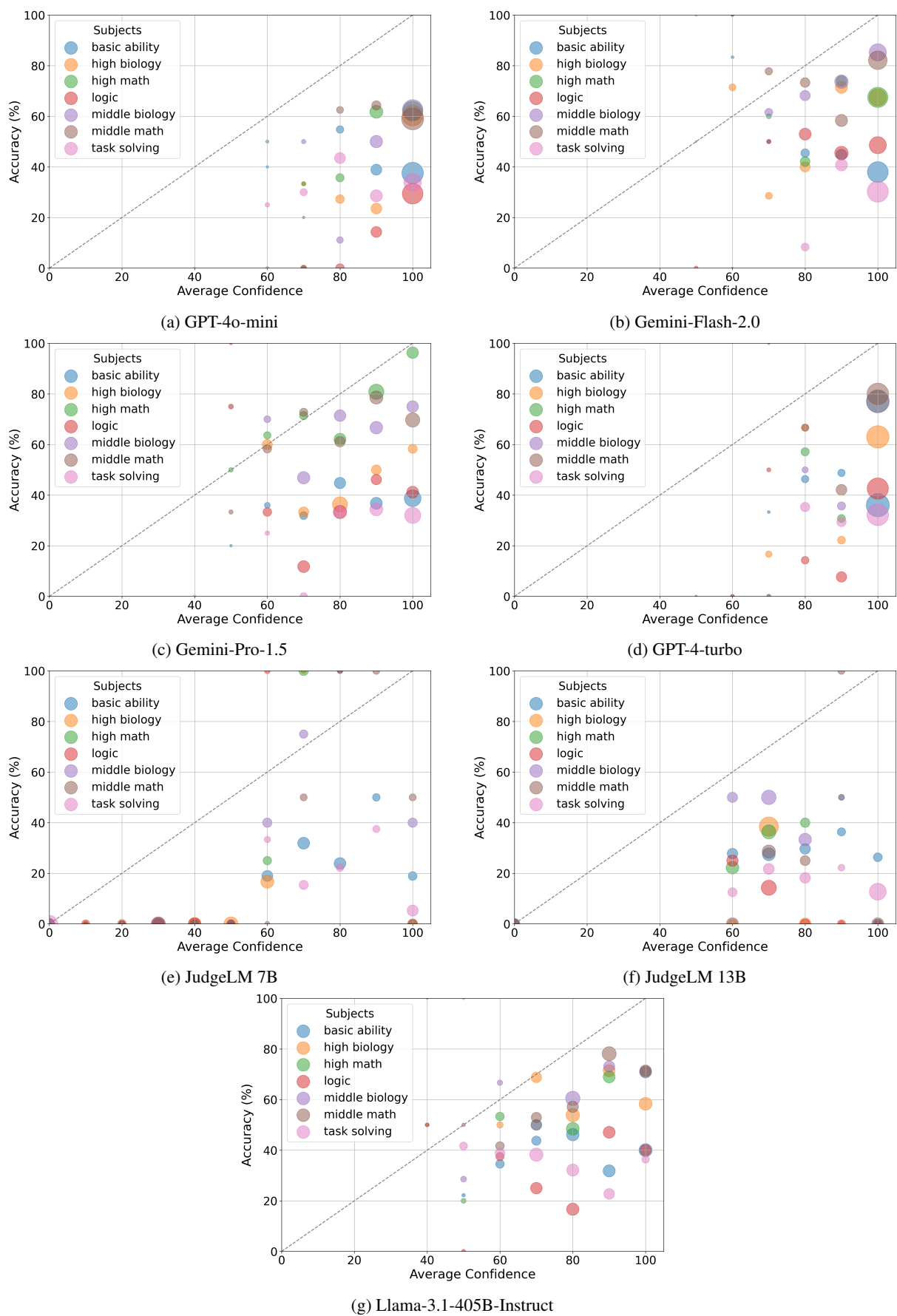


Figure 4: Reliability Diagram (Multiple-Prompting Setting)