LLMJudge: Meta-Evaluating Automatic Relevance Judgments

Hossein A. Rahmani
University College London
London, UK
hossein.rahmani.22@ucl.ac.uk

Clemencia Siro
University of Amsterdam
Amsterdam, The Netherlands
c.n.siro@uva.nl

Mohammad Aliannejadi
University of Amsterdam
Amsterdam, The Netherlands
m.aliannejadi@uva.nl

Nick Craswell
Microsoft
Bellevue, US
nickcr@microsoft.com

Charles L. A. Clarke
University of Waterloo
Waterloo, Ontario, Canada
claclark@gmail.com

Guglielmo Faggioli
University of Padua
Padua, Italy
faggioli@dei.unipd.it

Bhaskar Mitra
Independent Researcher
Montréal, Canada
bhaskar.mitra@acm.org

Paul Thomas
Microsoft
Adelaide, Australia
pathom@microsoft.com

Emine Yilmaz
University College London
London, UK
emine.yilmaz@ucl.ac.uk

Abstract

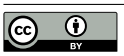
Large language models (LLMs) are increasingly used to produce the relevance labels that underpin the evaluation of information retrieval (IR) and retrieval-augmented generation systems. Yet delegating relevance assessment to an LLM raises a meta-evaluation question that is hard to answer with existing resources: *when, and how much, can we trust an automatic judge?* We present **LLMJudge**, a resource for *meta-evaluating* LLM-generated relevance judgments. LLMJudge collects **42** complete judgment sets over the TREC 2023 Deep Learning track, spanning fine-tuned, proprietary, and open-source models and a wide range of prompting strategies. Each judgment set is paired with official human `qrels`, enabling two complementary axes of analysis: *label agreement* with human assessors (e.g., Cohen's κ, Krippendorff's α) and *system-ranking fidelity* (e.g., Kendall's τ, Spearman's ρ). Beyond the data, we release a standardized format, baselines, and an evaluation harness so that new judges can be plugged in and compared reproducibly. Our analysis shows that high label agreement is neither necessary nor sufficient for faithful system ranking, that open-source judges are markedly more stable than proprietary ones, and that fine-tuning improves correlation at the cost of reliability. LLMJudge is publicly available and provides a reusable testbed for auditing the judges who increasingly judge our systems. Full resources, data, and detailed results are available at https://llm4eval.github.io/LLMJudge-benchmark/.

CCS Concepts

• **Information systems** → **Information retrieval**.

Keywords

LLM-as-a-Rel, LLM4Eval, Relevance Judgment, Benchmark



CIKM '26, Rome, Italy

ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840214

ACM Reference Format:
Hossein A. Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2026. LLMJudge: Meta-Evaluating Automatic Relevance Judgments. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 07–11, 2026, Rome, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3799682.3840214

1 Introduction

Since the Cranfield paradigm, the validity of an information retrieval (IR) experiment has rested on one fragile ingredient: *relevance judgments* produced by human assessors [3, 14]. Human judgment is expensive and slow, and the cost grows with every new query, corpus, and task. Large language models (LLMs) promise relief: prompted or fine-tuned, they can label query–document pairs at a fraction of the cost and have been reported to correlate strongly with human assessors [5, 11, 13]. As a result, LLM-generated labels are quietly becoming the default for evaluating retrieval and retrieval-augmented generation (RAG) systems.

This shift raises new questions, moving from "is this document relevant? to a *meta-evaluation* question: "*can we trust this judge?*" The concern is not hypothetical. Automatic judges are prone to well-documented failure modes. Positional and verbosity bias [15], *self-preference* where a model favors text in its own style [6], *circularity* when the same model family generates and evaluates content, benchmark memorization [1], and vulnerability to adversarial content [10]. Diagnosing these failures requires more than a single judge on a single benchmark; it requires *many* judges, evaluated the *same* way, against a common human ground truth.

No existing resource supports this. Public test collections release human `qrels` but not the alternative LLM judgments needed to study judge behavior; individual LLM-as-a-judge papers report one or two systems under bespoke protocols that cannot be compared across papers. What is missing is a *shared, comparable collection of judges* together with the data and tooling to meta-evaluate them.

We address this gap with **LLMJudge**, a resource for meta-evaluating LLM-generated relevance judgments built on the TREC 2023 Deep Learning track [4]. LLMJudge gathers **42** complete judgment sets

Table 1: Positioning of LLMJudge against representative resources. ✓ full, ○ partial, — absent.

