BEYOND THE VERDICT: WHAT ACCURACY MISSES WHEN CHOOSING LLM JUDGES FOR AGENT EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A coding agent can skip a required action and still claim that it completed it. A judge may accept that claim even though the trajectory shows otherwise. More broadly, when one judge looks better than another, a single accuracy score does not tell us why. The difference may come from the decision cutoff, from how the judge handles misleading trajectory content, or from useful information that never reaches its final verdict. We introduce the *Verdict Audit* to separate these possibilities. We apply it to two practical judge-selection settings. For Qwen3-8B and Qwen3-14B, a large accuracy gap mostly disappears after adjusting the PASS/FAIL cutoff, but the smaller model remains more vulnerable to false completion claims when reasoning is enabled. For Qwen3.6-27B and Gemma 4 31B, Gemma 4 makes better output-level rankings, while Qwen3.6 contains useful error information internally that its final verdict does not use. Reading that information narrows the gap, but the fixes we test do not reliably correct the errors. The result is simple: choosing a judge requires understanding why its decisions differ, not just which model has the higher score.

1 INTRODUCTION

LLM judges are increasingly used to score long agent trajectories, where an evaluator must decide whether a specific requirement was satisfied somewhere in a sequence of actions (Zheng et al., 2023; Liu et al., 2023; Peng et al., 2026). In practice, teams still need to choose which judge to deploy, often using public benchmarks or aggregate judge accuracy as the main evidence.

However, a single score can hide several different problems. First, accuracy mixes ranking quality with the placement of the final PASS/FAIL cutoff. Second, typical examples may not reveal how a judge responds to misleading trajectory content. Third, useful information may remain inside a model even when its verdict is wrong, without being practically usable to correct that verdict.

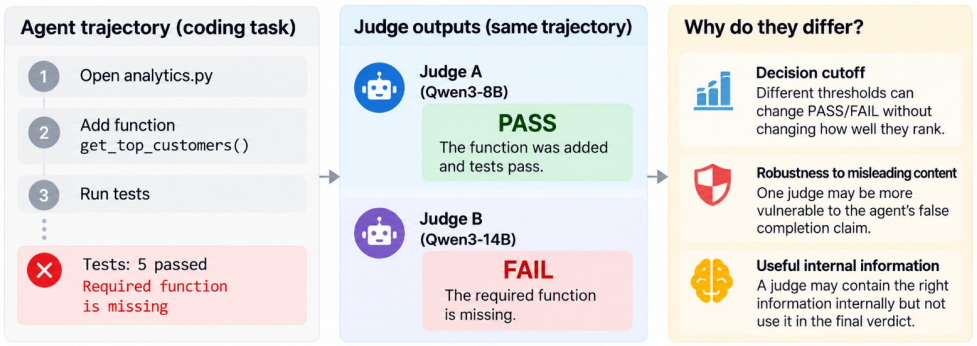
We ask a practical question: *when one judge looks worse than another, or two candidates look broadly similar, what does a team need to know before choosing between them?* We study two settings: **Case A** compares Qwen3-8B with Qwen3-14B as a possible within-family downgrade, while **Case B** compares Qwen3.6-27B with Gemma 4 31B, two strong open-weight candidates whose general-purpose benchmarks do not resolve the target-domain choice.

Our central idea is simple: *select judges by audit, not by score alone*. We introduce a four-stage *Verdict Audit* that moves from output ranking and calibration, to robustness under controlled stress tests, to internal readout, and finally to whether the recovered information can be used for selective correction. The goal is to separate sources of apparent model differences that an aggregate score conflates.

The two cases show why this separation matters. In Case A, the apparent disadvantage of Qwen3-8B largely disappears after recalibration, but stress testing reveals a robustness tradeoff that aggregate accuracy misses. In Case B, Qwen3.6 contains useful judgment information that its verdict does not expose, yet accessing that signal does not produce a demonstrated practical advantage over Gemma 4. The simple interventions we test also do not selectively correct the errors. These results show that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Figure 1: Why a single judge score can be misleading. The same failed trajectory can produce different judge decisions. The difference may arise from threshold placement, robustness to misleading content, or information that does not reach the final verdict.



what appears to be a model-quality difference can instead reflect threshold placement, robustness, or information that exists internally but is not yet practically usable.

2 Related Work

LLM judges and robustness. LLM-as-a-Judge has been studied for open-ended generation evaluation and known judge biases, including position, verbosity, and self-preference (Zheng et al., 2023; Liu et al., 2023; Wataoka et al., 2024). More recent work extends evaluation to agents through benchmarks such as RuVerBench, AgentRewardBench, and Counsel (Peng et al., 2026; Lù et al., 2025; Pisupati et al., 2026). A parallel literature shows that LLM-based evaluation can be manipulated through optimized prompt injection, universal adversarial strings, benchmark gaming, and persuasive or sycophantic interactions (Shi et al., 2024; Raina et al., 2024; Zheng et al., 2025; Kim & Khashabi, 2025). These works characterize judge accuracy, bias, and robustness under a range of evaluation conditions. We instead compare candidate judges under plausible trajectory failures—in particular, ordinary agent-generated false completion claims without adversarial optimization—and ask whether robustness changes the model-selection conclusion.

