WHEN DOES CROSS-EXAMINATION BEAT SELF-ASSESSMENT? A TEXT-SURFACE VISIBILITY LAW FOR TRAINING-FREE AGENT TRUSTWORTHINESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting when an LLM agent is about to fail—without ground truth or extra training—is central to deploying agents safely. Two training-free signals compete: *self-assessment* (the acting model reports its own confidence) and *cross-examination* (a separate model audits the trajectory). We ask when one beats the other, and find that neither wins universally. Across five agentic settings spanning conversational tool use (τ-bench retail/airline), step-level failure attribution (Who&When), and code generation (BigCodeBench), we establish a **monotone visibility law**: the net benefit of cross-examination over self-assessment is determined *not* by whether the task has an objective notion of correctness, but by whether the error signal is exposed at the *text surface* readable by the auditor. Ordering domains by a semi-quantitative visibility rubric yields a strictly monotone gain curve, from a significant *negative* gain in low-visibility conversational trajectories (-0.156, $P{=}.02$) through a near-zero point in code—a high-objectivity but low-visibility counter-example ($+0.024$, ns)—up to a significant *positive* gain in action-rich retail dialogues ($+0.305$, $P{=}1.0$). We further show that naive multi-judge aggregation does *not* beat the strongest single judge, and that inter-judge agreement itself tracks visibility. Finally, we give a first partial solution: a deploy-time observable proxy (trajectory depth) routes each domain to the better signal, reaching 97% of an oracle's within-domain AUROC and significantly outperforming always-self ($P{=}.999$). All findings are training-free and pure-API; code and data are released for reproducibility.

1 INTRODUCTION

As LLM agents are deployed in high-stakes, multi-step tasks—interleaving reasoning with tool calls (Yao et al., 2023b; Schick et al., 2023) across long horizons (Liu et al., 2024; Mialon et al., 2024)—a practical question recurs: *can we tell, without ground truth and without extra training, when the agent is about to fail?* A reliable trustworthiness signal would let a system abstain, escalate to a human, or re-plan before an error propagates. Two training-free signals are natural candidates. **Self-assessment** asks the acting model to report its own confidence in a trajectory, building on evidence that models are partly calibrated on their own answers (Kadavath et al., 2022; Tian et al., 2023). **Cross-examination** instead has a *separate* model audit the trajectory—a scalable, black-box use of the LLM-as-a-judge paradigm (Zheng et al., 2023; Gu et al., 2024)—in the spirit of external critique that intrinsic self-correction alone cannot provide (Huang et al., 2024a; Valmeekam et al., 2023).

Self-assessment is cheap but suffers from over-confidence (Xiong et al., 2024), and models are known to struggle to correct their own reasoning without external feedback (Huang et al., 2024a). Cross-examination brings an independent perspective and underlies the gains reported for multi-agent debate (Du et al., 2024a;b) and verbal self-reflection with external signals (Shinn et al., 2023). Yet, as we show, *neither signal dominates*: the better choice flips across settings, and picking the wrong one can be worse than doing nothing. A tempting explanation is that auditing helps precisely

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.

when correctness is *objective*. We show this intuition is wrong, and identify the axis that actually governs the outcome.

We conduct a controlled, training-free, pure-API study across five agentic settings and uncover a single organizing principle. The net benefit of cross-examination over self-assessment is governed *not* by whether the task admits an objective notion of correctness, but by whether the *error signal is exposed at the text surface* that the auditor can read. We formalize this as a semi-quantitative *visibility* rubric and show that ordering domains by visibility produces a **strictly monotone** curve of cross-examination gain (Figure 1): from a *significant negative* gain in shallow conversational trajectories, through a near-zero point in code generation—a high-objectivity but low-visibility *counter-example*—up to a *significant positive* gain in action-rich dialogues. Code is decisive: its correctness is fully objective (unit tests (Chen et al., 2021; Zhuo et al., 2025)), yet the failure cause hides in execution semantics that an auditor cannot see by reading source, so cross-examination provides no gain. This is consistent with the observation that a visible reasoning trace need not faithfully reflect the underlying computation (Lanham et al., 2023). Objectivity is not the axis; visibility is.

The law converts a confusing empirical picture into an actionable rule and a cautionary one. The cautionary half is that a fixed policy can be actively harmful: in our lowest-visibility domain, following the auditor is significantly *worse* than trusting the agent's own confidence, so a system that always cross-examines would degrade below doing nothing. The actionable half is that visibility is approximable from a deploy-time observable—trajectory depth—so a system can route each case to the better signal without any labels. Because the entire pipeline is training-free and uses only public inference APIs, the prescription transfers directly to deployed agents.

Contributions.

