智能知识
Human Intelligence

August 2026

Auditable LLM Judges for Complex Agent Evaluation

Research Team
Human Intelligence Research Institute

Abstract

Large-language-model (LLM) judges are increasingly used to evaluate open-ended agent outputs, yet production systems rarely isolate which parts of a judging pipeline improve correctness, auditability, or efficiency. A lower score is not necessarily a more accurate score, and agreement among model runs is not evidence of correctness. We therefore construct a Golden reference at both criterion and task levels and use it to evaluate four independently controllable design choices under frozen tasks, artifacts, rubrics, and aggregation semantics: judge model and reasoning effort, rubric batch size, evidence-acquisition strategy, and score-aggregation strategy. Statistical uncertainty is estimated by cluster bootstrap at the task level, with Holm correction within each experimental family.

The experiments reveal four distinct effects. First, a newer judge at medium reasoning effort is numerically stronger than the earlier baseline, but increasing the same judge from medium to xhigh does not yield a reliable gain after multiplicity correction. Second, smaller rubric batches materially reduce final-score error and positive bias. A single-criterion batch lowers mean absolute error (MAE) from 10.29 to 6.21 points, but requires 36.1 times as many judge calls; a batch size of ten reaches 7.03 MAE at approximately 4.0 times the call count and offers a more practical operating point. Third, requiring locatable evidence raises evidence coverage from 12.0% to 100% without a statistically established accuracy gain, whereas extracting an evidence ledger before scoring significantly worsens criterion-level accuracy and pass/fail stability. Fourth, allowing an LLM to aggregate frozen criterion verdicts does not outperform a deterministic formula and adds roughly 94 seconds per scoring unit. These results argue for a modular judge architecture in which correctness, auditability, and cost are measured separately: rubric batching controls accuracy, evidence requirements support review, and deterministic code owns the final machine score.

Keywords: LLM-as-a-Judge; agent evaluation; rubric-based evaluation; evidence auditing; score aggregation; evaluator reliability

Authors: Hongyi Yang; Yiqiao Huang
Email: yanghongyi@human-intelligence.cn; huangyiqiao@human-intelligence.cn
Research contact: research-team@human-intelligence.cn

1 Introduction

LLM-based evaluation has become a practical response to the difficulty of assessing open-ended model outputs at scale. MT-Bench and Chatbot Arena [Zheng et al., 2023], together with G-Eval [Liu et al., 2023], demonstrated substantial agreement between strong LLM judges and human preferences, making model-based assessment attractive for research and deployment. Subsequent work, however, has documented position bias [Shi et al., 2025], length and superficial-quality preferences [Wang et al., 2024a, Dubois et al., 2024], broader source and authority effects [Chen et al., 2024, Ye et al., 2024a], and sensitivity to prompt and contextual design [Chiang and Lee,

2023, Xu et al., 2025]. A recent survey consequently frames LLM judging not as a solved metric but as a reliability problem that requires explicit validation [Gu et al., 2026].

That reliability problem is sharper for agents than for short-form text generation. Agent outputs may include tool traces, files, structured artifacts, screenshots, and final answers; the judge must connect evidence across these sources, apply hard gates, score multiple criteria, and combine those scores into a task-level outcome. AgentBench [Liu et al., 2024] studies agents across interactive environments; WebArena [Zhou et al., 2024] and VisualWebArena [Koh et al., 2024] focus on realistic web interaction; GAIA [Mialon et al., 2024] targets general-assistant tasks; SWE-bench [Jimenez et al., 2024] evaluates repository-level software engineering; and OSWorld [Xie et al., 2024] evaluates open-ended desktop control. When evaluation is not fully executable, a seemingly singular "judge" is therefore a pipeline with at least four layers: the evaluator model, the amount of rubric shown per call, the path by which evidence is acquired, and the rule that aggregates criterion verdicts.

Most evaluation reports compare mean scores, win rates, or repeated-run variance. These quantities are useful for describing behavior, but they do not distinguish strictness from accuracy. If a rubric decomposition produces lower scores, the judge may have corrected lenient over-scoring or introduced new under-scoring. Likewise, repeated agreement may indicate a stable error rather than a correct decision. A fixed Golden reference is required to tell whether a design change moves predictions toward or away from the intended score.

This paper studies four pipeline layers against the same Golden reference. Within each experimental family, we keep tasks, artifacts, rubrics, evaluation slots, and scoring semantics frozen while varying only the target factor. Our contributions are:

- a criterion-level and task-level Golden alignment protocol that separates model consensus from source-grounded review;
- a controlled comparison of judge model and reasoning effort, rubric batch size, evidence acquisition, and score aggregation within one evaluation stack;
- paired estimates of error, signed bias, pass/fail flips, hard-gate errors, auditability, latency, and a transparent call-count cost proxy; and
- deployment recommendations that treat correctness, auditability, and efficiency as separate engineering objectives.

The central conclusion is not that a single model or prompt universally solves automated evaluation. Reliability improves when the pipeline is decomposed into testable modules: a frozen Golden target defines correctness, rubric batching controls cognitive load, evidence locators support audit, and deterministic code preserves reproducibility at the aggregation layer.