Resource	Many judges	Label agr.	Rank fidel.	Human aligned	Harness
TREC DL qrels [4]	—	—	—	✓	—
Thomas et al. [11]	—	✓	○	✓	—
UMBRELA [12]	—	✓	✓	✓	○
MT-Bench / judge [17]	○	✓	—	○	○
LLMJudge (ours)	✓	✓	✓	✓	✓

from **8** independent teams (comprising organizer baselines and participant runs), each scoring the same query–passage pairs on a four-point graded scale and each aligned to the official human `qrels`. This design lets us ask, for every judge, two questions at once: how well does it *agree* with humans at the label level, and how faithfully does it *rank* the systems that humans would rank? Figure 1 summarizes the resource.

Contributions. This paper makes the following contributions:

(i) **A meta-evaluation resource.** We release 42 LLM-generated judgment sets from 8 teams over TREC 2023 DL, the largest public collection of comparable relevance judges, paired with human `qrels` (Section 3).
(ii) **A standardized protocol and harness.** We define a common judgment format and an evaluation toolkit that scores any new judge on both label-agreement and system-ranking axes, enabling reproducible, plug-in comparison (Section 3).
(iii) **A benchmark analysis.** Across the 42 judges, we characterize how model family, fine-tuning, and prompting affect agreement and ranking, and show that the two axes can diverge sharply (Section 4).
(iv) **A reusable testbed for judge auditing.** The resource supports ongoing study of judge vulnerabilities (bias, circularity, self-preference, and adversarial robustness), and lowers the barrier for the community to submit and audit new judges (Section 5).

2 Related Work and Positioning

2.1 LLMs as relevance assessors

Faggioli et al. [5] framed the spectrum of human–machine collaboration for relevance assessment, while Thomas et al. [11] showed that a carefully prompted LLM can approximate human assessors on enterprise search. UMBRELA [12] operationalized this as an open, reproducible assessor, and the LLMJudge and related challenges [8, 9] began collecting judges at scale. These efforts establish that LLMs *can* judge; our resource targets the orthogonal question of *meta-evaluating* when they should be trusted.

2.2 Known vulnerabilities

LLM judges exhibit systematic biases (position, verbosity, and self-enhancement [15, 17]), and *self-preference*, favoring outputs that resemble their own generations [6]. Reusing one model family to both generate and evaluate introduces *circularity* [5], and contamination or benchmark memorization can inflate apparent quality [1].

Table 2: LLMJudge data statistics. Each query–passage pair carries a four-point human label used as ground truth.

Split	Queries	Passages	Human qrels
Development	25	7,224	7,263
Test	25	4,414	4,423
Total	50	11,638	11,686

Judges can also be manipulated by adversarial or optimized content [10]. Auditing these effects requires a diverse pool of judges scored under one protocol, which is precisely what LLMJudge provides.

2.3 Positioning

Table 1 positions LLMJudge against representative resources along the axes that matter for judge meta-evaluation: whether the resource releases multiple comparable *judges*, both *label-agreement* and *system-ranking* signals, alignment to *human* ground truth, a graded relevance scale, and a reusable *harness*. Unlike single-judge studies and human-only collections, LLMJudge is built specifically as a many-judge, two-axis, plug-in testbed.

3 The LLMJudge Resource

3.1 Task and Relevance Scale

LLMJudge instantiates *pointwise* relevance assessment: given a query q and a passage p, a judge assigns a graded label on the four-point TREC Deep Learning scale (i.e., **0** Irrelevant, **1** Related, **2** Highly relevant, **3** Perfectly relevant [4]). The graded scale (rather than binary) lets us study agreement at multiple relevance thresholds and matches the labels used to compute official TREC metrics.

3.2 Underlying Collection and Data Statistics

The resource is built on the TREC 2023 Deep Learning passage track. We adopt the official query pool (a mix of organic, T5-generated, and GPT-4-generated topics) and split the judged pairs into a *development* and a *test* portion so that prompt and model selection can be separated from final reporting. Table 2 summarizes the splits. Every pair in both splits carries an official human label, providing the ground truth against which all 42 judges are meta-evaluated.

3.3 The Judges

The 42 judgment sets come from 8 teams and span the contemporary design space: *model family* (proprietary vs. open-source), *adaptation* (zero-/few-shot prompting vs. fine-tuning), *prompt strategy* (direct, chain-of-thought, nugget- and rubric-based decomposition), and *label encoding* (numerical vs. semantic). This diversity is the core value of the resource: it turns judge design choices into controlled, comparable variables. Each judgment set is released in a single normalized format (`query_id`, `passage_id`, `predicted_label`) so that any judge (existing or new) can be scored identically.

