

Calibrate the Instrument First: Chance-Corrected Agreement as a Precondition for LLM-as-Judge Agent Evaluation

Sai Teja Alasyam
salasya2@asu.edu

July 28, 2026

Abstract

Automated evaluation of tool-using language-model agents increasingly relies on a second language model acting as judge. The validity of every downstream claim therefore rests on the judge, yet judges are most often reported—when they are reported at all—using raw agreement with human labels. We argue that raw agreement is not merely a weak diagnostic but an actively misleading one under the class imbalance typical of agent benchmarks, where most cases pass. We make this precise: any judge that emits a constant verdict attains raw agreement equal to the majority-class prevalence while achieving a Cohen’s κ of exactly zero, independent of the data. We demonstrate the gap empirically on 90 hand-labeled agent transcripts. A programmatic assertion judge, which checks whether required tools were called, reaches 81.1% raw agreement with $\kappa = 0.000$; a rubric-based LLM judge reaches 96.7% with $\kappa = 0.894$. A reader given only the agreement figures would conclude the two instruments differ modestly; the chance-corrected figures show that one carries no information at all. We then use the calibrated judge—and only the calibrated judge—to compare two agent models on the same suite, and report a non-significant difference (15/18 versus 14/18, McNemar exact $p = 1.0$) together with the diagnostic that explains it: the suite is saturated, with 14 of 18 cases passing unanimously for both models. We package the resulting *calibrate-then-measure* protocol as an open-source harness in which judge prompts are content-addressed, so that a stored evaluation result can never be compared against one produced by a different judge. We conclude that reporting κ before reporting any judge-derived metric should be a minimum standard for agent evaluation, and that a null result with a stated cause is more useful than a significant result with an unvalidated instrument.

1 Introduction

Language-model agents that call external tools are now evaluated at a scale that makes human grading impractical. A suite of a few dozen tasks, sampled several times to account for stochasticity, and run against a handful of candidate models, produces thousands of transcripts per experiment. The standard response is to delegate grading to another language model, an arrangement usually called LLM-as-a-judge [Zheng et al., 2023]. This is a reasonable engineering decision. It is also a measurement decision, and it is rarely treated as one.

When a judge assigns the labels, every subsequent number—pass rate, confidence interval, regression verdict, model ranking—is a function of the judge’s behavior. If the judge is biased, the bias propagates silently into all of them. The natural safeguard is to validate the judge against human labels before trusting anything it produces. In practice this validation, when it appears, is reported as raw agreement: the fraction of items on which judge and human assigned the same label.

Raw agreement is the wrong statistic for this job, and the reason is structural rather than incidental. Agent benchmarks are typically imbalanced: a suite is usually built so that a competent agent passes most of it, which means the pass class dominates. Under imbalance, a judge can achieve high agreement by exploiting the base rate rather than by discriminating between passes and failures. The pathological case is a judge that emits the same verdict for every input. Such a judge conveys no information whatsoever, yet its raw agreement equals the prevalence of the majority class, which on a well-constructed suite is high by construction.

This is not a hypothetical failure mode. It is the exact behavior of the programmatic baseline in our experiments. An assertion-based judge that verifies whether the required tools were invoked passes all 90 transcripts in our labeled set, including all 17 that a human marked as failures. Its raw agreement is 81.1%—a figure that, presented in isolation, looks like a serviceable instrument. Its Cohen’s κ [Cohen, 1960] is 0.000: exactly chance.

The corrective is well established outside machine learning. Chance-corrected agreement coefficients were developed precisely because percentage agreement is inflated by base rates, and κ has been standard in content analysis, epidemiology, and computational linguistics for decades [Cohen, 1960, Landis and Koch, 1977, Artstein and Poesio, 2008]. Its limitations, including the well-documented paradox in which high agreement coexists with low κ under skewed marginals [Feinstein and Cicchetti, 1990, Byrt et al., 1993], are equally well studied. Our contribution is not the statistic. It is the observation that the agent-evaluation literature has largely not adopted it, that the omission is consequential rather than cosmetic, and that a harness can be built to make the correct practice the default rather than an optional discipline.