2 Related Work

2.1 LLMs as evaluators

LLM-as-a-Judge spans scalar scoring, pairwise preference, critique generation, and benchmark meta-evaluation. MT-Bench and Chatbot Arena [Zheng et al., 2023] and G-Eval [Liu et al., 2023] established strong model judges as viable reference-free evaluators. A closer analysis of automatic evaluation showed that quality depends on prompting details and that explanations can improve agreement in some natural-language-generation settings [Chiang and Lee, 2023]. Dedicated evaluators include PandaLM [Wang et al., 2024b], Prometheus [Kim et al., 2024a], and Prometheus 2 [Kim et al., 2024b]; these systems pursue more reproducible evaluation, rubric conditioning, or support for direct and pairwise assessment. JudgeBench [Tan et al., 2025] and ContextualJudgeBench [Xu et al., 2025] demonstrate that strong evaluators can still struggle

on objectively difficult or context-dependent comparisons. Fine-tuned judges may also fail to generalize beyond the domains used to train them [Huang et al., 2025].

Reliability failures are not reducible to random noise. Position bias has been isolated directly [Shi et al., 2025]; other studies identify superficial-quality preferences, self-enhancement, and authority effects [Wang et al., 2024a, Chen et al., 2024, Ye et al., 2024a]; and length-controlled evaluation shows that candidate length itself can confound automatic preference judgments [Dubois et al., 2024]. Panels of heterogeneous judges have been proposed as a way to reduce correlated errors [Verga et al., 2024], but additional models increase cost and do not eliminate the need for an external target. Our work complements these studies by evaluating modular system interventions against a frozen criterion-level Golden reference rather than treating agreement with another model as ground truth.

2.2 Rubric-based and fine-grained evaluation

Fine-grained evaluation decomposes an overall judgment into explicit dimensions or skills. FLASK evaluates language-model responses across alignment skill sets [Ye et al., 2024b]; Prometheus conditions on customized score rubrics [Kim et al., 2024a]; and LLM-Rubric learns calibrated combinations of multidimensional judgments [Hashemi et al., 2024]. This decomposition improves interpretability but creates a systems question that has received less direct attention: how many criteria should the judge see in one call? A large batch is efficient but may introduce criterion interference, anchoring drift, or missed checks; a small batch isolates decisions but multiplies calls. We study this accuracy–cost frontier while holding the rubric and final scoring rule fixed.

2.3 Evidence, rationales, and auditability

Evidence-grounded evaluation is closely related to work on factual attribution and rationale faithfulness. FActScore decomposes long-form generations into atomic claims [Min et al., 2023]; RAGAs evaluates retrieval-augmented generation along separable dimensions [Es et al., 2024]; and RARR retrieves support and revises unsupported text [Gao et al., 2023]. ERASER provides a benchmark for rationalized NLP models [DeYoung et al., 2020], while subsequent work distinguishes a rationale's plausibility to a reader from its faithfulness to the model's decision process [Jacovi and Goldberg, 2020, Lyu et al., 2024]. Chain-of-thought prompting [Wei et al., 2022] and self-consistency [Wang et al., 2023] can improve task performance, yet generated explanations may rationalize decisions without revealing their true cause [Turpin et al., 2023]. We therefore treat evidence coverage as an auditability measure, not as a proxy for score correctness, and evaluate correctness separately against Golden labels.

2.4 Human labels, aggregation, and statistical dependence

Reference construction from multiple assessors has a long history in annotation research. The Dawid–Skene model estimates latent labels from annotator-specific error rates [Dawid and Skene, 1979], while inter-coder agreement research emphasizes that reliability statistics and construct validity answer different questions [Artstein and Poesio, 2008]. Our Golden process uses model consensus only for triage; disputed or high-impact items require source-grounded rule application under a frozen policy. For inference, repeated slots and criteria from the same task are dependent. We therefore use the bootstrap framework [Efron and Tibshirani, 1993] and resample at the task cluster, following clustered-inference principles [Cameron et al., 2008]. Family-wise multiplicity is controlled by Holm's sequential procedure [Holm, 1979].

3 Study Design

3.1 Evaluation unit and Golden reference

Let r denote a task, m an independent evaluation slot, and c a rubric criterion. The smallest scoring unit is (r, m, c). For each unit, a judge produces a raw score, a pass/fail verdict, a reason, and optionally a locatable evidence reference. A fixed aggregation function maps criterion outputs and hard-gate state $h_{r,m}$ to a task-level score:

$$\hat{y}_{r,m} = g(\{\hat{y}_{r,m,c}\}_c, h_{r,m}) \,. \tag{1}$$

The target is not a higher or lower $\hat{y}$, but a smaller discrepancy from the Golden reference y^*.

Each task contributes three independently produced artifacts. A task has a median of 36 criteria (interquartile range [IQR], 35–37; range, 32–44) and a median of 6 hard-gate criteria (IQR, 5–7; range, 3–14). Criterion sets are identical across the three artifacts for each task, and no duplicate task–slot–criterion keys are present. The workload is an internal, mixed multi-artifact agent workload that combines structured files, free-form documents, and execution traces. We therefore state conclusions for the observed workload rather than treating it as a universal benchmark for all agent domains.