3.4 Evaluation Protocol and Harness

We meta-evaluate every judge along two axes against the human `qrels`:

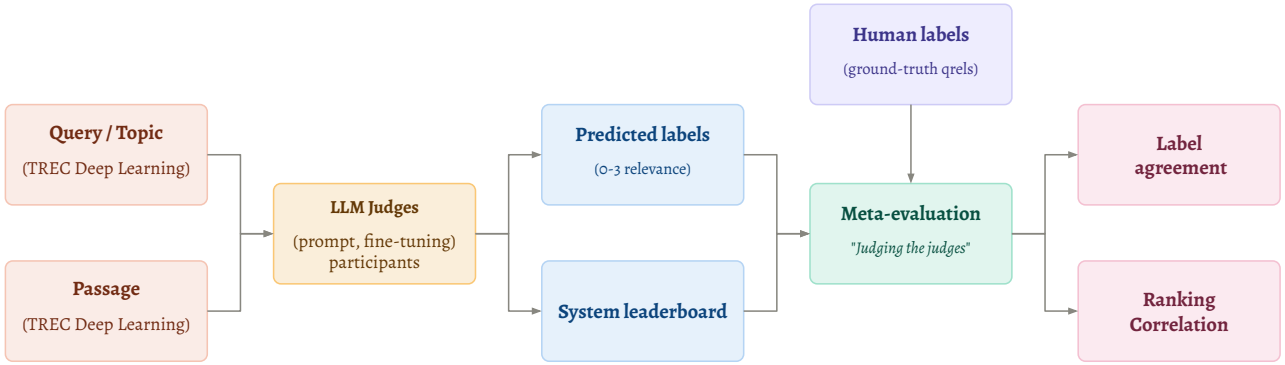


Figure 1: Overview of LLMJudge. Eight teams contribute 42 LLM judges that label TREC 2023 DL query–passage pairs on a four-point scale. Predicted labels yield both an induced system leaderboard and direct label comparisons; meta-evaluation against human `qrels` measures label agreement (κ, α) and ranking fidelity (τ, ρ).

Table 3: Representative results over the test split. κ: Cohen's κ; α: Krippendorff's α; τ: Kendall's τ; ρ: Spearman's ρ. Best per column in bold.

Judge	κ	α	τ	ρ
willia-umbrela1	**0.286**	0.492	0.901	0.981
h2oloo-fewself	0.277	**0.496**	0.909	0.982
h2oloo-zeroshot1	0.282	0.481	0.918	0.983
RMITIR-llama70B	0.265	0.487	0.935	0.988
prophet-setting2	0.176	0.314	**0.952**	**0.991**
NISTRetrieval-instruct2	0.188	0.382	0.944	0.991
TREMA-rubric0	0.078	0.104	0.828	0.954
TREMA-nuggets	0.060	0.169	0.866	0.972

- **Label agreement**: *Label correlation* measures how often the judge reproduces human labels. We use the automated evaluation metrics Cohen's κ and Krippendorff's α (the latter robust to the ordinal, multi-rater setting) on human judgments and the judgments submitted by participants;
- **System-ranking fidelity**: *System-ranking fidelity* measures whether the judge would reorder the systems that humans rank: we recompute the official leaderboard using each judge's labels and report Kendall's τ and Spearman's ρ against the human-induced ranking. The released harness takes a judgment file and emits all four metrics, plus threshold-binarized κ, with a single command, making submission of a new judge a one-step operation.

We use `scikit-learn`[1] to compute Cohen's κ, Kendall's τ, Spearman's ρ. Krippendorff's α is also calculated using the Fast Krippendorff[2] Python package.

4 Benchmark Analysis

We illustrate the resource with a reference analysis over all 42 judges. Table 3 reports the strongest judges on each axis; Figures 2 and 3 plot the full pool.

[1]https://scikit-learn.org/stable/index.html
[2]https://github.com/pln-fing-udelar/fast-krippendorff

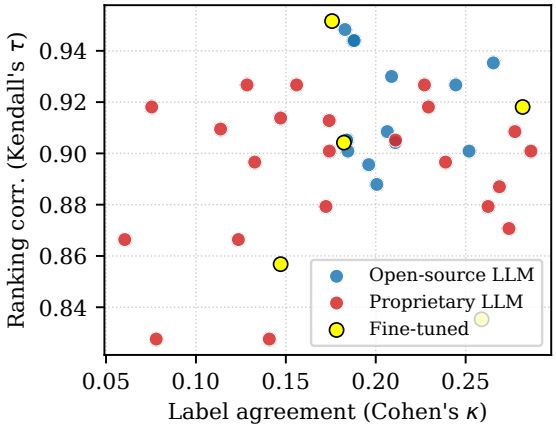


Figure 2: Label agreement (Cohen's κ) vs. system-ranking fidelity (Kendall's τ) for all 42 judges. High agreement is neither necessary nor sufficient for faithful ranking.