We make four contributions.

1. We state and prove the degenerate-judge result: a constant-verdict judge achieves $\kappa = 0$ exactly, with raw agreement equal to majority-class prevalence, for any dataset (Section 3). This gives a precise sense in which raw agreement is uninformative rather than merely optimistic.
2. We demonstrate the gap on 90 hand-labeled transcripts from a tool-using agent suite, contrasting a programmatic judge (81.1%, $\kappa = 0.000$) with a rubric-based LLM judge (96.7%, $\kappa = 0.894$), and characterize the failure modes that separate them (Section 5).
3. We apply the calibrated judge to a model comparison and report a non-significant result with its cause: 15/18 versus 14/18, McNemar exact $p = 1.0$, on a suite where 14 of 18 cases are saturated (Section 6).
4. We describe a harness design in which the calibrate-then-measure protocol is enforced mechanically: judge prompts are content-addressed, so results graded by different judges cannot be silently compared (Section 4).

We also report, in Section 8, a limitation that constrains the strength of our own headline number: our 90 labels are 5 repeated samples of each of 18 cases, so they are clustered rather than independent, and the effective sample size is closer to 18 than to 90.

2 Related Work

Agent benchmarks. Benchmarks for tool-using agents have grown rapidly in scope, from ReAct-style tool-use tasks [Yao et al., 2023] to multi-domain suites such as AgentBench [Liu et al., 2024], web-interaction environments such as WebArena [Zhou et al., 2024], assistant-style tasks with unambiguous answers such as GAIA [Mialon et al., 2024], and software-engineering benchmarks

such as SWE-bench [Jimenez et al., 2024]. These differ substantially in how success is determined. Benchmarks with executable ground truth—a passing test suite, an exact-match answer—sidestep the judge problem entirely, which is a considerable advantage. Benchmarks whose tasks admit many acceptable answers cannot, and it is in that regime that model-based grading becomes necessary and that judge validity becomes the binding constraint.

LLM-as-a-judge and its failure modes. Using a strong model to grade another model’s output was popularized by MT-Bench and Chatbot Arena [Zheng et al., 2023] and by reference-free metrics such as G-Eval [Liu et al., 2023]. Subsequent work has documented systematic biases: sensitivity to the order in which candidates are presented and to superficial features [Wang et al., 2023], and a tendency for judges to favor text generated by themselves or by related models [Panickssery et al., 2024]. A parallel line of work builds dedicated judge models, such as Prometheus [Kim et al., 2024] and JudgeLM [Zhu et al., 2023], and a growing body of meta-evaluation asks how well judges actually track human preference [Chiang and Lee, 2023, Thakur et al., 2024, Bavaresco et al., 2024]. Our concern is orthogonal to which judge one chooses: whatever the judge, the question is what evidence is offered that it works on *this* suite, and whether that evidence is a statistic capable of detecting a degenerate instrument.

Agreement statistics. Cohen’s κ corrects observed agreement for the agreement expected by chance given the raters’ marginal distributions [Cohen, 1960]. The interpretive bands we use—and treat with appropriate suspicion—are those of Landis and Koch [1977]. The statistic’s behavior under skewed marginals is well documented: Feinstein and Cicchetti [1990] describe the paradox in which high percentage agreement yields low κ , and Byrt et al. [1993] decompose the effect into bias and prevalence components, proposing adjusted variants. Artstein and Poesio [2008] survey the use of these coefficients in computational linguistics. Our degenerate-judge proposition is a corollary of standard κ algebra rather than a new result; we state it explicitly because its implication for judge reporting appears to be widely underappreciated in practice.