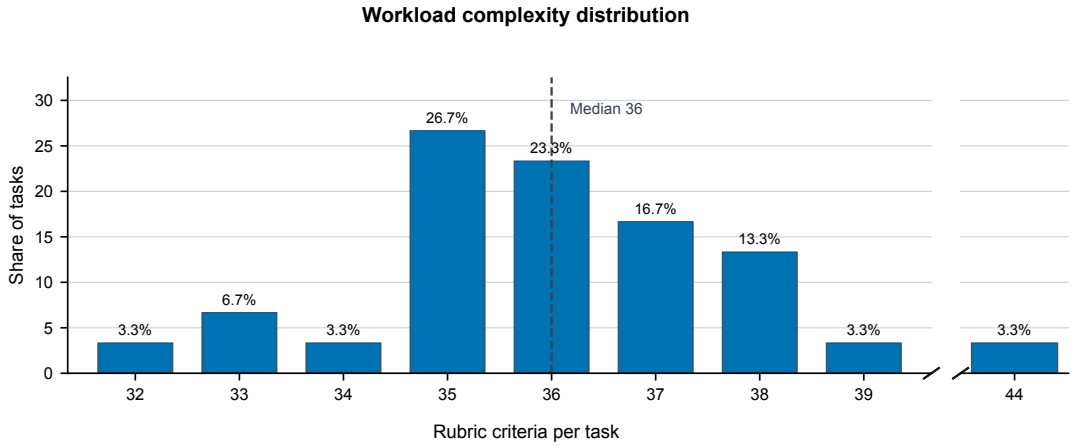

Figure 1. Normalized distribution of rubric criteria per task. Shares are reported instead of task counts; the dashed line marks the median.

The Golden evidence fields contain a median of 80 locatable evidence references per task artifact (IQR, 67–92; range, 51–136). This quantity measures the evidence-localization workload faced by a judge; it is not a count of bibliographic citations.

The Golden reference is frozen for each task–artifact–rubric–slot combination under a human-approved, versioned review policy. Every unit first receives a blind source-only review using only the query, frozen inputs, the current artifact, and the rubric. Historical medium- and xhigh-effort judge outputs are revealed only after this decision and are used to route discrepancies; they are never votes or ground truth. A second, fresh source-only review is run independently, and only units routed by the predeclared evidence, confidence, gate, and rubric-structure rules use that review for selection. A final source-only pass is restricted to the remaining pre-frozen high-risk queue. Score labels must have reproducible locators and valid input hashes before deterministic aggregation produces the task-level Golden score. The policy, prompt, schema, scorer, and source manifests are all identified by SHA-256 hashes.

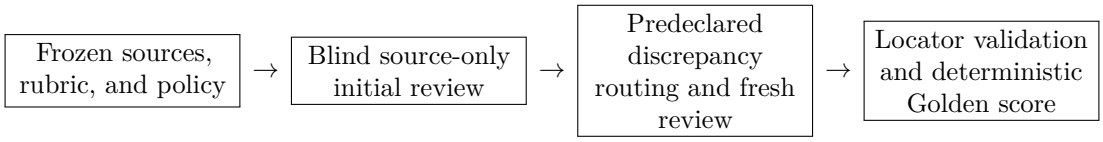

Figure 2. Golden construction workflow. Historical judge outputs only route discrepancies after the initial blind decision; they never determine the target by voting.

3.2 Controlled factors

Judge model and reasoning effort. We compare GPT-5.5 at medium reasoning effort, GPT-5.6-sol at medium effort, and GPT-5.6-sol at xhigh effort. These exact names and effort levels are provenance labels from the frozen experiment manifest, not claims about a separately published model release. The task material, rubric, evaluation slots, and aggregation function are unchanged. This factor tests whether a newer evaluator or a larger reasoning budget reliably reduces Golden-aligned error. Chain-of-thought prompting [Wei et al., 2022] and repeated-sampling strategies [Wang et al., 2023] can improve some tasks, but generated reasoning need not be faithful to the model's decision process [Turpin et al., 2023]; reasoning effort is therefore treated as an empirical intervention rather than an assumed monotonic improvement.

Rubric batch size. We compare full-rubric evaluation with batches containing 1, 2, 5, or 10 criteria. All conditions use GPT-5.6-sol at medium effort and the same deterministic merge rule. The only change is how many rubric criteria are presented in one judge call. We use a relative judge-call multiplier to compare the resource scale of batching configurations under frozen model and workload conditions. This proxy supports within-study engineering comparisons and is not interpreted as a billing estimate.

Evidence-acquisition strategy. The *direct* arm reads the original inputs and scores them directly. The *evidence-required* arm reads the same materials but must return a locatable evidence reference for every criterion. The *extract-then-score* arm first compresses the material into an evidence ledger and then scores only from that ledger. The comparison tests whether structured evidence primarily improves auditability or also improves Golden-aligned accuracy.