The two axes diverge. The judge with the best human *label agreement* (`willia-umbrela1`, κ=0.286) is *not* the one that best reproduces the human *system ranking* (`prophet-setting2`, τ=0.952), which itself has only middling agreement (κ=0.176). Figure 2 makes this concrete: τ stays high (> 0.9) across a wide band of κ, so a judge can rank systems faithfully while disagreeing with humans on many individual labels, and vice versa. Practitioners should therefore choose a judge by the axis that matches their use case rather than by agreement alone.

Open-source judges are more stable. Open-source judges form a tight, high-τ cluster in Figure 3, whereas judges based on proprietary models spread across the full range, including the weakest runs (`TREMA-rubric0`, `TREMA-nuggets`). The variability is driven largely by *prompt strategy*: heavy decomposition (nugget/rubric pipelines) degrades both reliability and ranking, while direct and lightly structured prompts are more robust.

Fine-tuning trades reliability for ranking. Fine-tuned judges (diamonds) reach the highest τ (`prophet-setting2`, `h2oloo-zeroshot1`) but do not lead on α, indicating that adaptation sharpens system-level discrimination without necessarily improving per-label reliability. Finally, numerical and semantic label encodings perform comparably, suggesting the encoding is a free design choice. These observations are reproducible end-to-end with the released harness and serve as baselines for future submissions.

5 Use cases

LLMJudge is designed to be reused. (i) *Judge selection*: practitioners can read off, from the two-axis profile, which released judge best fits their goal, that is, high label agreement for fine-grained annotation, or high ranking fidelity for leaderboard evaluation. (ii) *Benchmarking new judges*: a new model is scored against the same human `qrels` with one command, making results directly comparable to the 42 baselines. (iii) *Auditing vulnerabilities*: because the resource contains many same-family and cross-family judges, it supports controlled study of circularity, self-preference, and adversarial robustness. For example, recent work leverages such pools to systematically analyze "overrating behavior" and label inflation across different model families and evaluation paradigms [16]. (iv) *Judge ensembling*: the diverse pool is a natural substrate for blending judges, as explored by JudgeBlender [7], which combines multiple LLM assessments to improve robustness. Furthermore, the dataset enables advanced ensembling techniques that model inter-judge correlations and spurious confounders (such as generation length) via higher-rank latent variable inference. (v) *Resolving rating indeterminacy*: standard validation approaches that rely on forced-choice ratings often falter when multiple interpretations of a query are plausible. A diverse collection of judge outputs allows researchers to validate multi-label "response set" elicitation, establishing human-judge agreement even under subjective ambiguity. (vi) *Large-scale multi-task supervision*: synthetic labels generated by LLMs provide critical offline supervision for industrial systems. Datasets like ours guide the alignment of lightweight proxy judges that produce relevance signals, allowing ranking architectures to explicitly balance semantic relevance against user engagement metrics such as CTR and conversion rate [2].

6 Limitations

LLMJudge currently targets passage-level, pointwise relevance on a single host collection (TREC 2023 DL) and a single language. The two-axis protocol captures agreement and ranking but not downstream effects on, e.g., RAG answer quality. These are natural directions for community extension, which the standardized format is built to accommodate.

7 Conclusion

As LLMs increasingly produce the labels that decide which IR and RAG systems "win", the community needs a principled way to judge the judges. LLMJudge supplies the missing ingredient: a large, comparable collection of 42 LLM judges, aligned to human ground truth, with a standardized two-axis protocol and harness. Our analysis shows that label agreement and ranking fidelity can diverge, that open-source judges are more stable than proprietary ones, and that fine-tuning trades reliability for ranking, findings that are actionable precisely because the resource makes them reproducible. We release LLMJudge to support the ongoing auditing of automatic relevance assessment.