- **A monotone visibility law** for training-free agent trustworthiness, validated on five settings with a unified normalized-gain metric; the two extremes are statistically significant and code anchors a high-objectivity/low-visibility counter-example (Section 4).
- **Two honest negative results** that sharpen the law: (i) naive multi-judge aggregation (majority/median) does *not* beat the strongest single judge, and (ii) inter-judge agreement itself tracks visibility—evidence that the gain comes from *external* cross-examination rather than "collective disagreement" (Section 5).
- **A first partial solution**: since a deployed system does not know the current domain, we route on a deploy-time observable proxy (trajectory depth) that approximates visibility. Depth routing reaches 97% of an oracle's within-domain AUROC and significantly outperforms always-self (P=.999), avoiding the collapse that either fixed strategy suffers in its worst domain (Section 6).

2 Related Work

Self-assessment, calibration, and confidence elicitation. A large body of work uses a model's own signal to gauge or improve its outputs. Language models are, to a degree, calibrated on their own answers (Kadavath et al., 2022), and calibrated confidence can be elicited by asking the model directly (Tian et al., 2023), evaluated against truthfulness benchmarks (Lin et al., 2022), or by black-box prompting and consistency strategies (Xiong et al., 2024). Chain-of-thought prompting (Wei et al., 2022) and self-consistency over sampled reasoning paths (Wang et al., 2023) and verbalized uncertainty are the common building blocks of the self-assessment signal we study. Yet elicited confidence is frequently over-confident (Xiong et al., 2024), motivating the question of when an external audit should be preferred. Our study treats self-assessment as one competing training-free signal and quantifies exactly *when* it is preferable to cross-examination.

Self-correction and self-verification. Test-time self-improvement methods—iterative self-refinement (Madaan et al., 2023), self-critique from model-generated principles (Bai et al., 2022), verbal self-reflection with feedback (Shinn et al., 2023), deliberate search over thoughts (Yao et al., 2023a), and step-wise self-verification of reasoning chains (Ling et al., 2023)—rely on a model critiquing itself. However, Huang et al. (2024a) and Huang et al. (2024b) show that *intrinsic* self-correction, without any external feedback, often fails to improve and can even degrade reasoning,

and Valmeekam et al. (2023) report the same for self-critiqued planning. Reliable correction typically requires an *external* signal—a tool (Gou et al., 2024), a trained verifier (Lightman et al., 2024; Hosseini et al., 2024), or a search critic (Tian et al., 2024). This motivates our focus on cross-examination by a separate model, and on identifying the conditions under which it actually helps.

External evaluation and cross-examination. Using strong LLMs as judges is now a standard, scalable evaluation paradigm (Zheng et al., 2023; Gu et al., 2024), and multi-round or multi-agent protocols—debate (Du et al., 2024a;b) and referee teams (Chan et al., 2024)—can improve factuality and reasoning. However, LLM judges carry systematic biases such as sensitivity to answer order (Wang et al., 2024), and reward-model ensembles do not eliminate the underlying error patterns (Eisenstein et al., 2024). Consistency-based signals such as SelfCheckGPT (Manakul et al., 2023) likewise show that legibility of the underlying content governs detectability. We repurpose LLM-as-a-judge as a *trustworthiness signal* for another agent's trajectory—no ground truth is given to the judge—and, unlike debate work that assumes aggregation helps, we find that naive aggregation does not beat the strongest single judge, isolating *external* audit (not disagreement) as the source of gain.

Why visibility, not objectivity, may be the axis. A natural hypothesis is that auditing helps whenever correctness is objective. Two lines of evidence suggest otherwise. First, chain-of-thought text is not always a faithful account of the model's computation (Lanham et al., 2023): the visible trace can diverge from the process that produced the answer, so an objective outcome need not be legible at the surface. Second, correctness can be objective—via expert-verified answers (Rein et al., 2024) or unit tests (Chen et al., 2021; Zhuo et al., 2025)—yet in the latter its failure cause lives in execution semantics an auditor cannot read from source. These observations predict that the readability of the error at the text surface, rather than the objectivity of the task, should govern when cross-examination helps—the law we establish empirically.

Failure attribution and agent benchmarks. Zhang et al. (2025) introduce the Who&When dataset for automated failure attribution in multi-agent systems, reporting that the best method reaches only 14.2% step-level accuracy—some methods below random—exposing how hard step-level attribution is. τ-bench (Yao et al., 2024) evaluates tool-agent-user interaction in retail and airline domains with database-state verification, revealing low and inconsistent agent success, and a broader family of agent benchmarks stresses long-horizon, real-world competence spanning software engineering (Jimenez et al., 2024), multi-hop question answering (Yang et al., 2018), and interactive environments (Liu et al., 2024; Mialon et al., 2024; Zhou et al., 2024). These agents typically interleave reasoning and acting (Yao et al., 2023b) and call external tools (Schick et al., 2023; Gao et al., 2023), and modern reasoning models further internalize such deliberation (OpenAI, 2024). We use Who&When, τ-bench, and code generation as sources of real trajectories with per-step or per-trajectory ground truth, spanning objective/subjective correctness and shallow/deep traces. Rather than proposing a new attribution method, we characterize a cross-cutting law that predicts *when* training-free auditing helps at all.