Score-aggregation strategy. All aggregation strategies receive exactly the same frozen criterion verdicts. *Deterministic aggregation* applies the production scoring formula. *LLM aggregation* allows a language model to derive the final score. *LLM explanation + deterministic score* uses the model only for prose while retaining the fixed machine-readable score. This design isolates aggregation from criterion judgment.

3.3 Experimental control matrix

Table 1 makes the single-factor comparisons explicit. "Frozen" means frozen within an experimental family. The four families share aligned task, artifact, rubric, and slot keys, while separate judge executions remain stochastic runs rather than interchangeable cached outputs. In particular, the direct condition in the evidence family is an independently executed family-specific baseline; its final MAE is therefore not expected to equal the full/direct/deterministic run reused as the baseline in the model, batching, and aggregation families.

Table 1. Experimental control matrix. The varied factor is shown in bold.

Family	Varied levels	Fixed judge, batch, and evidence settings	Aggregation
Model / effort	**5.5 medium; 5.6-sol medium; 5.6-sol xhigh**	Full rubric; direct source access	Deterministic
Rubric batching	**Full; 1; 2; 5; 10**	5.6-sol medium; direct source access	Deterministic
Evidence acquisition	**Direct; required locators; extracted ledger**	5.6-sol medium; full rubric	Deterministic
Score aggregation	**Deterministic; LLM; explanation-only LLM**	Frozen 5.6-sol medium verdicts; full-rubric/direct upstream provenance	Varied

3.4 Metrics

The primary metric is mean absolute error:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i^*| \,. \tag{2}$$

MAE is the average distance between a predicted score and the Golden score, expressed in score points at the final level and in normalized units at the criterion level. For example, Golden scores of 80 and 70 paired with predictions of 86 and 66 have absolute errors of 6 and 4, hence an MAE of 5. Lower is better; zero denotes exact agreement.

Because MAE discards direction, we also report signed bias, $N^{-1}\sum_i(\hat{y}_i - y_i^*)$. Positive bias indicates systematic over-scoring and negative bias indicates under-scoring. Additional correctness measures include criterion MAE, criterion bias, pass/fail flip rate, and hard-gate error. The evidence-acquisition comparison additionally reports evidence coverage and the fraction of evidence locators that can be parsed. The aggregation comparison reports mean execution time. These measures intentionally separate correctness, auditability, and efficiency.

3.5 Statistical inference and integrity controls

Confidence intervals and paired contrasts use cluster bootstrap with the task as the resampling unit. This preserves dependence among repeated slots and criteria from the same task [Efron and Tibshirani, 1993, Cameron et al., 2008]. The random seed is fixed. Comparisons are grouped by experimental family, and p-values are adjusted with Holm's procedure; adjusted $p < 0.05$ is treated as statistically significant [Holm, 1979]. Inputs, artifacts, rubrics, model settings, and source provenance are hash-validated before combination. Re-executed units are accepted only when the frozen-input hashes and experimental configuration match the original condition.

4 Results

4.1 Higher reasoning effort does not produce a reliable gain

GPT-5.6-sol medium is numerically better than GPT-5.5 medium on final MAE, signed bias, hard-gate error, and criterion MAE (Table 2). However, none of the model-family paired contrasts remains significant after Holm correction. Increasing GPT-5.6-sol from medium to xhigh is also non-monotonic: xhigh slightly lowers criterion MAE and pass/fail flips, but increases

final MAE, positive bias, and hard-gate error. The evidence therefore does not support xhigh as a default accuracy intervention.

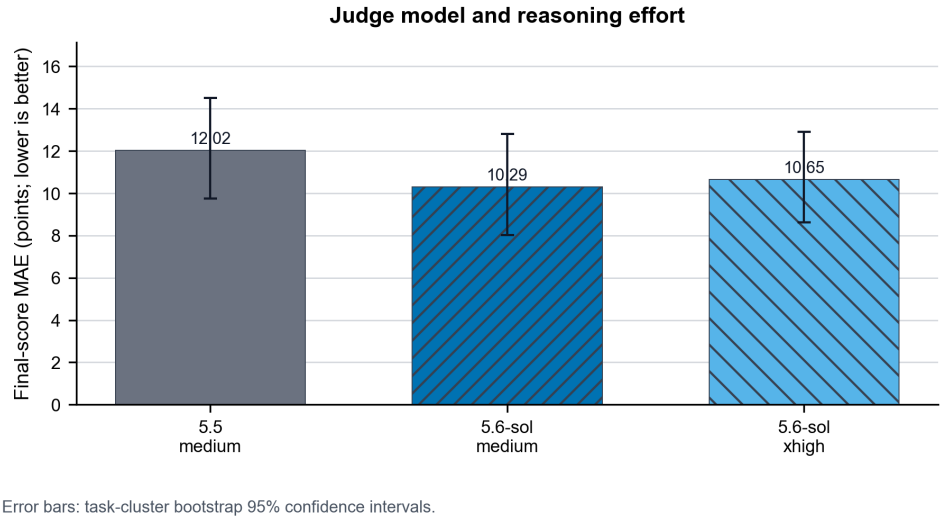

Figure 3. Golden-aligned final-score error across judge models and reasoning efforts. Lower MAE is better; error bars are task-cluster bootstrap confidence intervals.