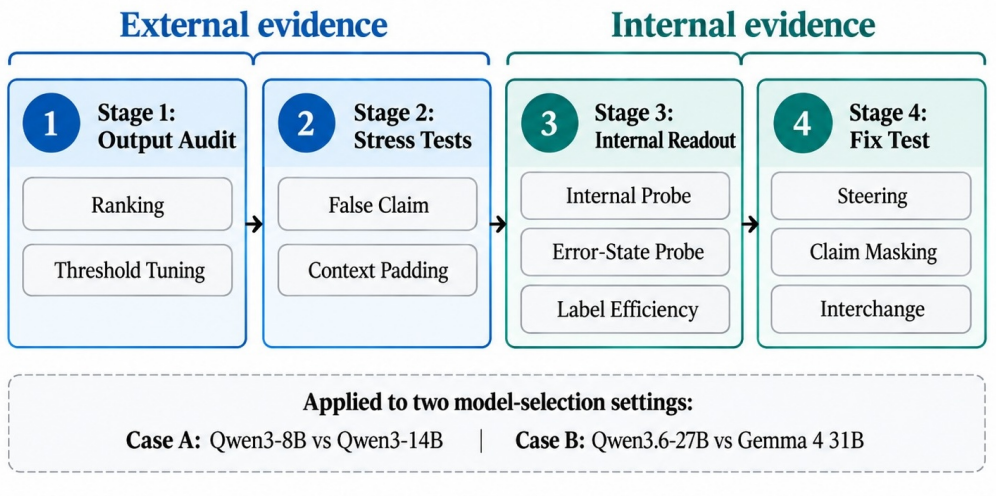
Ranking, calibration, and decision thresholds. Classifier evaluation has long distinguished the quality of a scoring function from the decision rule applied to its scores. ROC/AUC measures discrimination independently of a particular cutoff, while threshold-choice methods determine how continuous scores become binary decisions (Fawcett, 2006; Hernández-Orallo et al., 2012). This is related to, but distinct from, probabilistic calibration methods such as temperature scaling, which aim to align confidence scores with empirical probabilities (Guo et al., 2017). Our setting brings the ranking–threshold distinction into LLM judge selection: a PASS/FAIL accuracy gap may reflect threshold placement rather than weaker ordering of successful and failed trajectories. We therefore use held-out decision-threshold tuning diagnostically, rather than proposing a new calibration method, to test whether a candidate-model comparison survives adjustment of the decision boundary.

Internal evaluation signals and intervention. Language-model representations can contain correctness-relevant information that is not expressed in the final output, and Representation-as-a-Judge applies hidden-state readouts directly to model evaluation (Burns et al., 2023; Orgad et al., 2025; Li et al., 2026). Probing work cautions, however, that decodability does not establish how the model uses that information (Hewitt & Liang, 2019; Elazar et al., 2021). Judge-specific analyses such as Judge Circuits and Inside the Unfair Judge connect internal representations to evaluation behavior, while inference-time intervention, representation engineering, and interchange provide broader tools for manipulating internal states (Feldhus et al., 2026; Xu et al., 2026; Li et al., 2023; Zou et al., 2023; Geiger et al., 2021). We ask whether internally recoverable information changes the comparison between candidate judges and whether manipulating that information selectively corrects actual false-PASS errors rather than merely shifting verdict scores.

3 METHODOLOGY

The Verdict Audit moves from external evidence about judge behavior to internal evidence about what the model represents and whether that information can be used. Figure 2 summarizes the four stages and the two model-selection settings. Throughout, $y_i = 1$ denotes a ground-truth (gold) FAIL item, $y_i = 0$ a gold PASS item, and m_i is the FAIL-oriented verdict margin defined in Section 4.

Figure 2: Overview of the Verdict Audit. Stages 1–2 use external evidence, while Stages 3–4 use internal evidence to test whether useful judgment information is readable and actionable. The same audit is applied to both model-selection settings.



3.1 STAGE 1: OUTPUT AUDIT

1.1 Ranking. We measure threshold-independent ranking with verdict AUC on matched gold-FAIL and gold-PASS checks.

1.2 Threshold Tuning. We fit the decision threshold only on labeled training trajectories,

$$t^* = \arg\max_t \mathrm{BA}_{\mathrm{train}}(t),$$

and evaluate it on held-out trajectories. For Case A, we repeat this at $N \in \{50, 100, 200, 400, \mathrm{all}\}$ labeled checks and use a prespecified ± 0.05 balanced-accuracy equivalence margin for the threshold-tuned Qwen3-8B versus Qwen3-14B comparison.

3.2 STAGE 2: STRESS TESTS

2.1 Claim Injection. For omission items, the required action never occurs. We keep the action history fixed and modify only the agent's closing message using four conditions: a *False Claim* stating that the missing action was completed; a *Negated Claim* stating that it was not completed; a *Generic Claim* using compliance-like language without naming the requirement; and an *Unrelated Control* referring to another requirement.

The primary margin-based quantity is the paired Claim Effect,

$$\Delta_{\mathrm{claim}} = \mathbb{E}\left[m_{\mathrm{False}} - m_{\mathrm{Unrelated}}\right].$$

Because larger margins favor FAIL, negative values indicate movement toward PASS beyond the effect of adding unrelated text. The Negated Claim is a diagnostic control rather than a deployable correction because constructing it requires knowing that the required action is absent.

We evaluate Claim Injection with reasoning both disabled and enabled. Truncated Case B generations are rerun at larger budgets; unresolved truncations are counted as nondetections, with complete-case analyses reported separately.

2.2 Context Padding. We preserve the task, rubric, decisive evidence, and final response while inserting irrelevant trajectory content matched to the format of the original interaction. Each item is evaluated at its original length and at approximately $+0.5\times$ and $+1\times$ additional context, and we compare verdict AUC across doses on the same items.

3.3 Stage 3: Internal Readout

Stage 3 tests readability; Stage 4 tests actionability.

3.1 In-Domain Probe. We extract the model's hidden residual-stream representation at the verdict position and fit ℓ_2-regularized logistic-regression probes using five-fold trajectory-grouped cross-validation. Candidate probe sites are selected using only the training portion of each corresponding outer fold.

To test whether the probe adds information beyond the judge's own verdict margin, we fit

$$\operatorname{logit} P(y_i = 1) = \beta_0 + \beta_m m_i + \beta_z z_i,$$

where z_i is the probe score. The coefficient β_z measures the probe's association with the gold label after accounting for the verdict margin.

3.2 Error-State Probe. Among items the judge predicts PASS, we compare probe and verdict-margin AUC for separating false-PASS from true-PASS cases.

