ESTIMATING AND COMMUNICATING CORRECTNESS IN AGENT HANDOFFS

Anonymous authors
Paper under double-blind review

ABSTRACT

A correctness estimate from three sender responses enables lower-loss agent handoffs through a simple cost-aware decision rule. We fit a compact predictor to three token-probability features and communicate its correctness probability with an explanation. The sampling budget, predictor, and decision policy are configurable. The receiver reads only the note and selects a fault or requests specialist investigation. The estimator has useful ranking and lower probability error than raw self-reported confidence. On 125 incidents, its message reduces wrong diagnoses by 5.2 percentage points and increases escalation by 4.8 points relative to no warning. When escalation costs half as much as a wrong diagnosis, a gate on the same estimate lowers loss from .408 to .360 relative to the recorded probability-message decisions. Two simpler numerical messages do not reproduce this effect. Adding disagreement detail, even a complete twenty-answer tally, has no detectable effect under the original instructions. A separate comparison removes the explicit diagnosis while retaining evidence and votes. Fewer wrong diagnoses and more escalations recur on held-out incidents and with a second sender, with a clear effect in one of three receiver families. We evaluate both changes against the extra cost of escalation.

1 INTRODUCTION

One model investigates a problem; another decides what to do from its written findings. We call these models the *sender* and *receiver*, and their shared record the *handoff note*. In service diagnosis, a note might state "CPU overload" even though repeated investigations disagree. Should it report that disagreement, give a probability that the diagnosis is correct, or leave out the conclusion? These choices matter because the receiver may follow the stated diagnosis despite a warning beside it.

Naming a fault represents dispatch to the team responsible for it; a wrong diagnosis sends the incident to the wrong team. The alternative, *escalation*, requests specialist investigation. We compare the cost of wrong diagnoses with the cost of that extra investigation.

Selective prediction and learning to defer study whether to answer or pass a case to another decision-maker (Chow, 1970; Madras et al., 2018; Mozannar & Sontag, 2020). Model cascades often pass on the original input (Chen et al., 2023; Ong et al., 2024). Here the receiver sees only the note, so its decision also depends on what that note says. Prior work distinguishes predicting failure from improving intervention decisions (Zhang et al., 2026); we test concrete choices of reliability estimate, warning format, and diagnosis visibility at this written handoff.

We compare answer agreement, self-reported confidence, and a small fitted correctness predictor. We test their prediction quality, their messages, and a direct cost-aware escalation rule. Diagnosis removal is a separate comparison holding evidence and votes fixed.

We make three contributions:

1. A correctness-based handoff procedure with configurable sampling, prediction, decision policy, and note content, evaluated with a three-response estimator that reduces probability error relative to raw self-report and ranks errors even under unanimous voting (Section 3). A direct gate lowers loss relative to recorded probability-message handoffs at the evaluated low and moderate escalation costs (Section 6.1).

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.

2. Controlled handoff comparisons in which the explained probability reduces wrong diagnoses and increases escalation, while richer disagreement information alone has no detectable effect under the tested instructions (Section 4).
3. Confirmation that removing the explicit diagnosis changes decisions on held-out incidents and with a second sender, with the effect concentrated in one of three receiver families (Section 5).

2 Task and Evaluation

2.1 Incidents and datasets

RCAEval records faults deliberately injected into three microservice applications: Online Boutique, Sock Shop, and Train Ticket (Pham et al., 2024). These applications are the three *systems* referred to below. We retain 165 incidents after removing repeated injections of the same fault into the same service. Each incident has measurements and a known fault type. A diagnosis is *correct* when it matches that fault type. Cloud-OpsBench is a separate cloud-diagnosis benchmark in which the sender collects measurements through tools (Wang et al., 2026; LLM4Ops, 2026). We use it to test whether findings transfer from prepared summaries to tool-based investigations.

2.2 Sender: producing diagnoses

The primary sender is GPT-5.4. It receives a fixed text summary of the incident's measurements. Its instructions explain measurement patterns associated with network faults but omit comparable guidance for CPU, memory, and disk faults. The measurements remain visible; we withhold deployment-specific rules linking resource measurements to fault types, such as the different processor levels associated with CPU and disk saturation. This creates a controlled knowledge gap while preserving the measurements and true fault labels. Two distinct collections cover the same 165 incidents: the original twenty-response votes supply the initial warnings and diagnosis removal, while a later twenty-response probability-trace collection supplies the estimator. The probability-message study uses its first-three diagnosis and score, with the original twenty-response tally as a control. Appendix A.1 maps these sources and evaluation populations. A *draw* is one sampled response; a complete sender incident has twenty valid labels. Section 5 reports the sender replication.