Table 2. Judge model and reasoning-effort comparison.

Configuration	Final MAE	Bias	Gate err.	Crit. MAE	Flip
GPT-5.5 medium	12.02	+9.32	22.62%	.1078	8.72%
GPT-5.6-sol medium	**10.29**	**+6.76**	**15.48%**	.0926	7.76%
GPT-5.6-sol xhigh	10.65	+8.69	19.05%	**.0917**	**7.55%**

4.2 Smaller rubric batches reduce error at a steep call-count cost

Rubric batching yields the clearest accuracy change (Figure 4). Relative to full-rubric judging, a batch size of one reduces final MAE by 4.08 points, with a 95% confidence interval of $[-6.61, -1.91]$ and Holm-adjusted $p = .0099$. Positive bias falls from +6.76 to +0.80 points (adjusted $p = .0020$). The lower score is therefore not merely evidence of a stricter judge: on average, it is closer to the Golden score.

The improvement is expensive. Single-criterion judging requires 36.1 times the calls of the full-rubric condition. Batch size ten reaches a final MAE of 7.03 at approximately 4.0 times the call count, outperforming batch size five on both this error measure and the available cost proxy. Under the observed workload, batch size ten is the more practical production setting, whereas batch size one is an accuracy-first offline setting.

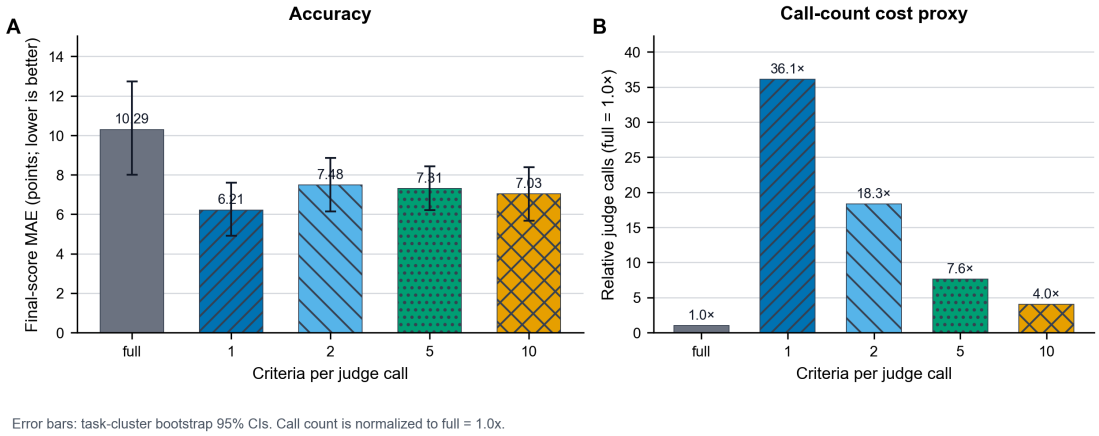

Figure 4. Accuracy–cost trade-off across rubric batch sizes. The cost axis reports the relative judge-call multiplier under frozen model and workload conditions.

Table 3. Rubric batch-size results.

Batch	Final MAE	Bias	Crit. MAE	Flip	Calls
full	10.29	+6.76	.0926	7.76%	1.0×
1	**6.21**	**+0.80**	.0855	7.27%	36.1×
2	7.48	+1.90	**.0816**	**6.90%**	18.3×
5	7.31	+2.76	.0838	**6.90%**	7.6×
10	7.03	+3.76	.0823	7.15%	**4.0**×

4.3 Evidence requirements improve auditability, not established accuracy

Requiring evidence changes observability dramatically. Evidence coverage rises from 12.0% in the direct strategy to 100%, and locator parseability rises from 5.9% to 99.7% (Table 4). These are operationally meaningful gains for review, appeal, and error localization. Yet final MAE, criterion MAE, hard-gate error, and pass/fail flip do not show a significant improvement over direct judging. Evidence-required judging is therefore supported as an audit feature, not as a proven accuracy enhancement.

Extract-then-score performs worse. Relative to direct judging, criterion MAE increases by 0.0535 (95% CI $[0.0309, 0.0749]$, adjusted $p = .0015$), and the pass/fail flip rate increases by 4.56 percentage points (95% CI $[2.45, 6.62]$, adjusted $p = .0018$). A plausible mechanism is an information bottleneck: an independently produced ledger can omit negative evidence, cross-file relationships, or context that later turns out to be criterion-relevant. This explanation is consistent with the observed pattern but requires a separately annotated evidence-recall study for causal confirmation.