3.3 Label Efficiency. We compare *Threshold Tuning Only* with *Threshold Tuning + Probe* at $N \in \{50, 100, 200, 400, \text{all}\}$ labeled checks. Each finite budget uses 25 fixed-seed draws of complete training trajectories, with every unused trajectory evaluated; the all-data condition uses a trajectory-level train/test split. We report held-out changes in balanced accuracy and AUC.

3.4 Stage 4: Fix Test

We test whether interventions move false-PASS errors toward FAIL more than already-correct true-PASS cases.

4.1 Steering. Using training trajectories, we fit a fail-minus-pass direction $\hat{v}$ and normalize it to unit length. At the verdict-position state,

$$\tilde{h}_i = h_i + \alpha \|h_i\|_2 \hat{v}, \qquad \alpha = 0.20.$$

Each intervention is compared with one equal-norm random-direction control per item. If Δm_g^{dir} and Δm_g^{rand} are the mean margin shifts for false-PASS and true-PASS groups, selectivity is

$$S = \left(\Delta m_{\text{FP}}^{\text{dir}} - \Delta m_{\text{FP}}^{\text{rand}}\right) - \left(\Delta m_{\text{TP}}^{\text{dir}} - \Delta m_{\text{TP}}^{\text{rand}}\right).$$

Positive S indicates greater movement on false-PASS than true-PASS items beyond the random-control effect.

4.2 Direction Geometry. We measure cosine alignment between the fitted fail-minus-pass direction and the FAIL-minus-PASS verdict readout direction, using the result descriptively rather than as a causal explanation.

4.3 Claim Masking. A fixed detector identifies completion claims in the agent's closing message without access to the gold answer. Detected claim text is removed and compared with a matched control that removes a similar amount of non-claim text; we compare balanced accuracy between the two conditions.

4.4 Interchange Control. For each eligible omission, the original run is the *recipient* and the corresponding Negated Claim run is the *donor*. At layer ℓ, we replace the recipient's verdict-position residual state with the donor state and continue the forward pass.

If m_R is the recipient margin, m_D the donor margin, and m_ℓ^{int} the margin after interchange, the recovered fraction of the donor–recipient gap is

$$R_\ell = \frac{m_\ell^{\text{int}} - m_R}{m_D - m_R}.$$

Because the Negated Claim donor is gold-informed, Interchange is diagnostic rather than deployable. The final layer is retained as an implementation sanity check; exact sampled layers are reported in the appendix.

4 Experimental Setup

4.1 Datasets

Our primary benchmark is RuVerBench AgenticCoding (Peng et al., 2026), which contains long coding-agent trajectories paired with binary rubric checks. We use it because it matches the setting studied in this paper: a judge must determine whether a specific requirement was actually satisfied somewhere in a multi-step agent trajectory. Our main analysis uses the same 700 rubric checks for all four judges.

We additionally use AgentRewardBench (ARB) (Lù et al., 2025) for selected Case A experiments. ARB contains human-annotated web-agent trajectories and provides a second agent domain for testing whether findings observed on coding trajectories persist beyond RuVerBench. We do not use ARB as a second target domain for Case B; that comparison is intentionally scoped to the coding-agent setting in which the deployment decision is being made.

4.2 Models

We evaluate four open-weight models: Qwen3-14B, Qwen3-8B, Qwen3.6-27B, and Gemma 4 31B (Yang et al., 2025; Qwen Team, 2026; Gemma Team et al., 2026).

The models were chosen to instantiate the two selection problems introduced in Section 1. Qwen3-14B and Qwen3-8B provide a same-family comparison in which model size can be reduced while holding the model family relatively fixed. Qwen3.6-27B and Gemma 4 31B provide a comparison between two strong open-weight models of similar scale whose general-purpose benchmark profiles do not directly determine which will be the better trajectory judge. All four models also permit access to internal activations, which is required for Stages 3 and 4 of the audit.

4.3 Evaluation Protocol and Judge Readout

Each judgment contains a task, one binary rubric item, and the full agent trajectory. The prompt ends with `Verdict:`, and the judge scores two fixed outputs: `SATISFIED` and `NOT_SATISFIED`. We define the FAIL-oriented verdict margin as

$$m = \log P(\texttt{NOT_SATISFIED}) - \log P(\texttt{SATISFIED}),$$

where each label score aggregates its first-token forms with and without a leading space. Positive margins favor FAIL; for readability, we refer to the two outcomes as PASS and FAIL throughout the paper.

We report verdict AUC for threshold-independent ranking and balanced accuracy for binary decisions. The shipped rule predicts FAIL when $m > 0$; recalibrated thresholds are fit only on labeled training trajectories.

All splits, probe fitting, threshold fitting, and resampling are grouped by trajectory, so rubric checks from the same trajectory never cross training and evaluation sets. Direct model comparisons use matched items whenever possible, with paired uncertainty estimates preserving item correspondence. Stage-specific sample sizes, label budgets, and intervention protocols are described with the corresponding methods.

5 RESULTS

Intervals are 95% trajectory-clustered bootstrap confidence intervals unless noted. Label-budget comparisons in Stage 1 use 90% intervals for the prespecified equivalence test.

5.1 STAGE 1: THRESHOLD TUNING NARROWS CASE A; CASE B RETAINS A RANKING DIFFERENCE

Table 1: Output-level performance on the matched RuVerBench evaluation. Threshold-tuned balanced accuracy uses a decision threshold fit only on labeled training trajectories.