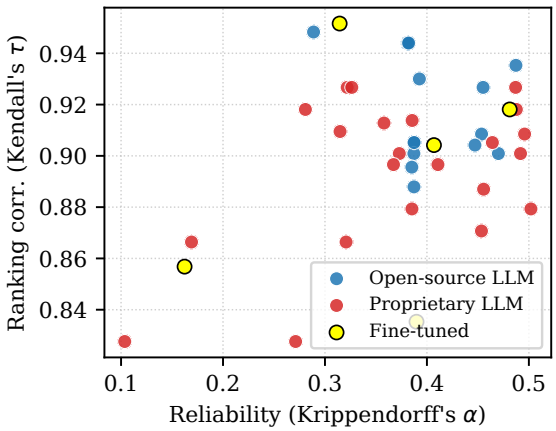


Figure 3: Reliability (Krippendorff's α) vs. ranking fidelity (Kendall's τ). Open-source judges cluster tightly; proprietary judges span a wider range.

Acknowledgments

The challenge is organized as a joint effort by University College London, Microsoft, the University of Amsterdam, the University of Waterloo, and the University of Padua. The views expressed in the content are solely those of the authors and do not necessarily reflect the views or endorsements of their employers and/or sponsors. This research is supported by the Engineering and Physical Sciences Research Council [EP/S021566/1], CAMEO, PRIN 2022 n. 2022ZLL7MW and by the Dreams Lab, a collaboration between Huawei Finland, the University of Amsterdam, and the Vrije Universiteit Amsterdam.

GenAI Usage Disclosure

Generative AI (GenAI) tools were used during multiple stages of this research project. Specifically, GenAI tools were used to assist with paraphrasing and language refinement, as well as debugging during implementation. All AI-assisted outputs were reviewed and validated by the authors, who take full responsibility for the content of this paper.

References

[1] Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, Cheat, Repeat: Data Contamination and Evaluation Malpractices in Closed-Source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

[2] Luming Chen, Jiaqi Xi, Raghav Saboo, Kenny Chi, Martin Wang, Sudeep Das, Danny Nightingale, Aditya Dodda, Elyse Winer, and Akshad Viswanathan. 2026. Joint Optimization of Relevance and Engagement in Multi-Task Ranking for E-Commerce with Efficient LLM Supervision. *arXiv preprint arXiv:2605.27704* (2026).

[3] Cyril Cleverdon. 1967. The Cranfield tests on index language devices. *Aslib Proceedings* 19, 6 (1967), 173–194.

[4] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Hossein A. Rahmani, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2024. Overview of the

TREC 2023 Deep Learning Track. In *Proceedings of the Thirty-Second Text REtrieval Conference (TREC)*.

[5] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR)*. 39–50.

[6] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[7] Hossein A. Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2024. JudgeBlender: Ensembling Judgments for Automatic Relevance Assessment. In *arXiv preprint arXiv:2412.13268*.

[8] Hossein A. Rahmani, Clemencia Siro, Mohammad Aliannejadi, Nick Craswell, Charles L. A. Clarke, Guglielmo Faggioli, Bhaskar Mitra, Paul Thomas, and Emine Yilmaz. 2024. LLMJudge: LLMs for Relevance Judgments. In *Proceedings of the LLM4Eval Workshop at SIGIR*.

[9] Hossein A. Rahmani, Xi Wang, Emine Yilmaz, Mohammad Aliannejadi, Simon Lupart, and Charles L. A. Clarke. 2024. Report on the 1st Workshop on Large Language Model for Evaluation in Information Retrieval (LLM4Eval 2024) at SIGIR 2024. *ACM SIGIR Forum* 58, 2 (2024).

[10] Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2024. Optimization-based Prompt Injection Attack to LLM-as-a-Judge. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

[11] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large Language Models can Accurately Predict Searcher Preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1930–1940.

[12] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Daniel Campos, Nick Craswell, Ian Soboroff, Laura Dietz, and Jimmy Lin. 2024. UMBRELA: UMbrela is the (Open-Source Reproduction of the) Bing Relevance Assessor. In *arXiv preprint arXiv:2406.06519*.

[13] Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. A Large-Scale Study of Relevance Assessments with Large Language Models: An Initial Look. In *arXiv preprint arXiv:2411.08275*.

[14] Ellen M. Voorhees. 2001. The Philosophy of Information Retrieval Evaluation. *Workshop of the Cross-Language Evaluation Forum (CLEF)* (2001), 355–370.

[15] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large Language Models are not Fair Evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

[16] Chuting Yu, Hang Li, Guido Zuccon, Joel Mackenzie, and Teerapong Leelanupab. 2026. When LLM Judges Inflate Scores: Exploring Overrating in Relevance Assessment. *arXiv preprint arXiv:2602.17170* (2026).

[17] 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. In *Advances in Neural Information Processing Systems (NeurIPS)*.