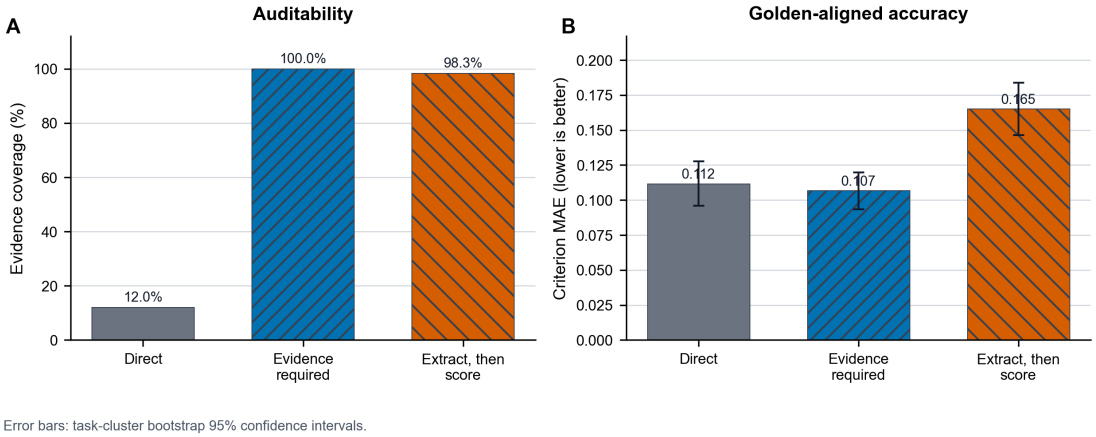

Figure 5. Auditability and Golden-aligned accuracy under three evidence-acquisition strategies.

Table 4. Evidence-acquisition results.

Strategy	Coverage	Locator	Final MAE	Crit. MAE	Flip
Direct	12.0%	5.9%	**11.53**	.1116	9.27%
Evidence required	**100.0%**	**99.7%**	12.78	**.1066**	**8.81%**
Extract then score	98.3%	96.1%	15.04	.1650	13.83%

4.4 Deterministic aggregation is faster, reproducible, and no less accurate

When criterion verdicts are frozen, LLM aggregation does not improve Golden alignment. Its final MAE is 1.46 points higher than deterministic aggregation, although the difference is not statistically significant. It agrees exactly with the deterministic score on 95.56% of units and introduces drift on the remainder, without improving hard-gate decisions. The mean additional execution time is approximately 94 seconds per scoring unit.

The explanation-only arm reproduces the deterministic machine score exactly, as required by design, but adds approximately 109.7 seconds. These results support a strict separation of responsibilities: deterministic code should own the score and hard-gate semantics, while an LLM may generate optional prose asynchronously without score-writing authority.

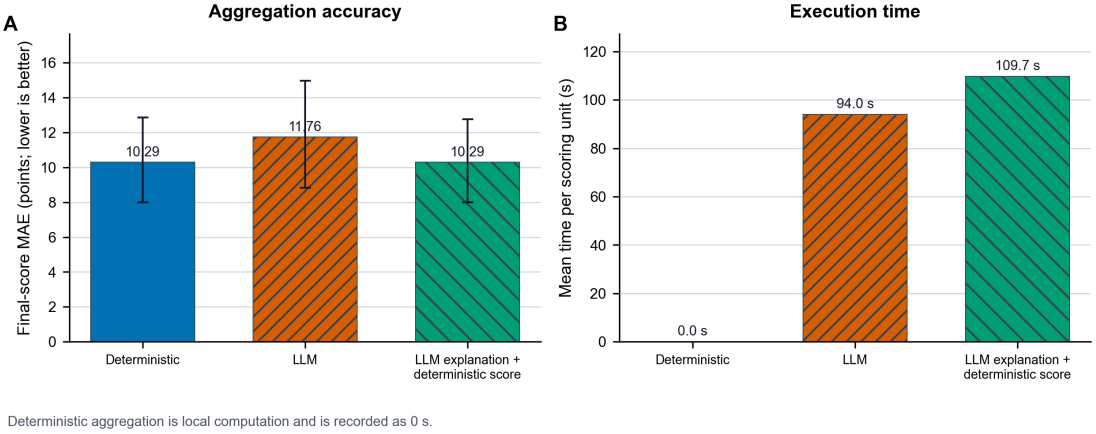

Figure 6. Golden-aligned final-score error and mean latency across aggregation strategies.

Table 5. Score-aggregation results.

Strategy	Final MAE	Bias	Mean time
Deterministic	**10.29**	+6.76	0.0 s
LLM aggregation	11.76	**+4.34**	94.0 s
LLM explanation + deterministic score	**10.29**	+6.76	109.7 s

4.5 Runtime implications of more complex judging pipelines

The evidence-acquisition artifacts preserve partial wall-clock telemetry. Median observed runtime is 465 seconds for direct judging (IQR, 386–603; 90/90 units observed), 609 seconds for evidence-required judging (IQR, 505–753; 84/90), and 881 seconds for extract-then-score (IQR, 806–995; 41/90). Because runtime is missing for a substantial fraction of the latter two strategies, these statistics describe the observed operating scale and are not used as unbiased speed comparisons.

Aggregation timing is complete. Deterministic aggregation is a local computation recorded as approximately zero seconds; median runtime is 86.8 seconds for LLM aggregation (IQR, 69.9–108.0) and 106.1 seconds for LLM explanation with a deterministic score (IQR, 91.5–126.2).

Taken together, the timing results show that auditability and explanation layers should be activated according to product risk. Evidence requirements exchange additional execution time for a stronger review trail, explanation generation is better suited to an asynchronous path, and deterministic aggregation should remain on the synchronous scoring path.