Case	Judge	Verdict AUC	Default bal. acc.	Threshold-tuned bal. acc.
A	Qwen3-8B	0.702	0.535	0.629
	Qwen3-14B	0.677	0.645	0.635
B	Qwen3.6-27B	0.676	0.629	0.629
	Gemma 4 31B	0.771	0.696	0.700

Case A. At the default threshold, Qwen3-8B trails Qwen3-14B in balanced accuracy by 0.110 and predicts FAIL on only 1.9% of checks. The ranking comparison does not show an advantage for the larger judge: the paired 8B–14B AUC difference is $+0.026$ $[-0.028, +0.078]$. After threshold tuning, the balanced-accuracy gap shrinks to 0.006. At $N = 50$, the paired 14B–8B difference is -0.014 [90% CI: -0.076, $+0.051$] (negative values favor Qwen3-8B). The prespecified ± 0.05 equivalence test is not met at any budget, so the absence of a detected difference should not be interpreted as equivalence.

Case B. Qwen3.6 and Gemma 4 already predict FAIL at similar rates under the default threshold (17.9% and 19.4%), making the Case A-style threshold imbalance less plausible. Gemma 4 retains a ranking advantage, with a paired AUC difference of $+0.095$ $[+0.029, +0.162]$. The threshold-tuned balanced-accuracy difference, $+0.071$ $[-0.008, +0.138]$, is not established.

5.2 STAGE 2: STRESS TESTS REVEAL DIFFERENT FAILURE PROFILES

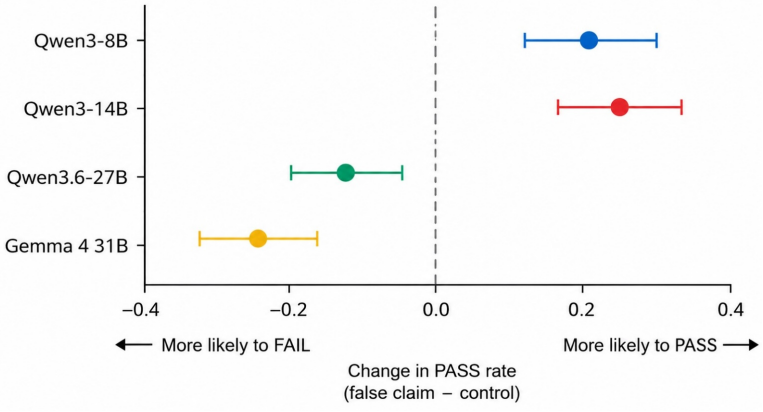
False completion claims. We test whether a judge changes its verdict when the agent falsely states that an omitted action was completed. On the shared 73-item cohort, both Qwen3 judges move toward PASS, whereas Qwen3.6 and Gemma 4 move toward FAIL (Figure 3; margin-based Claim Effects in Appendix A.2.1). For Qwen3-14B, the effect is also visible in binary decisions: on 84 omissions, detection falls from 63% to 38% under the False Claim, while the Unrelated Control leaves it at 61%.

The relative vulnerability of the two Case A judges reverses across domains. The paired 14B–8B Claim-Effect difference is $+1.15$ $[+0.47, +1.85]$ on RuVerBench, where 8B is more strongly misled, but -0.28 $[-0.47, -0.10]$ on AgentRewardBench, where 14B is more strongly misled. Robustness rankings therefore do not automatically transfer across domains.

Reasoning changes the effect, but not uniformly. Reasoning reduces the false-claim vulnerability for Qwen3-14B: its False Claim minus Unrelated Control detection gap narrows by $+0.14$ $[+0.02, +0.24]$. Qwen3-8B remains vulnerable with reasoning enabled, with a detection gap of -0.16 $[-0.26, -0.07]$. Its small reasoning-off detection gap reflects a floor effect because the default threshold rarely predicts FAIL; its margin-based Claim Effect still shows movement toward PASS.

For Qwen3.6, reasoning removes the detectable shift toward FAIL. Gemma 4 remains shifted toward FAIL, but we do not detect a reasoning-induced change in that effect. Case B required larger reasoning budgets; unresolved truncations are counted conservatively, and analyses restricted to completed generations agree in direction.

Figure 3: Effect of false completion claims on binary judge decisions. Each point is the change in PASS rate between the False Claim and Unrelated Control conditions. Positive values mean the false claim made the judge more likely to PASS the omitted requirement; negative values mean more likely to FAIL. Error bars show 95% trajectory-clustered bootstrap confidence intervals.



Context padding. Adding irrelevant context also produces model-specific behavior: at $+0.5\times$ context, Qwen3-14B's AUC decreases (-0.075) while Qwen3-8B's increases ($+0.080$), and neither Case B judge changes detectably. Case A values are reported in Appendix A.2.5.

5.3 Stage 3: Internal Readout Is Most Informative for Qwen3.6

Qwen3.6 shows the largest gap between internal and output-level information: probe AUC is 0.813 versus verdict AUC 0.676, with the strongest conditional probe association among the four judges.

This advantage persists specifically on predicted-PASS items, where Qwen3.6 is the only judge with an established probe-over-margin advantage.

Internal information does not automatically improve binary decisions. At $N = 200$, adding the Qwen3.6 probe improves AUC by $+0.112$ $[+0.018, +0.212]$, while the balanced-accuracy change of $+0.073$ $[-0.050, +0.184]$ is not established. The design can detect balanced-accuracy gains of roughly 0.15–0.23, so smaller benefits remain unresolved. On the same held-out items, Qwen3.6 with the probe differs from Gemma 4 by $+0.018$ AUC $[-0.064, +0.104]$, removing the detectable AUC gap without establishing superiority or equivalence.

Table 2: Internal information available beyond the final verdict. Probe AUC measures information decoded from the verdict-position hidden state. The error-state result asks whether the probe distinguishes false-PASS from true-PASS items better than the judge's own verdict margin. No tested judge shows an established balanced-accuracy gain from adding the probe at any label budget.

Judge	Verdict AUC	Probe AUC	Error-state result
Qwen3-14B	0.677	0.704	Not established
Qwen3-8B	0.702	0.691	Not established
Qwen3.6-27B	0.676	0.813	$+0.194$ $[+0.091, +0.295]$
Gemma 4 31B	0.771	0.806	Not established

5.4 Stage 4: Simple Interventions Do Not Selectively Correct Errors

Steering and claim masking. No judge shows established selectivity under the tested steering intervention. Qwen3-8B illustrates why movement alone is insufficient: its verdict margin shifts by roughly 2 units on both correct and incorrect items. Claim Masking also shows no established improvement over its matched control; the detector covers only 44 of 117 gold failures (38%), limiting the intervention's possible effect.