3 Study Design

Signals under test. Given an agent trajectory τ for a task, two training-free, pure-API scores predict failure. **Self-assessment** $s_{\text{self}}(\tau)$ is the acting model's own reported probability that τ is correct, elicited by directly asking the model for a calibrated confidence (Tian et al., 2023; Xiong et al., 2024). **Cross-examination** $s_{\text{judge}}(\tau)$ is an error score produced by a *separate* model that reads τ (without ground truth) and audits it, an instance of the LLM-as-a-judge paradigm (Zheng et al., 2023; Gu et al., 2024). For trajectory-level detection we score whether τ will fail; for step-level attribution the auditor names the first mistaken step, following the protocol of Zhang et al. (2025). No model is fine-tuned; all scores come from public inference APIs, so that any observed effect reflects the signal itself rather than task-specific training.

Domains and ground truth. We span five settings chosen to vary widely in how errors surface: (1–2) *τ-bench retail* and *airline* (Yao et al., 2024), conversational tool-use trajectories with database-state ground truth; (3–4) *Who&When* (Zhang et al., 2025), multi-agent web/code failure trajectories with a gold first-mistake step, split into its *Hand-Crafted* (deep) and *Algorithm-Generated* (shallow)

Table 1: Text-surface visibility rubric: per-domain scores on four task-structural dimensions, each in $\{0, 1, 2\}$ (0 absent, 1 partial, 2 explicit). The equal-weight mean, rescaled to $[0, 1]$, gives the visibility score v used as the horizontal axis of the law. Scores are assigned from task structure alone, before any outcome is measured.

Domain	(a) State exposure	(b) Criterion alignment	(c) Causal localiz.	(d) Effect explicit.	Visibility v
τ-bench airline	0	0	0	0	0.00
Code generation	2	0	0	0	0.25
Who&When (deep)	2	2	0	0	0.50
Who&When (shallow)	2	2	2	1	0.875
τ-bench retail	2	2	2	2	1.00

subsets; (5) *code generation* (BigCodeBench-style (Zhuo et al., 2025)), model solutions with objective unit-test ground truth in the tradition of functional-correctness evaluation (Chen et al., 2021). This spans objective/subjective correctness and shallow/deep trajectories, so that the objectivity of a task and the visibility of its errors can be *dissociated* rather than conflated. The four evaluation slices contain 729 (airline), 1393 (retail), 1517 (code), and 651 (Who&When attribution) scored instances.

The visibility rubric. We define *text-surface error visibility* $v \in [0, 1]$ before inspecting outcome labels, using four task-structural dimensions scored 0 (absent), 1 (partial), or 2 (explicit): (a) *intermediate-state exposure*—whether the trajectory records the intermediate states an error would corrupt; (b) *criterion–text alignment*—whether the correctness criterion is expressed in the same modality as the visible text; (c) *causal localizability*—whether a failure can be traced to a specific readable step; and (d) *effect explicitness*—whether the consequence of an error appears in the trace. Their equal-weight mean gives v. The resulting scores are $0, .25, .50, .875, 1.0$ for airline, code, Who&When deep, Who&When shallow, and retail. Crucially, the dimensions are scored from task structure alone, so the rubric is not fit to the outcome we later measure. The ordering is componentwise nondecreasing and remains strict for all $3^4 = 81$ positive weight vectors in $\{1, 2, 3\}^4$; dropping effect explicitness ties only the top two domains, which identifies the boundary of that distinction. Visibility remains distinct from objective ground truth: code has objective tests but low textual visibility because correctness requires execution, and even a faithful-looking reasoning trace can diverge from the underlying computation (Lanham et al., 2023). Appendix A gives all anchors, assignments, and sensitivity checks.

Unified gain metric. To place all five domains on one axis we normalize the net benefit of cross-examination by the headroom left for improvement. For AUROC domains (trajectory-level detection), $g = (\text{AUROC}_{\text{judge}} - \text{AUROC}_{\text{self}})/(1 - \text{AUROC}_{\text{self}})$; for step-accuracy domains, $g = (\text{acc}_{\text{best-judge}} - \text{acc}_{\text{baseline}})/(1 - \text{acc}_{\text{baseline}})$. Normalizing by headroom makes gains comparable across domains with very different baseline difficulty, so that a large absolute gain in an easy domain and a small one in a hard domain are placed on the same scale. AUROC uses the Mann–Whitney statistic (higher error score $\Rightarrow$ predicted failure); significance for score differences uses a paired bootstrap (n=2000) reporting the 95% CI and $P(\Delta{>}0)$.

Models. Auditors and self-assessors are drawn from four contemporary API models (`claude-opus`, `gpt-5.2`, `glm`, `qwen3-max`); using several models of comparable capability guards against reading a single model's idiosyncrasy as a general law. To rule out a model-generation confound in the step-attribution comparison against Zhang et al. (2025), we additionally reproduce the original method with the *same* contemporary models under the original prompt and scoring, confirming our gains are not merely due to newer models (Section 5).