5 Discussion

5.1 Correctness, auditability, and efficiency are separate objectives

The four experiments reject a one-dimensional view of judge quality. Rubric batching primarily changes Golden-aligned correctness but increases calls. Evidence requirements primarily change auditability and reviewability. LLM aggregation changes latency and reproducibility without improving accuracy. A production evaluation system should consequently maintain separate service-level objectives for score error, evidence coverage, execution time, and relative compute budget.

This separation also clarifies why apparently reasonable interventions can disappoint. More reasoning effort can amplify a stable scoring tendency rather than correct it. More evidence fields can produce well-formed locators without changing the underlying verdict. An additional LLM aggregation step can produce a persuasive explanation while adding score drift. These observations are consistent with broader findings that model judges remain sensitive to superficial and contextual factors [Wang et al., 2024a, Dubois et al., 2024, Xu et al., 2025] and that generated rationales should not be equated with faithful decision evidence [Jacovi and Goldberg, 2020, Turpin et al., 2023, Lyu et al., 2024].

5.2 Why rubric decomposition may help

Full-rubric judging asks a model to retain many criteria, evidence locations, score anchors, and gate interactions within one generation. Smaller batches reduce simultaneous competition among instructions and make each local decision easier to audit. The strong reduction in final-score bias suggests that decomposition changes not only isolated criterion judgments but also how local errors accumulate through the scoring formula. This mechanism is plausible but not directly identified: the experiment observes outputs, not internal attention or reasoning states.

The non-monotonic middle of the batch curve is equally important. Accuracy does not

improve in a perfectly smooth sequence as batch size shrinks. Consequently, the correct engineering question is not "how small can the batch be?" but "which batch lies on the empirical accuracy–cost frontier for this workload?" Under the present conditions, ten is the best production compromise among the tested settings, while one remains the accuracy-first extreme.

5.3 Why extract-then-score can fail

Separating extraction from judgment is architecturally appealing because it creates an explicit intermediate artifact. The risk is that the extractor must anticipate every distinction later required by the rubric. Information omitted from the ledger cannot be recovered by the scoring model, even when that model is otherwise strong. Evidence-required direct judging avoids this irreversible compression: it retains access to the original material while producing locators for audit. The result does not imply that all evidence ledgers are harmful; it implies that their recall must be independently validated before they replace source access.

5.4 Deployment recommendations

For the observed task distribution, we recommend GPT-5.6-sol at medium effort, rubric batches of ten, evidence-required output when human review is expected, and deterministic aggregation. An accuracy-first offline tier may use single-criterion batches when the call budget permits. Natural-language summaries may be generated asynchronously from the frozen machine-readable result. We do not recommend xhigh reasoning, extract-then-score, or free-form LLM aggregation as defaults without workload-specific evidence.

6 Limitations and Threats to Validity

First, the number of independent tasks is limited relative to the much larger number of within-task criteria. Cluster bootstrap reduces pseudo-replication but cannot substitute for broader task coverage. Second, the cost analysis uses relative judge-call counts and verified wall-clock fields. These quantities support controlled within-study comparisons but should not be interpreted as direct billing multipliers.

Third, the study covers one family of agent workloads and a small set of proprietary judge configurations. Results may not transfer to other languages, domains, artifact types, or model families. Finally, the proposed explanation for ledger-induced degradation is observational; direct evidence-recall annotation is needed to test the information-bottleneck mechanism.

7 Conclusion

We present a Golden-aligned, controlled study of four design layers in LLM judging for complex agent tasks. A larger reasoning budget does not reliably improve correctness. Smaller rubric batches substantially reduce final-score error and over-scoring, but create a steep call-count trade-off. Mandatory evidence makes decisions dramatically easier to audit without automatically making them more accurate. LLM-based aggregation adds latency and occasional drift while failing to improve over a deterministic formula.

The broader design principle is modularity. Use a Golden reference to measure correctness, evidence locators to measure auditability, and explicit resource accounting to measure efficiency. Keep criterion judgment, evidence review, and score aggregation independently testable. In this architecture, models interpret evidence, but deterministic mechanisms preserve scoring semantics and reproducibility.

A Evaluation Disclosure and Sample Accounting

The release uses 30 independent task clusters, with three separately produced artifacts per task. Across the frozen workload there are 1,084 unique task–criterion definitions. Golden-aligned analyses contain 84 paired final-score units and 3,246 paired task–slot–criterion units per condition. The difference between task-level and criterion-level coverage is handled by retaining only reproducible, fully aggregable Golden score units for final-score metrics; criterion metrics use every reproducible Golden label. The independent task cluster, rather than the much larger criterion count, is the resampling unit.

Table 6. Experimental matrix and analysis-unit accounting. Counts are per condition after provenance and Golden validation.