Interchange shows that richer internal states can move the verdict. Median recovery first exceeds 50% at approximately 0.67 of depth for Qwen3-14B, 0.63 for Qwen3-8B, and 0.69 for Gemma 4. Qwen3.6 behaves differently: it does not cross 50% before the final layer in 98.8% of bootstrap resamples and plateaus around 0.33–0.36 recovery. Because the donor run is explicitly told the answer, Interchange remains a diagnostic rather than a practical correction. Qwen3.6 also does not meet the preregistered pre-final recovery criterion, so the planned learned-subspace follow-up was not run.

Table 3: How each stage changes the two model-selection comparisons.

Audit stage	**Case A: Qwen3-8B vs. Qwen3-14B**	**Case B: Qwen3.6-27B vs. Gemma 4 31B**
Output audit	Large raw gap mostly disappears after threshold tuning; equivalence is not established.	Gemma 4 retains a detectable ranking advantage.
Stress tests	Qwen3-8B remains vulnerable to false claims with reasoning; Qwen3-14B's vulnerability decreases.	Neither judge shows the Case-A-style movement toward PASS.
Internal readout	No comparable internal rescue for Qwen3-8B.	Qwen3.6 contains additional error-relevant information.
Fix test	No tested intervention selectively corrects errors.	Internal information does not translate into demonstrated selective correction.

6 Discussion

Judge quality is not a single property. Ranking, threshold placement, robustness, internal information, and controllability can disagree, and those disagreements can change a model-selection conclusion. Case A makes this clear: the large apparent disadvantage of Qwen3-8B mostly disappears after threshold tuning, while stress testing reveals a robustness difference when false completion claims are introduced. The same pair can therefore look similar on tuned aggregate performance yet differ under plausible failure conditions.

Case B reveals a different mismatch between observable and internal judgment quality. Gemma 4 produces the stronger target-domain ranking directly, while Qwen3.6 contains error-relevant information that its verdict does not fully expose. Recovering that information requires labeled data, activation access, and additional inference machinery, without producing a demonstrated practical advantage over Gemma 4. More broadly, the internal experiments suggest a useful ladder: information may be *present*, *readable*, *causally influential*, and *selectively controllable*, and these are not the same. Probes establish readability, Interchange shows causal influence, and the tested steering intervention does not show selective correction. Decodability and causal influence therefore do not by themselves imply usable control.

This supports a progressive approach to judge selection. Start with the cheapest evidence that can resolve the comparison: output ranking and threshold tuning first, then targeted robustness tests when failure conditions matter. Internal analysis is most useful when observable evidence leaves an unresolved question, and its added complexity should be justified by whether it changes the comparison or enables a usable correction. Judge selection is therefore better viewed as distinguishing several properties of judgment quality rather than reducing them to a single score.

7 Limitations

Our conclusions are scoped to the evaluated judges and domains. RuVerBench AgenticCoding is the primary target domain, with AgentRewardBench used for selected Case A stress tests. The reversal in false-claim vulnerability across those two domains shows directly that robustness rankings need not transfer, so the same comparisons should be re-audited before being generalized to other domains or model families.

Some conclusions are also limited by statistical precision. In Case A, threshold tuning removes the detectable performance gap, but the prespecified equivalence test is not met, so we do not conclude that Qwen3-8B and Qwen3-14B are equivalent. In Stage 3, smaller balanced-accuracy gains may remain unresolved by the available sample sizes; throughout the paper, an interval spanning zero is therefore treated as no detectable difference rather than evidence of equality.

Our internal-intervention conclusions apply only to the tested readouts and intervention strategies. Other representation sites, nonlinear readouts, or more complex interventions remain outside the scope of this study. The preregistered learned-subspace follow-up for Qwen3.6 was not run because the pre-final Interchange stopping criterion was not met, so that intervention remains unresolved; Interchange also relies on gold-informed donor states. Finally, Reasoning-On comparisons partly reflect model-dependent generation length and truncation, with complete-case analyses reported separately to distinguish this from the behavior of completed generations.

8 Conclusion

We introduced the Verdict Audit to separate sources of apparent judge differences that a single benchmark score can conflate. The two case studies show why that distinction matters. In Case A, recalibration removes most of Qwen3-8B's apparent performance disadvantage relative to Qwen3-14B, while stress testing reveals a robustness tradeoff that aggregate accuracy misses. In Case B, Qwen3.6 contains useful judgment information that its verdict fails to expose, but recovering that signal does not produce an established decision-level advantage over Gemma 4 or a selective correction of its errors.

The broader lesson is that judge quality is not one scalar property: calibration, robustness, internal information, and practical usability can disagree. What appears to be a model-quality gap may instead be a threshold problem, a robustness difference, or information that exists internally but cannot yet be used reliably. Judge selection is therefore better treated as an audit problem than a leaderboard problem.

AI use statement

Generative AI tools were used to provide feedback on research methodology and experimental presentation, assist with the interpretation and organization of results, draft and edit manuscript text, create or revise scientific figures, identify and summarize relevant literature, and help format and check references. The authors reviewed all AI-assisted text, figures, claims, and citations against the underlying experimental outputs and primary sources and made the final scientific and editorial decisions. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

This work studies the reliability of LLM-based evaluators for agent trajectories. We use existing evaluation datasets and open-weight language models and do not collect new human-subject data or personal information. The controlled modifications used in our stress tests are designed to characterize evaluator failure modes rather than to facilitate harmful behavior. We report both successful and unsuccessful interventions and avoid interpreting internally decodable information as evidence of practical control when such control is not demonstrated.