2.3 Receiver: choosing a diagnosis or escalation

The main receivers are GPT-5.2 and Grok-4.3, called the GPT and Grok receivers below. Their evidence is restricted to the handoff note. They may retain its diagnosis, choose another allowed fault, or output `ESCALATE`. Choosing any fault is a *commitment*; choosing an incorrect fault is a *wrong commitment*, or wrong diagnosis. Choosing a different fault from the sender is an *override*, which can itself be correct or wrong. The third-receiver check is introduced with the diagnosis-removal results.

2.4 Measures and paired comparisons

Correctness probability estimates whether the selected diagnosis is right. AUROC measures how often a correct diagnosis scores higher than an incorrect one, with half credit for ties; .5 is chance ranking (Fawcett, 2006).

Calibration is agreement between estimated probabilities and observed correctness frequencies (Guo et al., 2017). For example, about 80% of comparable diagnoses assigned probability .8 should be correct. The Brier score measures mean squared probability error (Brier, 1950). Expected calibration error (ECE) compares predictions with correctness in five equal-width bins. Lower is better for both: Brier measures overall probability error, while ECE summarizes calibration.

For receiver decisions, let w be the wrong-diagnosis rate and e the escalation rate. A correct diagnosis costs zero, a wrong dispatch costs one, and escalation costs c. Expected loss is

$$L(c) = w + ce. \tag{1}$$

We compare costs over a range of c. Each comparison pairs conditions on the same incidents. We average responses within incident and weight receivers equally. Incidents are the independent units.

Receiver tests jointly compare wrong diagnoses and escalation, with Holm correction (Holm, 1979); a corrected $p < .05$ supports a change in the outcome pair. Appendices J and A give inference procedures and study status.

3 Choosing the Reliability Estimate

3.1 Repeated sampling and disagreement

A straightforward way to warn a receiver is to run the sender repeatedly and report where its answers disagree (Wang et al., 2023). We use twenty responses. A *vote tally* counts responses for each diagnosis; the *leading share* is the largest count divided by the total. The tally, leading share, or vote entropy can report agreement. An incident is *unanimous* when all responses agree and *divided* otherwise. Votes count the final fault labels, so different explanations of the same diagnosis belong to the same category.

With twenty responses, leading share reaches correctness AUROC .659 (Appendix C). But repeated calls add cost, and unanimous responses can still be wrong (Appendix C.4). Agreement is not a calibrated correctness probability. Can we obtain a more useful reliability estimate from fewer responses? Our method selects response count k, features ϕ, and predictor g according to prediction quality and operating cost. We evaluate these choices here and their messages in Section 4.

Algorithm 1 organizes the procedure. The configuration θ fixes the sampling budget, features, fitted predictor, decision policy, and note content. Policies are gate-only, receiver-only, or gate-then-receiver. Selection uses prediction quality, sampling cost, and decision loss at a stated cost or cost weighting. Appendix C gives the detailed procedure.

Algorithm 1 Configuring and using a correctness-based handoff

Require: Sender S, receiver R, development cases D; prediction and cost criteria
Require: Candidate budgets, features, predictors, decision policies, and note formats
Offline configuration
1: Cache sender responses on D; construct disagreement-warning baselines.
2: Compare (k, ϕ, g) using held-out prediction quality and sampling cost.
3: Compare gate, receiver, and hybrid policies with candidate notes on matched cases.
4: Separately compare answer inclusion/removal with other note content fixed.
5: Choose an evaluated configuration for the stated cost objective; freeze θ.
Online handoff for case x, evidence E, and escalation cost c
6: Sample r with k_θ responses; choose $\hat{y}$ by the fixed tie rule; compute $\hat{q} = g_\theta(\phi_\theta(r, \hat{y}, E))$.
7: If π_θ includes a gate and $\hat{q} < 1 - c$, **return** `ESCALATE`.
8: Otherwise, **return** $\hat{y}$ for gate-only, or the receiver's decision on the configured note.

3.2 Alternatives to repeated-sampling warnings

Step 2 compares predictors, features, and response budgets; the following three subsections evaluate these choices. The final-answer token probability is an inexpensive candidate, but it is 1.0 on every RCAEval incident and has AUROC .500. Another option is to ask the sender for a 0–100 confidence judgment about its fixed diagnosis, called *self-reported confidence*, obtained with one additional model call. A learned alternative maps uncertainty-related measurements in the responses to an estimated probability that the diagnosis is correct.

For a receiver choosing between accepting the diagnosis and escalating, let $q(X)$ be the probability that the diagnosis is correct given handoff information X. Correctness has a direct decision interpretation.

Proposition 1 (Accept-or-escalate decision)**.** *With zero loss for a correct acceptance, unit loss for a wrong acceptance, and escalation loss $c \in [0, 1]$, an expected-loss-minimizing receiver accepts when*

$$q(X) \geq 1 - c. \tag{2}$$

Knowing $q(X)$ therefore determines the optimal accept-or-escalate choice for any escalation cost c.

The optimum uses the true probability $q(X)$. Our gate applies Chow's rule to its estimate $\hat{q}$ (Chow, 1970), and Section 6.1 measures its loss. Receivers can also choose another fault; Proposition 2 in Appendix B gives that decision rule.