Family	Conditions	Slots	Final pairs	Criterion pairs	Validation and execution provenance
Model / effort	3	3	84	3,246	Frozen artifacts, rubrics, direct evidence access, and deterministic aggregation; no post-validation gaps in the analysis set.
Rubric batching	5	3	84	3,246	Same model/effort and direct evidence access; only criteria per call varies; no post-validation gaps in the analysis set.
Evidence acquisition	3	3	84	3,246	Full paired coverage after hash-matched completion; 55 isolated re-executions retain explicit provenance and the same frozen configuration.
Score aggregation	3	3	84	3,246	Identical frozen upstream criterion verdicts; deterministic and two LLM-mediated aggregation paths.

Golden review provenance. The Golden policy was approved before final analysis and is identified by SHA-256 `29a4823c8a46ba828914e8c3a9a62c5cdc17d6f7fe2a46ada138aeca033593b4`. Review A used GPT-5.6-sol at medium effort in blind source-only sessions; Review C used a fresh blind source-only pass at high effort; and Review D used xhigh effort only for the pre-frozen final-review queue. Review sessions could not see experimental outputs or one another before committing their decisions. Historical medium/xhigh judge results were restricted to discrepancy routing and never entered the Golden score by voting. Because this is a human-governed, model-assisted protocol rather than two independent human annotations, we do not report a human inter-rater agreement statistic.

Workload disclosure. Task content and internal identifiers are withheld, but the workload-level structure is reported in aggregate: criteria per task have median 36, IQR 35–37, and range 32–44; hard-gate criteria have median 6, IQR 5–7, and range 3–14; and locatable evidence references per artifact have median 80, IQR 67–92, and range 51–136. The workload includes structured files, free-form documents, and execution traces. Shared task-specific rubric templates create within-task dependence, which is why all confidence intervals cluster at the task level.

Reproducibility Statement

The experimental matrix was checked for configuration completeness before analysis. Input, artifact, rubric, and evaluation-slot provenance were hash-validated. Bootstrap resampling used

the task as the cluster and a fixed random seed; multiple comparisons were corrected within experimental families. The public manuscript reports anonymized aggregate statistics and does not disclose task content or internal identifiers.

References

Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. doi: 10.1162/coli.07-034-R2. URL https://aclanthology.org/J08-4004/.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414. URL https://direct.mit.edu/rest/article/90/3/414/57731/Bootstrap-Based-Improvements-for-Inference-with.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Cheng-Han Chiang and Hung-yi Lee. A closer look into using large language models for automatic evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8928–8942, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.599. URL https://aclanthology.org/2023.findings-emnlp.599/.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979. doi: 10.2307/2346806. URL https://www.jstor.org/stable/2346806.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408/.

Yann Dubois, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993. ISBN 978-0-412-04231-7.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.eacl-demo.16. URL https://aclanthology.org/2024.eacl-demo.16/.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. A survey on LLM-as-a-judge. *The Innovation*, 7(6):101253, 2026. doi: 10.1016/j.xinn.2025.101253. URL https://doi.org/10.1016/j.xinn.2025.101253.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.745. URL https://aclanthology.org/2024.acl-long.745/.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979. URL https://www.jstor.org/stable/4615733.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. An empirical study of LLM-as-a-judge for LLM evaluation: Fine-tuned judge model is not a general substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.306. URL https://aclanthology.org/2025.findings-acl.306/.

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages

4198–4205, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL https://aclanthology.org/2020.acl-main.386/.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/edac78c3e300629acfe6cbe9ca88fb84-Abstract-Conference.html.

Seungone Kim, Jay Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Ryan S. Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*, 2024a. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/803485352e61e3ebf41221e4776c9fd4-Abstract-Conference.html.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. VisualWebArena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 881–905, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.50. URL https://aclanthology.org/2024.acl-long.50/.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html.

Yang Liu, Dan Iter, Yichong Xu, Shuohang 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*, pages 2511–2522, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards faithful model explanation in NLP: A survey. *Computational Linguistics*, 50(2):657–723, 2024. doi: 10.1162/coli_a_00511. URL https://aclanthology.org/2024.cl-2.6/.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/25ae35b5b1738d80f1f03a8713e405ec-Abstract-Conference.html.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India, 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. doi: 10.18653/v1/2025.ijcnlp-long.18. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y. Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. JudgeBench: A benchmark for evaluating LLM-based judges. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9e720fce64f91114c49cfd640d821da3-Abstract-Conference.html.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pages 74952–74965, 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models, 2024. URL https://arxiv.org/abs/2404.18796.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450,

Bangkok, Thailand, 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. PandaLM: An automatic evaluation benchmark for LLM instruction tuning optimization. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/be3b0d51a2b86cb4ffe50f13480217e0-Abstract-Conference.html.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5d413e48f84dc61244b6be550f1cd8f5-Abstract-Datasets_and_Benchmarks_Track.html.

Austin Xu, Srijan Bansal, Yifei Ming, Semih Yavuz, and Shafiq Joty. Does context matter? ContextualJudgeBench for evaluating LLM-based judges in contextual settings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9541–9564, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.470. URL https://aclanthology.org/2025.acl-long.470/.

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

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations*, 2024b. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f41b4a6b202adcd8e150a9d4f124d8f6-Abstract-Conference.html.

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*, volume 36, pages 46595–46623, 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

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*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/4410c0711e9154a7a2d26f9b3816d1ef-Abstract-Conference.html.