Reproducibility statement

We provide the experimental setup, evaluation protocol, model and dataset specifications, decision metrics, stress-test construction, probing procedures, intervention methods, and statistical analyses in Sections 3–5 and the appendix. Data splits are grouped by trajectory, and the paper reports the label budgets, controls, sample sizes, uncertainty estimates, and other stage-specific implementation details needed to reproduce the comparisons. The appendix provides additional experimental details and results supporting the analyses in the main paper.

REFERENCES

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010.

Nils Feldhus, Tanja Baeumel, Elena Golimblevskaia, Qianli Wang, Van Bach Nguyen, Aaron Louis Eidt, Selin Kahvecioglu, Christopher Ebert, Wojciech Samek, Jing Yang, Vera Schmitt, Sebastian Möller, and Simon Ostermann. Judge circuits. *arXiv preprint arXiv:2605.16023*, 2026. URL https://arxiv.org/abs/2605.16023.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Gemma Team et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 2017.

José Hernández-Orallo, Peter Flach, and Cèsar Ferri. A unified view of performance metrics: Translating threshold choice into expected classification loss. *Journal of Machine Learning Research*, 13:2813–2869, 2012.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2733–2743. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1275.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

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.

Sung Won Kim and Daniel Khashabi. Challenging the evaluator: LLM sycophancy under user rebuttal. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22461–22478. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1222.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhuochun Li, Yong Zhang, Ming Li, Yuelyu Ji, Yiming Zeng, Ning Cheng, Yun Zhu, Yanmeng Wang, Shaojun Wang, Jing Xiao, and Daqing He. Rethinking LLM-as-a-judge: Representation-as-a-judge with small language models via semantic capacity asymmetry. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VAISvCsrvG.

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*, pp. 2511–2522. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. AgentRewardBench: Evaluating automatic evaluations of web agent trajectories. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=fQcUZMPIvu.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations*, 2025.

Yangda Peng, Yunjia Qi, Hao Peng, Haotian Xia, Guanzhong He, Xintong Shi, Richeng Xuan, Songyuanyi Lu, Yixian Liu, Zhichao Hu, Yuhong Liu, Lei Hou, Bin Xu, and Juanzi Li. Can LLM-as-a-judge reliably verify rubrics in agentic scenarios? *arXiv preprint arXiv:2606.29920*, 2026. URL https://arxiv.org/abs/2606.29920.

Sashank Pisupati, Henry Broomfield, Eujeong Choi, Antonia Calvi, Charlie Wang, Roman Engeler, Max Bartolo, and Patrick Lewis. Counsel: A meta-evaluation dataset for agentic tasks. *arXiv preprint arXiv:2606.21627*, 2026. URL https://arxiv.org/abs/2606.21627.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27b dense model, April 2026. URL https://qwen.ai/blog?id=qwen3.6-27b.

Vyas Raina, Adian Liusie, and Mark Gales. Is LLM-as-a-judge robust? investigating universal adversarial attacks on zero-shot LLM assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7499–7517. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.427.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based prompt injection attack to LLM-as-a-judge. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 660–674, 2024. doi: 10.1145/3658644.3690291.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-preference bias in LLM-as-a-judge. In *NeurIPS 2024 Workshop on Safe Generative AI*, 2024.

Zixiang Xu, Sixian Li, Huaxing Liu, Xiang Wang, Shuai Li, Zirui Song, and Xiuying Chen. Inside the unfair judge: A mechanistic interpretability account of LLM-as-judge bias. *arXiv preprint arXiv:2607.11871*, 2026. URL https://arxiv.org/abs/2607.11871.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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, 2023.

Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. Cheating automatic LLM benchmarks: Null models achieve high win rates. In *International Conference on Learning Representations*, 2025.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Additional Experimental Results

Unless noted otherwise, intervals are 95% trajectory-clustered bootstrap confidence intervals. Comparisons used for the prespecified Case A equivalence test use 90% intervals.

A.1 Threshold Tuning Across Label Budgets

The Case A conclusion could depend on how much labeled data is available for threshold tuning. We therefore repeat the comparison at $N \in \{50, 100, 200, 400, \text{all}\}$ labeled checks, fitting the decision threshold only on training trajectories.

At $N = 50$, the paired threshold-tuned balanced-accuracy difference (Qwen3-14B minus Qwen3-8B) is -0.014 [90% CI: -0.076, $+0.051$]. No detectable difference is found at the other tested budgets. The prespecified ± 0.05 equivalence criterion is also not met at any budget. The Case A interpretation therefore does not depend on a single label budget: threshold tuning removes the large default-threshold gap without establishing equivalence.

A.2 Stress-Test Analyses

A.2.1 False Completion Claims

The primary Claim Injection comparison asks whether falsely stating that an omitted action was completed changes the verdict beyond the effect of adding unrelated text. Table 4 reports this False Claim minus Unrelated Control effect on the shared 73-item RuVerBench omission cohort.

Because the verdict margin is oriented toward FAIL, negative values indicate movement toward PASS and positive values indicate movement toward FAIL.

The two Qwen3 judges move toward PASS, whereas Qwen3.6 and Gemma 4 move toward FAIL. For Qwen3-14B, the effect is also visible in binary decisions: omission detection falls from 63% to 38% under the False Claim, while the Unrelated Control leaves detection at 61%.

The Negated Claim is used as a diagnostic control rather than a proposed correction. Constructing it requires knowing that the required action was not completed.

A.2.2 Reasoning Changes False-Claim Sensitivity

Reasoning changes the false-claim effect differently across judges. For Qwen3-14B, the False Claim minus Unrelated Control detection gap narrows from -0.22 [-0.32, -0.12] with reasoning disabled to -0.08 [-0.18, $+0.01$] with reasoning enabled. The corresponding change is $+0.14$ [$+0.02$, $+0.24$].