Statistical practice in evaluation. For comparing two systems on the same items, McNemar’s test on discordant pairs [McNemar, 1947] is the appropriate paired procedure, and Dietterich [1998] recommends it for classifier comparison specifically because it respects the pairing. Broader critiques of statistical practice in NLP evaluation—underpowered comparisons, unreported variance, significance testing applied incorrectly—are given by Dror et al. [2018] and Card et al. [2020]. For interval estimation of pass rates we use the Wilson score interval [Wilson, 1927], following the recommendation of Brown et al. [2001] against the normal approximation at small n and extreme proportions.

3 Background: Why Raw Agreement Fails

Let a judge and a human each assign a binary verdict, pass or fail, to each of n transcripts. Let p_o denote observed agreement, the fraction of items on which the two assign the same label. Cohen’s κ is

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{1}$$

where p_e is the agreement expected if the two raters assigned labels independently at their observed marginal rates:

$$p_e = P_h(\text{pass}) P_j(\text{pass}) + P_h(\text{fail}) P_j(\text{fail}). \quad (2)$$

Here P_h and P_j are the marginal distributions of the human and judge respectively. The construction asks a specific question: of the agreement available beyond chance, how much was achieved?

Proposition 1 (Degenerate judge). *Let a judge assign the same verdict to every item. Then $\kappa = 0$ exactly, regardless of the data, and p_o equals the prevalence of that verdict in the human labels.*

Proof. Suppose without loss of generality that the judge passes every item, so $P_j(\text{pass}) = 1$ and $P_j(\text{fail}) = 0$. Let $\pi = P_h(\text{pass})$. The judge and human agree exactly on the items the human passed, so $p_o = \pi$. Substituting the judge’s marginals, $p_e = \pi \cdot 1 + (1 - \pi) \cdot 0 = \pi$. Hence $\kappa = (\pi - \pi)/(1 - \pi) = 0$ for any $\pi < 1$. The argument for a judge that fails every item is symmetric. $\square$

Proposition 1 is elementary, but its consequence is worth stating plainly. On a suite where 80% of transcripts pass, a judge that does nothing but print “pass” reports 80% agreement with human labels. If judge validation is reported as raw agreement, this judge is indistinguishable in the report from one that has genuinely learned the task. The chance-corrected statistic separates them immediately, because p_e absorbs exactly the part of the agreement that the base rate explains.

The converse caution applies as well, and we state it because we rely on κ throughout. Under highly skewed marginals, κ can be low even for a judge that is substantively useful, since a single misclassification in a rare class moves the statistic sharply [Feinstein and Cicchetti, 1990]. κ is a better default than raw agreement for judge validation, not a sufficient statistic for all purposes. The confusion matrix should be reported alongside it; we do so below.

4 Harness Design

The experiments in this paper were run with an open-source harness we developed for the purpose. We describe it briefly, because two of its design choices are what make the protocol enforceable rather than merely recommended.

Protocol-oriented decoupling. The evaluation core depends on three structural interfaces—an agent, a tool provider, and a judge—expressed as Python protocols rather than base classes. An agent is anything implementing `run(case) → Transcript`. Because protocols are satisfied structurally, an existing agent implementation satisfies the interface without importing the harness or inheriting from it. The practical consequence is that the runner, the statistics, and the calibration machinery contain no provider-specific code; reference adapters for two commercial APIs exist as peers, not as privileged implementations.

Budget enforcement before spending. Step and token budgets are checked at the top of each iteration of the agent loop, before the next API call is issued, rather than after the fact. A post-hoc check can report that a run exceeded its budget but cannot prevent the expenditure. The runner retains a post-hoc check as a backstop for agents that are not under the harness’s control.

Judge	n	Raw agreement	Cohen’s κ
Assertion judge	90	73/90 (81.1%)	0.000
LLM judge (v1)	90	87/90 (96.7%)	0.894

Table 1: Agreement with human labels. Raw agreement differs by 15.6 percentage points; the chance-corrected statistic separates a chance-level instrument from an almost-perfect one.