We fit regularized logistic regression (LR) to historical correct/wrong diagnosis labels, leaving the sender unchanged. The predictor family is configurable; all reported probability messages use the three-feature LR. Appendix C gives the feature calculation and logistic map (Equations 9 and 10).

Each evaluation fold trains on 110 incidents from two RCAEval systems and predicts the 55 in the third. This *cross-fitting* prevents an incident's label from training its own prediction. Scaling and five-fold incident-grouped regularization selection by log loss use only training data, with equal total fitting weight per incident. Full response-grouping and training details are in Appendix C.

For the fixed first-three diagnoses on all 165 incidents, the estimator reduces Brier error from .278 for raw self-report to .200 (Holm $p = .0089$). Self-reports average about 25 percentage points above observed correctness. Fitting a logistic calibration correction to self-reports on other systems reduces their Brier score to .218; the remaining difference is inconclusive ($p = .359$), as is the ranking comparison.

On the 125 probability-message incidents, AUROC is .748. Calibration varies by system: pooled ECE is .033, while within-system ECE is .092, .081, and .095. Appendix C gives the system-level results, calibrated self-report comparison, and an exploratory combination with improved external ranking. The latter two are predictor comparisons; the receiver study uses the original three-feature score.

3.3 Which features should the predictor use?

Each complete sender response provides probabilities for its generated text pieces, or *tokens*. Our evaluated feature set summarizes these probabilities before the final answer declaration:

1. **Mean chosen-token log probability:** how probable the generated tokens were, averaged on a logarithmic scale.
2. **Mean top-two probability gap:** the average difference between the two most likely next-token probabilities.
3. **Coarse-grained token entropy:** how spread out the logged next-token probabilities were, using the top five alternatives, the chosen token, and one bin for all unlisted probability mass.

We first average over tokens within each response, then average the response summaries. The number of feature types and the number of responses are separate choices. Two responses can give the same fault label while differing in the probabilities throughout their preceding text. The token features preserve this additional information.

Follow-up feature comparisons found no reliable gain from adding diagnosis and symptom inputs (Appendix C.3).

A single token-gap feature also supports effective correctness estimation. On the same 165 incidents with whole-system holdouts, gap-only LR achieves AUROC .752 and Brier .203, close to .756 and .200 for the three-feature model (Appendix C.1). These two evaluated predictor configurations support selecting the feature set within Algorithm 1. The receiver experiments use the three-feature configuration.

Token-level information distinguishes errors even under unanimity. Among 80 originally unanimous diagnoses, including 23 wrong ones, the three-feature score reaches AUROC .738 [.613, .853] and the raw token gap .739 [.616, .852], versus .500 for constant agreement. The paired LR–gap difference is inconclusive; both retain information that the votes omit (Appendix C.4).

3.4 How many responses are needed?

Table 1 compares scores on the same first k responses, matching the handoff's first-three diagnosis. At three responses, LR improves AUROC over leading share by .179 [.086, .267] (Holm $p = .0024$),

Table 1: Budget-matched AUROC on 165 incidents using the fixed first k traced responses. LR, leading share, one minus normalized vote entropy, and mean token gap target the same diagnosis within each row. LR predictions exclude the evaluated system; all other scores are unfitted. Diagnoses can differ across budgets.

Responses	Diagnosis accuracy	LR	Share	Entropy	Gap
1	55.2%	.735	.500	.500	.701
3	54.5%	.756	.576	.576	.762
5	55.8%	.753	.605	.602	.744
20	57.0%	.747	.653	.664	.745

Table 2: Warning formats and their meanings. Examples are illustrative; the full tested texts are in Appendix K. A single number can describe agreement or correctness, which are different quantities.

Format	Information in the assessment
No warning	No assessment sentence.
Qualitative warning	"The analysis was divided on this."
Agreement score	One minus normalized vote entropy.
Full vote tally	Response counts for every diagnosis, including zeros.
Leading share	Fraction of responses choosing the leading diagnosis.
Self-reported confidence	The sender's own stated chance that its fixed diagnosis is correct.
Estimated correctness	The trained predictor's probability, with an explanation of correctness for similar cases.

but not over the mean token gap ($-.006$ [$-.044$, $+.031$]). Appendix C.1 gives all paired comparisons; cyclic-window averages in Appendix C assess response-selection sensitivity.

Three responses provide similar observed ranking to twenty with 85% fewer sender calls. LR turns token information into a correctness probability; the raw gap provides similar ranking. A single response may suit a tighter budget. The message study uses three responses and the original LR.

3.5 Evaluated configuration

Our RCAEval handoff uses $k = 3$, the three token features, and LR, with response indices $0, 1, 2$ and a fixed fault-option tie rule. It states $\hat{q}$ as an integer percentage with an explanation of correctness for similar cases. Two of three votes can still yield a low estimate. Figure 2 in Appendix C shows the shared estimator and the three decision policies; Appendix K.3 gives the exact message. The receiver chooses whether to keep the diagnosis, override, or escalate.