Qwen3-8B remains vulnerable with reasoning enabled. Its detection gap is -0.16 [-0.26, -0.07], and its margin-based Claim Effect remains negative at -2.65 [-4.41, -1.01]. Its reasoning-off binary comparison is affected by a floor effect because the default threshold rarely predicts FAIL; the reasoning-off margin effect remains strongly negative.

For Qwen3.6, the detection gap changes from $+0.11$ [$+0.01$, $+0.21$] with reasoning disabled to 0.00 [-0.07, $+0.07$] with reasoning enabled. Gemma 4 has a Reasoning-On detection gap of $+0.07$ [0.00, $+0.14$]. Its change from Reasoning Off to Reasoning On is -0.08 [-0.21, $+0.04$], so a reasoning-induced change is not established.

Table 4: Effect of a false completion claim on the FAIL-oriented verdict margin with reasoning disabled.

Judge	Claim Effect (logits)
Qwen3-14B	-1.94 [-2.54, -1.32]
Qwen3-8B	-3.09 [-3.72, -2.39]
Qwen3.6-27B	$+1.39$ [$+0.75$, $+2.04$]
Gemma 4 31B	$+1.62$ [$+0.77$, $+2.50$]

A.2.3 Reasoning-Length and Truncation Checks

The Case B judges require longer reasoning generations than the two Qwen3 judges. Table 5 reports unfinished generation rates under a common 1,024-token budget and after rerunning truncated Case B generations with larger budgets.

Qwen3.6 truncated generations are rerun at 2,048 tokens, while Gemma 4 is rerun with budgets up to 4,096 tokens. Analyses restricted to completed generations agree with the primary results in direction. The reasoning results therefore include the operational cost of obtaining the reasoning trace while not depending solely on how unfinished generations are scored.

A.2.4 Cross-Domain False-Claim Sensitivity

We test whether the relative false-claim sensitivity of the two Case A judges transfers from coding trajectories to web-agent trajectories. Table 6 reports the paired difference between their Claim Effects.

On RuVerBench, the positive difference reflects a stronger movement toward PASS for Qwen3-8B. On AgentRewardBench, the ordering reverses and Qwen3-14B is more strongly affected. The comparison therefore supports evaluating robustness in the deployment domain rather than assuming that a pairwise robustness ordering transfers unchanged.

A.2.5 Context Padding

Context Padding changes only irrelevant trajectory content while preserving the task, rubric, decisive evidence, and final response. Table 7 gives the numerical Case A result at the $+0.5\times$ dose highlighted by the main analysis.

The two Case A judges move in opposite directions under the same padding intervention. Neither Case B judge shows a detectable change under the tested padding conditions. This provides a second example, independent of false completion claims, in which robustness behavior is not summarized by aggregate output performance alone.

A.3 Internal Readout Analyses

A.3.1 Information Beyond the Verdict

The main-text probe comparison shows the largest discrepancy between internal and output-level information for Qwen3.6. Its verdict AUC is 0.676, while the internal probe reaches 0.813.

We additionally test whether the probe remains informative after accounting for the judge's own verdict margin. For Qwen3.6, the conditional probe coefficient is $+0.403$ [$+0.310$, $+0.485$], the largest conditional association among the four tested judges. This supports the interpretation that the Qwen3.6 representation contains information not already captured by its observable verdict score.

A.3.2 Error-State Readout

We next restrict the analysis to items that the judge predicts PASS. Within this subset, the question is whether the internal readout can distinguish false-PASS errors from true-PASS decisions when the observable verdict is already the same.

Table 5: Reasoning-generation truncation rates. Remaining unfinished generations are counted as omission nondetections in the primary binary analysis.

Judge	Unfinished at 1,024 tokens	After rerun
Qwen3-14B	1.4%	Not required
Qwen3-8B	1.4%	Not required
Qwen3.6-27B	17.0%	5.0%
Gemma 4 31B	29.0%	8.2%

Table 6: Cross-domain comparison of Case A false-claim sensitivity. The reported quantity is the Qwen3-14B minus Qwen3-8B Claim-Effect difference.

Domain	n	Paired Claim-Effect difference
RuVerBench AgenticCoding	73	$+1.15$ [$+0.47$, $+1.85$]
AgentRewardBench	200	-0.28 [-0.47, -0.10]

For Qwen3.6, the verdict margin reaches AUC 0.567 on this comparison, while the internal probe reaches 0.761. The paired probe-minus-margin difference is $+0.194$ [$+0.091$, $+0.295$]. The corresponding differences for the other three judges all include zero.

The Qwen3.6 result therefore reflects information that remains available specifically on errors, rather than only reproducing information already expressed by the final verdict.

A.3.3 Label Efficiency

To test whether the Qwen3.6 internal signal translates into better held-out evaluation, we compare Threshold Tuning Only with Threshold Tuning + Probe across $N \in \{50, 100, 200, 400, \text{all}\}$ labeled checks.

At $N = 200$, adding the probe improves AUC by $+0.112$ [$+0.018$, $+0.212$]. The corresponding balanced-accuracy change is $+0.073$ [-0.050, $+0.184$] and is not established.

No tested label budget establishes a balanced-accuracy gain. The experimental design can detect gains of approximately 0.15–0.23, depending on the budget, so smaller decision-level improvements remain unresolved rather than ruled out.

On the same held-out items, Qwen3.6 with the probe differs from Gemma 4 by $+0.018$ AUC [-0.064, $+0.104$]. Internal readout therefore removes the detectable output-level AUC gap without establishing either superiority or equivalence.

A.4 Intervention Analyses

A.4.1 Steering Selectivity

The steering analysis asks whether a fitted fail-minus-pass direction moves false-PASS errors toward FAIL more strongly than already-correct true-PASS cases, after accounting for an equal-norm random-direction control.

No tested judge shows established selectivity under this intervention. Qwen3-8B illustrates why overall movement is not sufficient: the intervention shifts the verdict margin by roughly two logits on both false-PASS and true-PASS examples. The intervention therefore changes the verdict score without selectively correcting the errors that motivate it.