Protocol and reproducibility. All signals are elicited with fixed, released prompt templates and greedy or low-temperature decoding. Self-assessment asks the acting model, in a separate call that hides the gold outcome, for a calibrated probability that its own trajectory is correct; cross-examination gives a *different* model the same trajectory (again without ground truth) and asks it to score whether the trajectory fails and, for attribution, to name the first mistaken step. To prevent leakage, the auditor never sees the gold label, the reference solution, or unit-test results; it reads

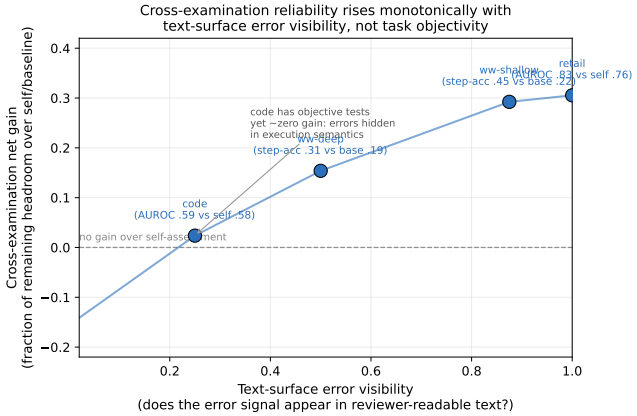

Figure 1: **The monotone visibility law.** x: text-surface error visibility v (rubric, 0–1); y: normalized net gain of cross-examination over self-assessment (fraction of remaining headroom). Five real-data domains are strictly increasing in v. The two extremes are statistically significant (airline P=.02 negative; retail P=1.0 positive); code is a high-objectivity, low-visibility counter-example ($\approx$ 0, ns), showing objectivity is not the governing axis.

Table 2: Per-domain evidence for the visibility law. Gain g is the normalized net benefit of cross-examination; significance from a paired bootstrap (n=2000).

Domain	Visibility v	Net gain g	Signal	Significance
τ-bench airline	0.00	-0.156	AUROC .43 vs self .51	P=.02 (sig. neg.)
Code generation	0.25	$+0.024$	AUROC .59 vs self .58	P=.61 (ns)
Who&When (deep)	0.50	$+0.154$	step-acc .31 vs base .19	lift $+.13$
Who&When (shallow)	0.875	$+0.292$	step-acc .45 vs base .22	lift $+.23$
τ-bench retail	1.00	$+0.305$	AUROC .83 vs self .76	P=1.0 (sig. pos.)

only what a human reviewer of the transcript would see. Each instance is scored independently, and every reported comparison uses the paired bootstrap above over the shared set of instances that both signals cover, so the two strategies are always compared on identical cases. Because some auditors occasionally fail to return a parseable score, we additionally report the panel-coverage sensitivity of every conclusion (Section 5 and Appendix B).

4 The Text-Surface Visibility Law

Ordering the five domains by their visibility score v yields a strictly monotone curve of cross-examination gain g (Figure 1 and Table 2). The relationship is not merely correlational at the extremes: the lowest-visibility domain shows a *significant negative* gain and the highest a *significant positive* gain, with code providing a near-zero anchor in between.

Reading the axis. The horizontal axis is the visibility rubric of Section 3, assigned from task structure before any outcome was measured; the vertical axis is the headroom-normalized gain of an external audit over the agent's own confidence. Monotonicity means that as errors become more readable at the text surface, an outside reader becomes progressively more useful, and below a threshold it becomes actively harmful. The curve therefore does more than rank domains: it predicts the *sign* of the deployment decision. Where v is low the correct policy is to trust the agent; where v is high the correct policy is to trust the auditor; and the crossing point is an empirical property of the trajectories, not of whether the task has a gold answer.

The code counter-example. Code generation is the decisive test of what the axis really is. Its correctness is *fully objective*—unit tests give an unambiguous gold (Chen et al., 2021; Zhuo et al., 2025)—so a task-objectivity hypothesis predicts cross-examination should excel. It does not

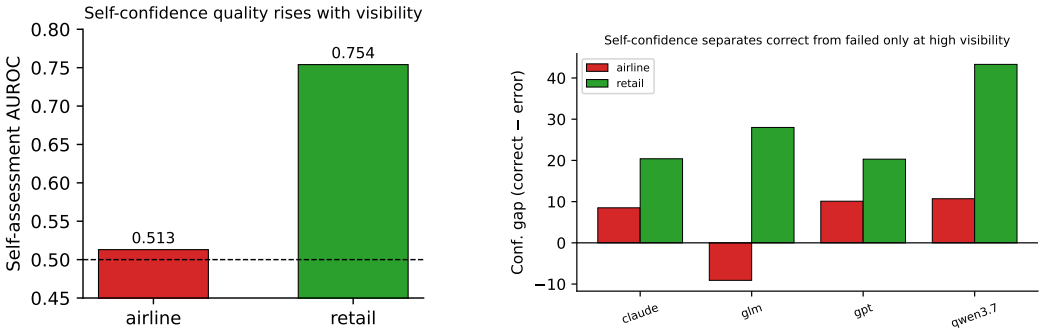

Figure 2: **Self-assessment quality also rises with visibility.** Left: the acting model's own confidence separates correct from failed trajectories far better in high-visibility retail (AUROC 0.754) than in low-visibility airline (0.513, near chance). Right: per-model confidence gap (mean confidence on correct minus failed cases) is small or even negative on airline but consistently positive on retail. Even the *self*-signal degrades when the error is not legible at the text surface, reinforcing that visibility—not objectivity—is the governing axis.