Content-addressed judge identity. This is the mechanism that enforces the protocol. Every stored result is keyed by the triple (commit SHA, agent model, judge version). Built-in judge prompts are versioned and frozen. A user-supplied judge prompt is identified not by its filename but by a hash of its contents, so that editing a prompt necessarily produces a new judge version. The regression machinery can therefore never compare a run graded by one judge against a run graded by another and attribute the difference to the agent. This closes a failure mode that is easy to hit in practice: an evaluator tunes a judge prompt, re-runs the suite, observes a changed pass rate, and reports it as a change in agent behavior.

Deterministic tool execution. Tool responses can be mocked from fixtures or recorded and replayed, keyed by tool name and a canonical hash of the arguments. Replay pins tool results but not model sampling; we return to this limitation in Section 8.

5 Experiment 1: Judge Calibration

5.1 Setup

Our labeled set comprises 90 transcripts produced by a tool-using agent on an 18-case suite, sampled 5 times per case. Each transcript was labeled pass or fail by a single human annotator against the case’s stated success criterion. The resulting distribution is 73 pass and 17 fail, a pass prevalence of 81.1%. All 17 failures are concentrated in 5 of the 18 cases; the remaining 13 cases were labeled pass on every sample.

We compare two judges.

The **assertion judge** applies programmatic criteria only: a regular expression over the final answer, and the presence or absence of required and forbidden tool calls. It performs no natural-language understanding. On this suite, its criteria reduce in effect to whether the required tools were invoked.

The **LLM judge** grades the transcript against the case’s natural-language rubric. It is invoked with a forced tool call so that its verdict is returned as a structured object with three fields: a boolean verdict, a free-text justification, and a self-reported confidence. The prompt is frozen as version v1; it was not revised after seeing results, and no second version was required.

5.2 Results

Table 1 gives the headline comparison. The confusion matrices explain it.

The assertion judge passes every transcript. Its verdict is constant, so by Proposition 1 its κ is zero by construction, and its raw agreement is exactly the pass prevalence, $73/90 = 81.1\%$. It detects none of the 17 human-labeled failures. No amount of raw-agreement reporting would reveal this; the confusion matrix and κ reveal it immediately.

Assertion judge	Human pass	Human fail
Judge pass	73	17
Judge fail	0	0
LLM judge (v1)	Human pass	Human fail
Judge pass	71	1
Judge fail	2	16

Table 2: Confusion matrices. The assertion judge’s verdict column is constant, placing it exactly in the regime of Proposition 1.

The LLM judge disagrees with the human on 3 of 90 transcripts: one false pass and two cases where it was stricter than the annotator. Its κ of 0.894 falls in the band Landis and Koch [1977] label almost perfect. We attach the usual caution to that label—the bands are conventions, not thresholds with inferential content—but the contrast with 0.000 does not depend on where the band boundaries are drawn.

5.3 What the LLM judge catches

The 17 transcripts that the assertion judge blind-passed are not arbitrary. They concentrate in failure modes that are invisible to structural checks, and each was justified by the assertion judge with the same string: *all required tools were called*.

Partial answers. The task required two pieces of information and the agent supplied one. The required tool was invoked, so a structural check passes; the answer does not satisfy the criterion.

Fabrication after tool error. The tool returned an error and the agent produced a confident answer regardless. From a structural standpoint the tool was called and the run terminated normally. This is arguably the most serious failure class in the set, and it is precisely the one that tool-invocation checks cannot see.

Hedging where commitment was required. The rubric demanded a specific value; the agent returned a qualified non-answer.

The common structure is that each requires reading the transcript against an intent, which is what the rubric-based judge does and the assertion judge does not. This is not an argument that programmatic checks are useless—they are cheap, deterministic, and free of model bias—but an argument about what they can be trusted to certify. On this suite they certify tool invocation, and tool invocation is not the criterion of interest.