4 Choosing the Warning Format

4.1 Candidate warning formats

The receiver policy in Algorithm 1, step 3, compares the messages in Table 2: agreement summaries and correctness probabilities. We first test whether more disagreement detail helps, then test the estimated probability against matched controls and simpler numerical messages.

Why the tally is a reasonable baseline. An ideal receiver can decode the full tally into a correctness probability under the fitted model in Appendix B; the experiment tests whether LLM receivers use that information.

4.2 Disagreement detail alone has no detectable effect

Under the original receiver instructions, all 165 incidents receive four note versions: no warning, a qualitative warning, an agreement score, and the full tally. Evidence and the twenty-response leading diagnosis stay fixed; note lengths and fault option order are matched. Each receiver responds three times per version, giving 3,960 decisions. The tally changes wrong diagnosis by -0.1 points [-2.3,

Table 3: Estimated-probability message minus each comparison format on 125 incidents. Differences are percentage points with simultaneous 95% intervals for the two outcomes. Holm adjustment uses the original two-comparison family for rows 1–2 and the separate five-comparison follow-up family for rows 3–4. The last two rows are later comparisons that reuse the earlier probability-message responses; Table 27 gives the full follow-up family.

Comparison format	Wrong diagnosis	Escalation	Holm p
No warning	-5.2 [-8.0, -2.4]	$+4.8$ [$+2.0$, $+7.6$]	.0005
Full tally	-2.9 [-6.7, $+0.8$]	$+4.8$ [$+1.1$, $+8.5$]	.0135
Leading share	-3.5 [-6.8, -0.1]	$+4.7$ [$+1.3$, $+8.0$]	.0265
Raw self-report	-3.2 [-6.0, -0.4]	$+4.0$ [$+1.2$, $+6.8$]	.0288

$+2.1$] and escalation by 0.0 [-2.2, $+2.2$] relative to no warning. The other planned comparisons likewise show no detectable effect (Appendix D).

Count encoding, warning position, and fuller-evidence controls also fail to establish a pooled tally effect (Appendix D).

Can the warning be recovered and explained? A separate decoder reconstructs the candidate diagnoses and disagreement on 100% of draws when both are explicit. Concealing this information reduces diagnosis recovery to 64.6% for GPT and 20.6% for Grok (Appendix G). Instructions explaining divided versus unanimous votes raise Grok's escalation to 81.7% on divided tallies, versus 0% without the tally; GPT reaches 8.3%. More domain background does not improve diagnosis accuracy (Appendices G and H).

4.3 The estimated-probability message changes decisions

We compare three notes on 125 incidents: no warning, the exact twenty-answer tally, and the probability message. Each states the same first-three diagnosis and evidence; lengths differ pairwise by at most 5%. The tally labels its values as response frequencies. We compare each numerical estimate together with its explanation. Each receiver makes three high-reasoning responses per note, with paired option order and one request per draw: all 2,250 responses parse. The two probability-versus-control comparisons were specified before these calls.

The probability message reduces wrong diagnoses and increases escalation relative to no warning (Table 3). The benefit is avoiding wrong dispatches through increased review. Against the tally, the message raises escalation. Appendix F gives the absolute outcome rates.

Two simpler numerical messages show the leading share or raw self-report as an explained percentage. Neither shows a statistically reliable change from no warning in the two-outcome test; the estimated-probability message produces fewer wrong diagnoses and more escalation than either. These follow-ups reuse the original control responses (Appendix F).

Before the probability-message calls, we also divided the ranked scores into low, middle, and high thirds. Relative to no warning, the low group has 11.9 points fewer wrong diagnoses and 12.3 more escalation points. The corresponding changes are $-2.4/+2.0$ in the middle and $-1.2/0.0$ in the high group. The score groups identify where the message has its greatest effect; the next experiment changes the number within matched incidents. Both receivers respond relative to no warning.

4.4 Changing only the displayed value

Exploratory experiments change only an assigned number, holding evidence, diagnosis, and sentence fixed. On twelve RCAEval incidents, GPT escalation is 0% at correctness values 80, 65, 50, and 35%, and 11.1% at 20%; direct uncertainty remains at 0% across the corresponding 20–80 scale. Only one incident changes its majority decision. An "agent is sure" version also yields 0% escalation. Appendix E gives the full results and continuation criteria.

On eighty tool-based Cloud-OpsBench incidents, assigned probabilities of 80, 20, and 5% raise pooled escalation from 7.8% to 30.0% and 43.4% (Holm $p \approx .0001$ for both changes). For Grok, escalation rises from 6.25% at the 80% value to 72.5% at the 5% value; GPT's changes are inconclusive.