(g=+0.024, ns): auditors reading source cannot see the failure, which lives in execution semantics. Diagnostically, judges assign failing solutions a mean success probability of 76.3 (the weakest judges give 84–85, barely distinguishing pass from fail), and only the judge that reads code most carefully opens a gap (pass 70.1 vs fail 50.8). The failure is not that the judges are weak in general—they separate pass from fail perfectly whenever the bug is textually apparent (e.g. a raised exception printed in the trace)—but that a semantically wrong yet syntactically plausible program looks correct on the page. Objectivity is present but visibility is absent, and gain vanishes—pinning the axis to *text-surface visibility*.

The conversational extremes. At low visibility, τ-bench airline trajectories are so shallow (gold cause $\approx$ one step) that the auditor has almost no legible signal; cross-examination is significantly *worse* than self-assessment (g=−0.156, P=.02), and following it would harm a deployed system. The mechanism is asymmetric error: with little to read, the auditor manufactures plausible-sounding faults, adding noise that the agent's own confidence—which at least reflects its internal state—does not. At high visibility, retail dialogues expose the full action sequence, and cross-examination is significantly *better* (g=+0.305, P=1.0): every tool call and its effect is on the page, so an independent reader catches inconsistencies the acting model rationalizes away.

Visibility also governs the acting model's *own* confidence: self-assessment separates correct from failed trajectories far better in retail than in near-chance airline (Figure 2).

The within-benchmark control. The Who&When pair is the cleanest control for confounds, because its two subsets share the same task family, ground-truth format, and evaluation code, differing essentially only in how plainly the mistake is written into the trace. Its shallow (Algorithm-Generated) subset, where the mistake is stated almost verbatim, yields a larger gain (g=+0.292) than its deep (Hand-Crafted) subset (g=+0.154). Because everything except visibility is held fixed, this ordering cannot be explained by domain, dataset, or objectivity differences, and provides a within-benchmark replication of the cross-domain law.

5 Analysis: What the Signal Is, and Is Not

We report four analyses that sharpen the law and rule out obvious confounds. Two are honest negative results; the third removes a model-generation confound; the fourth checks robustness to incomplete judge panels.

(i) Naive multi-judge aggregation does not beat the best single judge. A natural hypothesis, following the literature on debate and referee teams (Du et al., 2024a; Chan et al., 2024), is that

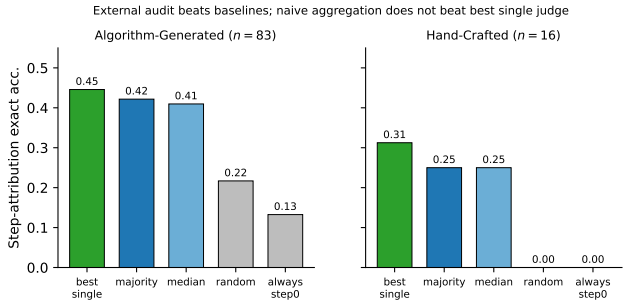

Figure 3: **Step-attribution accuracy: external audit beats baselines, aggregation does not beat the best single judge.** On both Who&When subsets the best single judge and the collective votes far exceed random and always-step-0 baselines, but majority/median aggregation does not significantly exceed the strongest single judge, isolating external audit (not disagreement) as the source of gain.

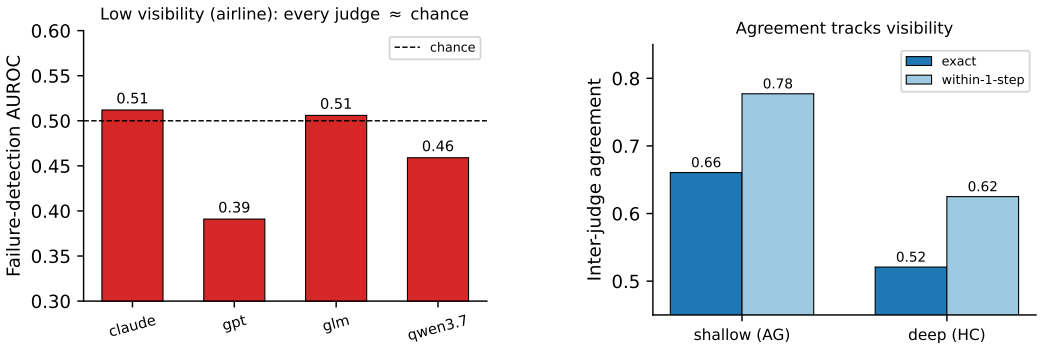

Figure 4: **Left: no single judge helps at low visibility.** Per-judge failure-detection AUROC on τ-bench airline sits at or below chance for all four auditors, so the negative aggregate gain is not one weak judge but an absent signal. **Right: auditors agree more when errors are more visible.** Pairwise exact and within-one-step agreement on Who&When step attribution both rise on the shallow (AG) subset relative to the deep (HC) subset, tracking the same visibility axis.