6 Experiment 2: Model Comparison Under a Calibrated Judge

6.1 Setup

Having a judge with measured $\kappa = 0.894$, we used it—and only it—to ask a downstream question: do two models of different capability tiers differ on this suite? Both models ran the same 18-case suite at 5 samples per case. Both were graded by the *same* judge, the calibrated v1 judge backed by the smaller model, held fixed across both runs.

Holding the judge constant is a deliberate choice with a cost. The stronger of the two candidate models would ordinarily be the more attractive judge, but it was not the model we calibrated, and

Model	Cases passed	Pass rate
Smaller model	15 / 18	83%
Larger model	14 / 18	78%

Table 3: Single-run pass counts under the calibrated judge. McNemar’s exact test on the paired outcomes gives $p = 1.0$.

substituting it would discard the calibration evidence entirely. A judge is validated as a specific instrument, not as a family.

6.2 Results

The aggregate counts favor the smaller model, 15 to 14. McNemar’s exact test on the discordant pairs returns $p = 1.0$: the difference is not distinguishable from noise. Only one case differed between the two models, and a single discordant pair carries essentially no evidence in either direction. We use the exact binomial form of the test rather than the χ^2 approximation, since the number of discordant pairs is far below the regime where the approximation is reliable [Dietterich, 1998].

We emphasize what a less careful analysis would have reported here. The pass counts differ. A single-run comparison, reported as “83% versus 78%,” would have licensed the conclusion that the smaller model is better at agentic tool use—a surprising and therefore attention-getting claim, and an unsupported one. The paired test and the repeated sampling are what prevent it.

6.3 Why the comparison is uninformative, and how we know

The null result has an identifiable cause rather than being a bare absence of evidence. Fourteen of the 18 cases pass on every sample for both models. The suite is saturated: it lacks the dynamic range to discriminate between systems at this capability level, so most of its cases contribute no information to the comparison. This is a property of the instrument, not of the models, and it is detectable from the per-case pass rates without reference to either model’s identity.

The per-case view also shows behavioral differences that the aggregate conceals. On one case whose rubric required committing to a specific value, the smaller model passed 4 of 5 samples while the larger model passed 0 of 5, consistently hedging where the task penalized hedging. Two cases showed intermediate pass rates across samples and were flagged as flaky by the harness: the agent is genuinely inconsistent on them, which a single run would have rendered invisible as either a clean pass or a clean fail.

We report the non-significant result rather than searching for a case subset that would produce a significant one. Given a saturated suite and 4 informative cases, such a subset could likely be found, and it would not mean anything.

7 Discussion: Calibrate, Then Measure

The two experiments compose into a protocol we would advocate as a default for model-graded agent evaluation.

First, validate the judge on the suite it will grade. Judge validity is not a property of a model but of a model-prompt-suite triple. A judge that performs well on summarization quality has demonstrated nothing about grading tool-use transcripts.

Second, report the chance-corrected statistic and the confusion matrix, not raw agreement. Proposition 1 shows that raw agreement cannot distinguish a working judge from a degenerate one under class imbalance, and agent suites are class-imbalanced by design.

Third, hold the validated judge fixed. If the judge changes, the measuring instrument has changed, and results are not comparable across the change. We enforce this mechanically by content-addressing judge prompts, so a modified prompt yields a different judge version and cannot be diffed against results from the original.

Fourth, sample repeatedly and test pairwise. Agent behavior is stochastic; single-run comparisons confound sampling variance with capability differences.

Fifth, report the null result and diagnose it. A non-significant comparison accompanied by a saturation diagnostic tells a reader something actionable—build a harder suite—whereas a significant comparison from an uncalibrated judge tells them nothing they can rely on.

None of these steps is novel in isolation; they are standard measurement practice. The argument of this paper is that the first two are not currently routine in agent evaluation, that their omission is consequential, and that the tooling should make the correct path the path of least resistance.