A.4.2 Claim Masking

Claim Masking tests whether removing detected completion claims from the agent's closing response improves evaluation relative to removing a similar amount of non-claim text.

Claim Masking does not establish an improvement over the matched control. Its 38% coverage also limits the fraction of failures on which the intervention can act.

Table 7: Selected context-padding result on the matched Case A evaluation cohort.

Judge	Original AUC	$+0.5\times$ AUC	Change
Qwen3-14B	0.763	0.688	-0.075
Qwen3-8B	0.674	0.754	$+0.080$

Table 8: Coverage of the Claim Masking analysis on the gold-failure set.

Quantity	Value
Gold failures	117
Completion claims detected	44
Detection coverage	38%
Items admitting a length-matched control	39

A.4.3 Interchange Recovery

Interchange tests whether replacing the recipient's internal state with the corresponding state from a Negated Claim donor can causally move the downstream verdict. Because the donor explicitly states that the required action was not completed, this experiment is diagnostic rather than deployable.

For Qwen3-8B, Qwen3-14B, and Gemma 4, median recovery exceeds 50% before the final layer. Qwen3.6 behaves differently: it fails to cross 50% before the final layer in 98.8% of bootstrap resamples and pre-final recovery plateaus around 0.33–0.36.

The final-layer intervention is retained only as an implementation sanity check and is not interpreted as evidence of a deployable correction.

A.4.4 Learned-Subspace Stopping Rule

A learned-subspace intervention was preregistered as a follow-up for Qwen3.6, the judge showing the clearest additional internal signal. Proceeding required sufficient pre-final recovery under full-state Interchange. Because that criterion was not met, the learned-subspace follow-up was not run.

The result therefore remains unresolved rather than negative: the current experiments show that the tested full-state and steering interventions do not provide a selective practical correction, while more complex internal interventions remain outside the scope of this study.

B Additional Experimental Details

B.1 Evaluation Cohorts

Different analyses use different subsets because they require particular omission types, matched interventions, or available controls. Table 10 summarizes the principal evaluation cohorts.

B.2 Trajectory-Level Splitting and Pairing

Training and evaluation are separated at the trajectory level throughout the audit. Rubric checks originating from the same trajectory therefore do not appear on both sides of a train/test split.

Threshold fitting, probe fitting, and label-efficiency experiments use the same trajectory-level separation. Direct judge comparisons use matched items whenever possible so that uncertainty estimates preserve the correspondence between the two judges' evaluations of the same underlying examples.

B.3 Uncertainty Estimation

Unless otherwise stated, uncertainty intervals are obtained with trajectory-clustered bootstrap resampling. Clustering preserves dependence among rubric checks originating from the same agent trajectory.

Table 9: Normalized model depth at which median Interchange recovery first exceeds 50% of the donor–recipient verdict-margin gap.

Judge	First $> 50\%$ median recovery
Qwen3-8B	0.63
Qwen3-14B	0.67
Gemma 4 31B	0.69
Qwen3.6-27B	No pre-final crossing

Table 10: Principal evaluation cohorts used in the analyses.

Analysis	Evaluation cohort
Main RuVerBench comparison	700 rubric checks per judge
Shared cross-judge Claim Injection	73 omission items
Qwen3-14B binary false-claim analysis	84 omission items
AgentRewardBench cross-domain analysis	200 items
Claim Masking coverage analysis	117 gold-failure items
Claim Masking matched-control analysis	39 detected-claim items

Paired model comparisons resample the matched evaluation units jointly. The Case A label-budget analysis uses 90% intervals because it is tied to the prespecified ± 0.05 equivalence analysis; other reported intervals use 95% confidence intervals.

B.4 Probe Training and Site Selection

Internal probes are ℓ_2-regularized logistic regressions trained on the residual-stream representation at the verdict position. Evaluation uses five-fold trajectory-grouped cross-validation.

When more than one candidate representation site is considered, site selection is performed using only the training portion of the corresponding outer fold. The held-out fold is therefore not used to choose the representation site.

For label-efficiency experiments, finite training budgets use 25 fixed-seed draws of complete training trajectories. All unused trajectories are evaluated. The all-data condition instead uses a trajectory-level train/test split.

B.5 Claim-Injection Controls

Claim Injection preserves the underlying action history and modifies only the agent's closing message. The False Claim states that the omitted requirement was completed. The Unrelated Control adds a statement about a different requirement and is used as the primary control for the additional text.

The Generic Claim uses compliance-like language without naming the missing requirement. The Negated Claim states that the action was not completed. The latter is used diagnostically, including as the donor in the Interchange experiment, because constructing it requires knowledge of the omission.

B.6 Claim-Masking Control

The Claim Masking detector operates without access to the gold outcome. It identifies completion assertions in the closing response using rubric overlap and predefined completion-assertion patterns.

For each detected claim, the intervention removes the claim text. Where possible, the matched control removes a similar amount of non-claim text from the same response. Comparing these two conditions helps distinguish removal of completion assertions from the generic effect of deleting text.

B.7 Reasoning Budgets

Reasoning-On evaluations initially use a common 1,024-token budget. Truncated Qwen3.6 generations are rerun at 2,048 tokens, while truncated Gemma 4 generations are rerun with budgets up to 4,096 tokens.

Any generations that remain unfinished after rerunning are counted as omission nondetections in the primary binary analysis. Completed generations are also analyzed separately as a sensitivity check.

B.8 Interchange Interpretation

For each eligible omission, the original run serves as the recipient and the corresponding Negated Claim run serves as the donor. At a tested depth, the recipient's verdict-position residual state is replaced by the donor state and the forward computation then continues normally.

Recovery measures how much of the donor–recipient verdict-margin gap is restored after the replacement. Intermediate depths test whether the donor information survives the model's remaining computation. The final layer is used only as an implementation sanity check.

Because the donor is gold-informed, Interchange is used to test causal influence of a richer internal state rather than as a proposed deployment-time correction.

C Appendix