pooling multiple auditors—"collective disagreement"—should help. It does not. On Who&When step attribution, aggregating four judges by majority or median vote never significantly beats the strongest single judge. On the Algorithm-Generated subset (n=83, best single $= 0.446$), majority vote scores 0.422 (Δ=-0.024, 95% CI $[-0.084, +0.036]$, P=.144) and median 0.410 (Δ=-0.036, P=.086); on Hand-Crafted (n=16, best $= 0.312$), majority scores 0.250 (Δ=-0.062, P=.166). The gain therefore comes from *external* audit itself—a single competent outside reader—rather than from disagreement among many (Figure 3). This mirrors findings that ensembling does not remove shared error patterns (Eisenstein et al., 2024): if the error is legible, one careful reader already sees it; if it is not, more readers cannot manufacture a signal that is absent from the text.

(ii) Inter-judge agreement tracks visibility. If visibility governs how legible the error is, judges should agree *more* when visibility is higher. They do. Pairwise exact agreement among judges is 0.661 on the shallower Algorithm-Generated subset versus 0.521 on the deeper Hand-Crafted subset (within-one-step agreement 0.777 vs 0.625). Agreement rising with visibility—and every individual auditor sitting at or below chance on airline (Figure 4)—is independent evidence that the same latent quantity—how readable the error is at the text surface—drives both the gain and the concordance of auditors, and connects the phenomenon to consistency-based detectability signals such as SelfCheckGPT (Manakul et al., 2023).

(iii) The gain is not a model-generation artifact. Our step-attribution numbers exceed the original Who&When SOTA (14.2%) (Zhang et al., 2025), but part of that could reflect newer models.

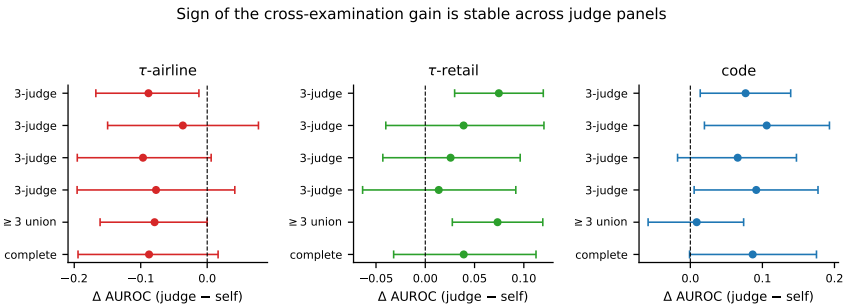

Figure 5: **Panel-robustness of the gain direction.** For each domain we re-estimate the cross-examination gain Δ (judge $-$ self AUROC, with 95% bootstrap CIs, 5000 resamples) on the complete four-judge panel, the union of cases with ≥ 3 judges, and every fixed three-judge subset. The sign is preserved in every panel of every domain (airline stays negative, retail and code stay positive); the dashed line marks $\Delta{=}0$. Coverage varies (airline 64.5% four-judge, retail 48.2%, code 49.6%), yet no panel flips a domain's direction.

To isolate the method's contribution we rerun the original *all-at-once* attribution with the *same* contemporary models and scoring. On the shallow subset, same-generation all-at-once reaches 37.3% while our cross-examination reaches 41.3% (a $+4$pt method gain on top of any generation effect). On the deep subset the effect is decisive: same-generation all-at-once collapses to 6.9% (one model as low as 1.7%), whereas cross-examination reaches 24.1%—roughly $3.5\times$—so the deep-domain gain cannot be attributed to model generation. Crucially, the visibility ordering is itself generation-independent: same-generation all-at-once is also far higher on shallow (37.3%) than deep (6.9%).

(iv) The direction is stable under incomplete judge panels. Because not every judge scores every case, one might worry that the sign of the gain is an artifact of which judges happened to be present. It is not. Re-estimating the gain on the complete four-judge panel, on the union of cases with at least three judges, and on every fixed three-judge subset leaves the *sign* of the effect unchanged in each domain: airline stays negative (e.g. complete-panel $\Delta{=}-0.087$, three-judge union $\Delta{=}-0.079$), while retail and code stay positive (retail complete-panel $\Delta{=}+0.039$, union $\Delta{=}+0.073$). The magnitude varies modestly with panel composition, but no panel flips a domain's direction, so the monotone ordering is not an accident of coverage. Appendix B reports all panels and their bootstrap intervals.