Table 4: Diagnosis removed minus diagnosis present, pooled over the two main receivers; the reduction is concentrated in Grok. Differences are percentage points with simultaneous 95% intervals. The first row is held out from discovery; the second uses the replacement sender. Both pooled results have $p \leq .0001$.

Sender	Incidents	Wrong diagnosis	Escalation
GPT-5.4	125	-5.7 [-9.2, -2.3]	$+11.9$ [$+8.4$, $+15.3$]
GPT-4.1	164	-7.8 [-11.2, -4.5]	$+14.8$ [$+11.5$, $+18.2$]

Because these values are assigned independently of correctness, this tests numerical response under the fixed sentence, not calibration (Appendix E.5).

The smaller sender budget and fewer wrong diagnoses than leading-share and raw-self-report messages support our correctness-based approach (Table 3; Appendix F.1). This gain comes with more escalation, whose cost Section 6 evaluates.

5 Should the Note State the Diagnosis?

5.1 Remove the conclusion, retain the evidence

The probability-message experiment retains the diagnosis. A separate comparison implements Algorithm 1, step 4: does the explicit conclusion encourage the receiver to follow it? Two notes contain identical measurements and the full twenty-answer tally; one also states the twenty-response leading diagnosis. Both versions preserve the diagnoses named by the tally and its vote counts. Length and fault-option order are matched. The text change is:

> **Diagnosis present:** diagnosis + evidence + full tally.
> **Diagnosis removed:** evidence + the same full tally.

An initial forty-incident experiment found an effect. We then specified the comparison on the remaining 125 incidents before collecting their responses, with separate tests for all, divided, and unanimous incidents. Removal lowers wrong diagnoses by 5.7 points and raises escalation by 11.9 points (Table 4). The changes occur on divided incidents; unanimous incidents are unchanged, where the tally already identifies a single supported diagnosis.

5.2 Confirmation with a second sender

We repeat the comparison with GPT-4.1 as sender, keeping the note format and receiver instructions fixed. It supplies twenty responses per incident; 164 incidents are complete, of which 96 have divided votes. Diagnosis removal again reduces wrong diagnoses and increases escalation. This is a replication of the diagnosis-visibility comparison with a second sender.

5.3 Receiver differences and later disclosure

The Grok receiver shows the reduction with both senders. GPT's effects remain inconclusive, as does a prospective check with the third receiver, Kimi-K2.6. Appendix G gives the per-receiver estimates and the divided/unanimous comparisons. In an exploratory two-turn test, receivers first decide without the diagnosis, then see it without new evidence. Of 120 decisions, 37 move toward that diagnosis and none move away; wrong diagnosis rises by 14.2 points (Appendix G). Appendix H separately counts copying, overriding, and escalation.

6 Costs and Transfer

Section 6.1 compares the gate and hybrid policies in Algorithm 1, step 3; costs and transfer inform the configuration choice in step 5.

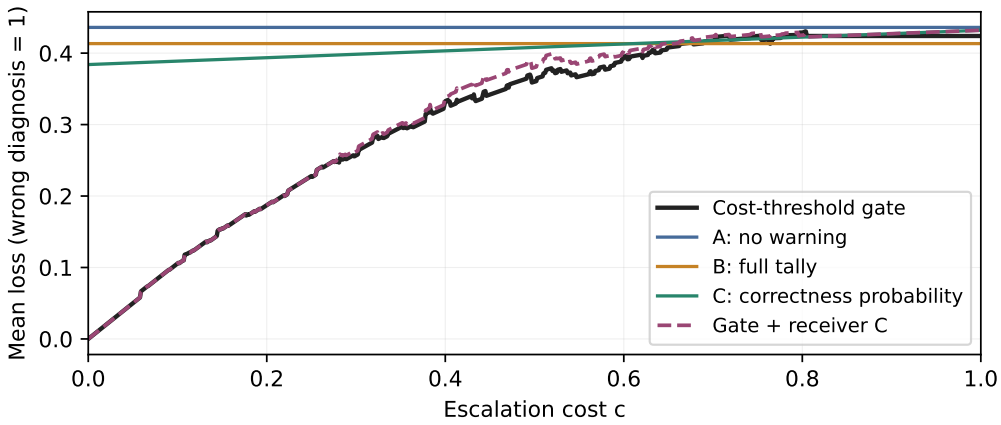

Figure 1: Pooled decision loss on 125 incidents. The hybrid uses receiver C after a case passes the gate. Gate actions vary with c; the recorded A/B/C decisions stay fixed. Lower is better.

6.1 A COST-AWARE GATE LOWERS DECISION LOSS

The estimated probability supports a direct decision rule: accept the fixed sender diagnosis when $\hat{q} \geq 1 - c$ and otherwise escalate (Proposition 1). The gate receives numerical c; receivers retain their decisions under the original qualitative cost instructions. On the same 125 incidents, averaging loss uniformly over $c \in [0, 1]$ gives .312 for this gate versus .436, .413, and .408 for no warning, the full tally, and the probability message. The reduction against the probability message is 23.5%, or .096 [.060, .133] in loss units. All three pooled comparisons reject after Holm correction ($p = .00015$ each).