8 Limitations

We state the limitations of our own results in the same spirit.

Clustered labels. Our 90 transcripts are 5 repeated samples of each of 18 cases, not 90 independent items. Cohen’s κ assumes independent units, so treating $n = 90$ as the effective sample size overstates precision. Transcripts from the same case share a task, a rubric, and a tool fixture, and are therefore correlated. The effective sample size is nearer 18 than 90, and a clustered or case-level analysis would give a wider uncertainty interval around 0.894 than an item-level one. We report no confidence interval on κ for this reason, and we regard the point estimate as indicative rather than precise.

Single annotator. All 90 labels come from one person, who is also an author. There is no inter-annotator agreement figure, so what we call ground truth is one individual’s reading of the rubrics, and the LLM judge’s κ measures agreement with that individual rather than with a consensus. The annotator’s familiarity with the suite is a plausible source of bias in both directions.

Concentrated failures. All 17 failures occur in 5 of the 18 cases. The judge’s demonstrated ability to detect failures is therefore an ability to detect these particular failure modes, and generalization to other modes is unevidenced.

Clean-room suite. The tasks have explicit rubrics and deterministic mocked tools. This makes the judge’s job substantially easier than grading open-ended interaction with real, noisy tools. We regard 0.894 as an optimistic bound rather than a transferable estimate.

Saturation limits Experiment 2. As reported above, 14 of 18 cases are uninformative for the model comparison. The correct reading of the null result is that this suite cannot separate these models, not that the models are equivalent.

Replay pins tools, not sampling. Recorded tool responses make tool behavior deterministic, but model sampling remains live, so exact reproduction of a transcript is not guaranteed.

Single suite, single domain, two models. All results come from one 18-case suite. We make no claim about other suites, task families, or model families.

9 Conclusion

Model-based evaluation of agents makes the judge part of the measuring apparatus, and an unvalidated apparatus yields unvalidated results. We showed that the statistic most commonly used to report judge quality, raw agreement, is incapable in principle of detecting a degenerate judge under the class imbalance that agent suites exhibit by construction, and we exhibited a judge in exactly that regime: 81.1% agreement, $\kappa = 0.000$, zero of 17 failures detected. A rubric-based judge on the identical transcripts reached $\kappa = 0.894$. We then used the calibrated judge to compare two models, found no significant difference, and identified suite saturation as the reason.

The practical recommendation is narrow and cheap to adopt: report Cohen’s κ and the confusion matrix for your judge, on your suite, before reporting anything the judge produced. The harness, the labeled transcripts, and the commands needed to reproduce both experiments are available at <https://github.com/salasya2/ae hf>.

Reproducibility

Both experiments are reproducible from the published package. Judge calibration requires no model access for the assertion baseline, which is deterministic and runs offline. The commands, the frozen v1 judge prompt, and the labeled transcript set are included in the repository.

References

- Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008.
- Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K. Surikuchi, Ece Takmaz, and Alberto Testoni. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. *arXiv preprint arXiv:2406.18403*, 2024.
- Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–133, 2001.
- Ted Byrt, Janet Bishop, and John B. Carlin. Bias, prevalence and kappa. *Journal of Clinical Epidemiology*, 46(5):423–429, 1993.
- Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. arXiv:2010.06595.
- Cheng-Han Chiang and Hung-yi Lee. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.
- Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

- Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker’s guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- Alvan R. Feinstein and Domenic V. Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549, 1990.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.06770.
- Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.08491.
- J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2308.03688.
- Yang Liu, Dan Iter, Yichong Xu, Shuhang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023. arXiv:2303.16634.
- Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
- Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2311.12983.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*, 2024.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. *arXiv preprint arXiv:2406.12624*, 2024.
- Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.
- Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.03629.
- 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. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023. arXiv:2306.05685.
- Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2307.13854.
- Lianghui Zhu, Xinggang Wang, and Xinlong Wang. JudgeLM: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*, 2023.