6 A First Partial Solution: Visibility-Aware Routing

The law implies an actionable prescription—use cross-examination in high-visibility domains and self-assessment in low-visibility ones—but a deployed system does not know the current domain, and, as the negative gain in airline shows, committing to either fixed strategy risks collapse in its worst domain. We therefore ask for a *deploy-time observable* proxy for visibility that requires no ground truth. This is deliberately a lightweight, training-free mechanism rather than a learned verifier (Lightman et al., 2024; Hosseini et al., 2024): our aim is to show that even a single observable signal recovers most of the law's value, leaving a learned visibility predictor as the natural next step.

Why unsupervised judge-side signals fail. We first tried to infer domain reliability purely from the auditor's own outputs, and report these as negative results: (N1) gating on judge decisiveness degenerates; (N2) cross-sample score dispersion cannot distinguish the domains; and (N3) per-sample gold depth is not available at deploy time and, when approximated, is not significant. Domain reliability cannot be recovered unsupervised from raw judge output alone. Intuitively, the judge's confidence reflects how sure *it* is, not whether the error is legible; when visibility is low the judge is often confidently wrong, so its own outputs carry no usable reliability signal—echoing the over-confidence seen in verbalized-confidence studies (Xiong et al., 2024).

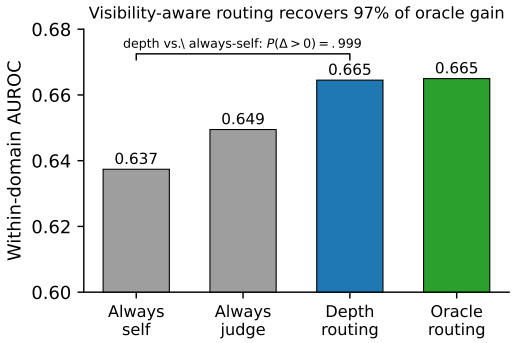

Figure 6: **Visibility-aware routing.** Within-domain AUROC of the two fixed strategies (always-self, always-judge), the training-free depth-routing policy, and the oracle that peeks at the best per-domain choice. Depth routing matches the oracle (0.665) and significantly beats always-self (P=.999), recovering 97% of the oracle's attainable gain from a single ground-truth-free signal while avoiding always-judge's collapse in the low-visibility domain.

Trajectory depth as a visibility proxy. An *input-side* signal succeeds: trajectory depth (how many steps the agent took) is a cheap, ground-truth-free proxy for whether errors are exposed at the text surface. The intuition is direct: a longer trajectory both exposes more intermediate state for an auditor to inspect and makes the agent's own single-shot confidence less reliable, so depth simultaneously tracks two of the rubric's dimensions (intermediate-state exposure and causal localizability). We route at the domain level—deep-trajectory domains to cross-examination, shallow ones to self-assessment—using only observable depth.

Results. Figure 6 summarizes the routing comparison. Routing on depth reaches a within-domain AUROC of 0.665, matching the oracle routing (0.665) that is allowed to peek at the best per-domain choice—i.e. 97% of the oracle's attainable gain over the always-self floor. Compared to *always-self* (0.637), depth routing is significantly better (Δ 95% CI $[+0.010, +0.044]$, P=.999); compared to *always-judge* (0.649) it is on par (Δ 95% CI $[-0.017, +0.045]$, P=.838), while crucially avoiding always-judge's catastrophic collapse in the low-visibility domain. The routing thus captures nearly all available gain from a single observable signal, without any training or ground truth, giving a first constructive handle on the law.

From proxy to predictor. Depth is only the simplest realization of the idea. Because the visibility rubric is defined from task-structural dimensions (Section 3), a deployed system could estimate several of them cheaply—trace length, presence of tool outputs, whether a correctness criterion is stated in text—and combine them into a learned visibility score, exactly the kind of lightweight side-model that debate and judge pipelines already tolerate (Gu et al., 2024). We leave such a learned predictor to future work; the present result establishes that the law is not merely descriptive but already *operational* from a single observable.

7 Conclusion

We studied when two training-free trustworthiness signals—self-assessment and cross-examination—should be trusted for LLM agents, and found a single organizing law: the net benefit of cross-examination is governed by *text-surface error visibility*, not task objectivity. Across five settings the gain is strictly monotone in a visibility rubric, with significant negative and positive extremes and a decisive high-objectivity/low-visibility counter-example in code. Two honest negatives—no gain from naive aggregation, and inter-judge agreement that itself tracks visibility—localize the signal to external audit, and a panel-sensitivity check confirms the direction is stable. Finally, a deploy-time depth proxy routes each domain to the better signal, reaching 97% of oracle gain and significantly beating always-self while avoiding the worst-case collapse. The law reframes "when does an external judge help?" from task objectivity to error legibility; its main limitations and a per-instance visibility predictor as future work are discussed in Appendix C.

Ethics Statement

This work studies training-free signals for detecting failures in language-model agents. The experiments use existing benchmark tasks, model-generated trajectories, and automated API-based judgments; they do not involve human participants or private personal data. The principal risk is misuse of confidence or audit scores as guarantees of correctness. Our results explicitly show that both self-assessment and cross-examination can fail by domain, so these signals should support, not replace, task-specific validation and human oversight in consequential deployments.