The gain is strongest when escalation is inexpensive (Figure 1). At $c = .5$, loss falls from .408 to .360. The hybrid escalates below the same threshold and otherwise uses the recorded receiver decision. Both gate-first policies lower average loss relative to receiver-only handoffs; receiver review adds no demonstrated gain over the direct gate. Appendices J.2 and J.3 give the paired comparisons and outcome counts.

6.2 INFERENCE AND ESCALATION COSTS

Three responses reduce sender inference expense by 85.1% at the illustrative prices in Appendix J.1; the overall trade-off also includes receiver and specialist costs.

Decision loss measures wrong diagnoses and escalation in units of a wrong dispatch, separately from token expense. Substituting the probability-message versus no-warning rates into Equation 1 gives

$$\widehat{L}_{\text{probability}}(c) - \widehat{L}_{\text{none}}(c) = (.384 - .436) + c(.048 - 0) = -.052 + .048c. \tag{3}$$

The point estimates favor the probability message throughout $c \in [0, 1]$; the break-even point $.052/.048 \approx 1.08$ lies beyond that range. The tally comparison has estimated break-even cost .611 with a wide interval $[-.206, 1.303]$; Appendix F gives the remaining message comparisons.

Diagnosis removal incurs more escalation per avoided wrong diagnosis. Its break-even cost is .483 [.305, .667], about half the cost of a wrong dispatch (Figure 3; Appendix G). These are relative cost estimates, not probability thresholds for showing or hiding the diagnosis.

6.3 WHAT TRANSFERS TO ANOTHER SETTING?

We test transfer from RCAEval to tool-based Cloud-OpsBench investigations. The unchanged correctness estimator retains useful ranking, while ECE rises to .351 (Appendix C). This separates transferable error ranking from probability calibration. In a new deployment, validate calibration and gate loss at the intended cost, and recalibrate as needed. The assigned-number experiment in Section 4.4 establishes receiver responsiveness to the displayed values.

In Cloud-OpsBench the receiver sees tool calls and observations. On fifty divided cases, diagnosis removal changes wrong diagnosis by $+2.5$ points $[-4.0, +9.0]$ and escalation by -6.0 $[-12.5, +0.5]$, joint $p = .1057$. These point changes oppose RCAEval, with intervals spanning zero (Appendix I).

The workplace-decision dataset SteerBench-Work supplied eleven divided sender cases, below the 26 required to continue. Receiver evaluation therefore stopped before collection (Appendix I).

7 Related Work

Correctness estimation and deferral. Selective prediction studies when to answer or decline (Chow, 1970; Geifman & El-Yaniv, 2017; 2019); learning to defer and model cascades select another decision-maker (Madras et al., 2018; Mozannar & Sontag, 2020; Chen et al., 2023; Ding et al., 2024; Ong et al., 2024; Hu et al., 2024; Kotte, 2026). Agent benchmarks extend these questions to missing information, failed tools, and risky actions (Luo et al., 2026; Liu et al., 2026a; Piatrashyn et al., 2026). Self-consistency and semantic entropy estimate uncertainty from repeated answers (Wang et al., 2023; Farquhar et al., 2024; Kossen et al., 2024). Kadavath et al. (2022) study model self-evaluation of answer correctness; LM-Polygraph collects uncertainty estimators for language generation (Fadeeva et al., 2023). UCCI selects calibrated routing thresholds under an accuracy constraint, while ReDAct selects thresholds for a deferral budget (Kotte, 2026; Piatrashyn et al., 2026). Routing chooses which model should answer; our receiver acts on an existing diagnosis and note. We distinguish correctness prediction from changes in receiver decisions, as does work on oversight (Zhang et al., 2026).

Communication and reliance. Guided deferral and explanatory abstention attach information to a handoff (Strong et al., 2024; Xie et al., 2026). Language models express confidence inconsistently (Xiong et al., 2024; Yona et al., 2024; Zhou et al., 2024), and repackaging can overstate it (Shi et al., 2026). Whereas Liu et al. (2026b) estimate reliability across an agent graph, we hold a note's evidence fixed and test what its recipient does with different warnings. Human–AI studies likewise find context-dependent effects of explanations and interface changes (Vasconcelos et al., 2023; de Jong et al., 2025; Lee & Chew, 2023). Blackwell's comparison of information sources (Blackwell, 1951; 1953) and task-oriented communication (Gündüz et al., 2023) motivate the appendix's comparison of messages by the decisions they can support.

Root-cause analysis. RCACopilot combines diagnostic information with LLM-based cause prediction and explanation (Chen et al., 2024). OpenRCA evaluates root-cause reasoning over telemetry (Xu et al., 2025), and Cloud-OpsBench provides interactive, evidence-grounded investigations (Wang et al., 2026). We study the subsequent handoff: how a recipient uses the investigation's diagnosis and reliability estimate.

8 Limitations

The sender's instructions deliberately omit resource-fault guidance. Appendix A records the exploratory stages and reused controls. Escalations request review, and specialist resolution is outside the measured outcomes. RCAEval includes a primary sender and a replication sender, with transfer checks in Cloud-OpsBench. Deployment requires rechecking prediction quality, receiver decisions, and costs for the current models and data.

9 Conclusion

Our three-response estimator turns token information into a useful correctness probability with 85% fewer sender responses than twenty. Its explained message reduces wrong diagnoses and increases escalation relative to no warning and the tested simpler numerical messages. At the evaluated low and moderate escalation costs, a direct gate lowers loss relative to the recorded probability-message handoffs.

Removing the explicit diagnosis also reduces wrong diagnoses with two senders, with clear effects for Grok. Use the estimate to make escalation decisions explicit. Receiver review permits diagnosis revision; here it added no demonstrated loss benefit beyond the gate.

AI USE STATEMENT

We used generative AI for research organization, drafting, and code assistance. Experiments use saved responses under the stated protocols; analysis code computes their results. The authors reviewed the AI-assisted material and are responsible for all claims, citations, and final content.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix C specify the estimator. Appendix A records study designs; Appendix J explains the analysis; Appendices D–I give full results and request policies. Verbatim prompts are last, in Appendix K.

The gate and hybrid comparisons are computed from saved scores and receiver decisions using the rules in Appendices J.2 and J.3.

REFERENCES

David Blackwell. Comparison of experiments. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 1951.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24 (2), 1953.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xue-Chao Gao, Hao Fan, Ming Wen, Jun Zeng, Supriyo Ghosh, Xuchao Zhang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024. URL https://arxiv.org/abs/2305.15778.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Sander de Jong, Ville Paananen, Benjamin Tag, and Niels van Berkel. Cognitive forcing for better decision-making: Reducing overreliance on AI systems through partial explanations. *Proceedings of the ACM on Human-Computer Interaction*, 2025. doi: 10.1145/3710946.

Dujian Ding, Ankur Mallick, Chi Wang, et al. Hybrid LLM: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, 2024.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. LM-Polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 446–461, 2023. doi: 10.18653/v1/2023.emnlp-demo.41. URL https://aclanthology.org/2023.emnlp-demo.41/.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630, 2024.

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

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2151–2159, 2019.

Deniz Gündüz, Zhijin Qin, Iñaki Estella Aguerri, Harpreet S. Dhillon, Zhaohui Yang, Aylin Yener, Kai-Kit Wong, and Chan-Byoung Chae. Beyond transmitting bits: Context, semantics, and task-oriented communications. *IEEE Journal on Selected Areas in Communications*, 41(1):5–41, 2023. doi: 10.1109/JSAC.2022.3223408.

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, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

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

Qian Hu, Jacob Bieker, Xiang Li, et al. RouterBench: A benchmark for multi-LLM routing system. *arXiv preprint arXiv:2403.12031*, 2024.

Saurav Kadavath, Tom Conerly, Amanda Askell, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Jannik Kossen, Jiatong Han, Muhammad Razzak, et al. Semantic entropy probes: Robust and cheap hallucination detection in language models. *arXiv preprint arXiv:2406.15927*, 2024.

Varun Kotte. UCCI: Calibrated uncertainty for cost-optimal LLM cascade routing. *arXiv preprint arXiv:2605.18796*, 2026.

Min Hun Lee and Chong Jun Chew. Understanding the effect of counterfactual explanations on trust and reliance on AI for human-AI collaborative clinical decision making. *Proceedings of the ACM on Human-Computer Interaction*, 2023. doi: 10.1145/3610218.

Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Boroojeny, and Varun Chandrasekaran. AgentAbstain: Do LLM agents know when not to act? *arXiv preprint arXiv:2607.10059*, 2026a.

Yaokun Liu, Yifan Liu, Daniel Yue Zhang, Ruichen Yao, Zelin Li, and Dong Wang. PropUQ-MAS: Propagation-aware uncertainty quantification for LLM multi-agent systems. *arXiv preprint arXiv:2608.22130*, 2026b. EMNLP 2026 Main Conference.

LLM4Ops. Cloud-OpsBench: A reproducible benchmark for agentic root cause analysis in cloud systems. GitHub repository, 2026. URL https://github.com/LLM4Ops/Cloud-OpsBench/tree/64891869d082639b221ae190fc479fadc275b83e. Commit 64891869d082639b221ae190fc479fadc275b83e.

Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act? *arXiv preprint arXiv:2606.28733*, 2026.

David Madras, Toni Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 7076–7087, 2020.

Isaac Ong, Amjad Almahairi, Vincent Wu, et al. RouteLLM: Learning to route LLMs with preference data. *arXiv preprint arXiv:2406.18665*, 2024.