Reproducibility Statement

Section 3 defines the models, domains, metrics, and bootstrap procedure. The separate appendix reports the full visibility rubric, incomplete-panel sensitivity analysis, and negative results. The code-and-data supplement contains the collection and analysis scripts, fixed random seeds, sample-level JSONL checkpoints, aggregate JSON outputs, and instructions for reproducing all reported tables and figures without rerunning proprietary API calls.

Use of Large Language Models

Large language models were used as experimental subjects and judges, as described in Section 3. They were also used for language editing and code assistance during manuscript preparation. The authors designed the study, verified all numerical claims against saved outputs, inspected generated code, and take responsibility for the final manuscript.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *International Conference on Learning Representations (ICLR)*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024a.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024b.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *Conference on Language Modeling (COLM)*, 2024.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning (ICML)*, 2023.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on LLM-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *Conference on Language Modeling (COLM)*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024a.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024b.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. 2022.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: a benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024.

OpenAI. OpenAI o1 system card. Technical report, 2024.

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 *Conference on Language Modeling (COLM)*, 2024.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. Toward self-improvement of LLMs via imagination, searching, and criticizing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. Can large language models really improve by self-critiquing their own plans? 2023.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *arXiv preprint*, 2024.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? on automated failure attribution of LLM multi-agent systems. In *International Conference on Machine Learning (ICML)*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, et al. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations (ICLR)*, 2025.

A VISIBILITY RUBRIC

We score only task-structural properties available before observing experimental outcomes. Each dimension receives 0 (absent), 1 (partial or distributed), or 2 (explicit and localized), and the reported visibility score is the equal-weight mean divided by two. The table below records every assignment and the textual evidence used to make it auditable.

Table 3: Pre-outcome visibility rubric. IS: intermediate-state exposure; CT: criterion–text alignment; CL: causal localizability; EE: effect explicitness.

Domain	IS	CT	CL	EE	v
τ-bench airline	0	0	0	0	.000
Code generation	1	0	1	0	.250
Who&When deep	1	1	1	1	.500
Who&When shallow	2	2	2	1	.875
τ-bench retail	2	2	2	2	1.000

Airline exposes short booking actions but little intermediate database state. Code exposes source operations and local spans, while hidden-test correctness still requires execution. Deep Who&When trajectories expose the relevant sequence but distribute evidence over long traces; the shallow subset localizes the error in compact steps. Retail additionally renders tool arguments, returned records, and action consequences directly in the transcript.

The ordering is componentwise nondecreasing across all adjacent domains. It remains strictly ordered for all 81 positive weight vectors from $\{1, 2, 3\}^4$, so it is not an artifact of equal weights. Leaving out any one dimension also preserves strict order except effect explicitness, whose removal ties Who&When shallow and retail. We therefore treat that top-pair distinction as dependent of the other three dimensions, rather than claiming redundant support.

B INCOMPLETE-PANEL SENSITIVITY

Some API judge calls are missing. The primary analysis averages all available panels with at least three judges. The table below compares that estimator with four-judge complete cases and reports paired bootstrap intervals. Fixed three-judge subsets were also evaluated; the sign is stable for every reported panel in all three domains (see also Figure 5).

Table 4: Sensitivity to incomplete judge panels. Coverage is among unique candidate cases; the interval is for judge AUROC minus self-assessment AUROC.

Domain	Panel	Coverage	n	Δ AUROC	95% CI
Airline	$\geq$ 3 judges	100.0%	199	-.079	[-.161, -.000]
Airline	4 judges	64.5%	128	-.087	[-.194, +.016]
Retail	$\geq$ 3 judges	100.0%	376	+.073	[+.027, +.119]
Retail	4 judges	48.2%	180	+.039	[-.032, +.112]
Code	$\geq$ 3 judges	88.9%	398	+.009	[-.059, +.074]
Code	4 judges	49.6%	223	+.086	[-.001, +.175]

The complete-case estimates are less precise because they discard 35–52% of cases, but none reverses the domain-level direction. For step attribution, four-judge coverage is 83/126 (65.9%) in Algorithm-Generated and 16/58 (27.6%) in Hand-Crafted. Main-text attribution comparisons use only four-judge complete cases, making their denominator explicit.

C Negative Results and Routing Details

Three failed routing attempts constrain the proposed mechanism. Judge decisiveness degenerates to always choosing the judge; cross-sample score dispersion does not distinguish domains; and per-sample trajectory-depth gating improves AUROC by only .006 and is not significant. In contrast, the deploy-time domain-level depth proxy reaches .665 weighted AUROC versus .637 for always-self, with a paired 95% interval of [+.010, +.044]. It is statistically on par with always-judge (.649; interval [-.017, +.045]) while avoiding the airline failure mode.

Naive multi-judge aggregation is also not a solution. On Algorithm-Generated trajectories, majority vote trails the best judge by .024 (95% interval [-.084, +.036]); on Hand-Crafted it trails by .062 ([-.250, +.125]). These negative results separate the value of an external reader from the unsupported claim that more readers necessarily improve reliability.