Luan Pham, Hongyu Zhang, Huong Ha, Flora Salim, and Xiuzhen Zhang. RCAEval: A benchmark for root cause analysis of microservice systems with telemetry data. *arXiv preprint arXiv:2412.17015*, 2024.

Dzianis Piatrashyn, Nikita Kotelevskii, Kirill Grishchenkov, Nikita Glazkov, Ivan Nasonov, Ilya Makarov, Timothy Baldwin, Preslav Nakov, Roman Vashurin, and Maxim Panov. ReDAct: Uncertainty-aware deferral for LLM agents. *arXiv preprint arXiv:2604.07036*, 2026.

Kaiwen Shi, Zheyuan Zhang, Han Bao, Colby Nelson, and Yanfang Ye. Confidence laundering in agent systems: Why uncertainty needs a latent carrier. *arXiv preprint arXiv:2606.20662*, 2026.

Joshua Strong, Qianhui Men, and Alison Noble. Trustworthy and practical AI for healthcare: A guided deferral system with large language models. *arXiv preprint arXiv:2406.07212*, 2024.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, 2023. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S. Bernstein, and Ranjay Krishna. Explanations can reduce overreliance on AI systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 2023. doi: 10.1145/3579605.

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Yilun Wang, Guangba Yu, Haiyu Huang, Yujie Huang, Zirui Wang, Pengfei Chen, and Michael R. Lyu. Cloud-OpsBench: A reproducible benchmark for agentic root cause analysis in cloud systems. *arXiv preprint arXiv:2603.00468*, 2026. URL https://arxiv.org/abs/2603.00468.

Shuang Xie, Yunan Lu, Han Li, and Lingyun Wang. EARS: Explanatory abstention for reliable sub-agent modeling in large-scale multi-agent systems. *arXiv preprint arXiv:2606.18668*, 2026.

Miao Xiong, Zhiyuan Hu, Xue Lu, et al. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. *International Conference on Learning Representations*, 2024.

Junjielong Xu, Qinan Zhang, Zhiqing Zhong, Shilin He, Chaoyun Zhang, Qingwei Lin, Dan Pei, Pinjia He, Dongmei Zhang, and Qi Zhang. OpenRCA: Can large language models locate the root cause of software failures? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M4qNIzQYpd.

Gal Yona, Roee Aharoni, and Mor Geva. Can large language models faithfully express their intrinsic uncertainty in words? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Chubin Zhang, Zhenglin Wan, Xingrui Yu, Jingxuan Wu, Qi Wen, Pengfei Zhou, Wangbo Zhao, and Ivor Tsang. Calibration is not control: Why LLM-agent oversight needs intervention. *arXiv preprint arXiv:2606.21399*, 2026.

Kaitlyn Zhou, Jena D. Hwang, Xiang Ren, and Maarten Sap. Relying on the unreliable: The impact of language models' reluctance to express uncertainty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

A Study Designs

A.1 Data sources and evaluation populations

The RCAEval studies reuse 165 incidents but not one identical set of sender responses. The original collection supplies the twenty-vote histograms and leading diagnoses for warning-detail and diagnosis-removal comparisons. A later, separate collection records twenty full probability traces per incident, 3,300 responses, under the same resource-blind prompt. Each trace contains its sampled answer, token log probabilities and alternatives, usage, option order, and prompt/request hashes. The predictor uses

these traces, not the original vote counts, as input features. The deterministic telemetry summaries supply evidence; no true-fault label is included in a sender or receiver prompt.

"Held out" has two distinct meanings. Every correctness score excludes the incident's whole software system from fitting. The 125-case receiver panel excludes the forty diagnosis-removal discovery incidents, but is not a new dataset held out from every prior analysis. On eight of these 125 incidents the first-three traced diagnosis differs from the original twenty-vote leader. We preserve that evidence rather than alter the tally; the 117-case aligned sensitivity is reported in Appendix F.1. The original-vote and traced twenty-response diagnoses also need not agree.

A.2 Study status

Table 6 separates the evidence used in the main argument from the follow-up studies. A *confirmatory* comparison has its population, comparisons, and statistical correction specified before the relevant responses are collected. *Exploratory* comparisons investigate questions suggested during development. These descriptions refer to our recorded protocols, not a public preregistration. A *continuation criterion* determines whether the next experiment will run. When it is not met, we still report the completed experiment and state which follow-up was not run. A rule specified before one experiment can still have been motivated by an earlier result; the history below makes this distinction. The 125-incident set is held out from the original forty-incident diagnosis-removal discovery, not from every subsequent analysis in this research program. The full prompt templates are collected in Appendix K, after the results and analysis conventions.

A.3 Execution settings and provenance

The repeated probability-trace collection uses the deployment label `gpt-5.4`, temperature 1.0, and the top five token alternatives. The runner sets neither top-p, a reasoning-effort setting, nor an explicit output-token cap for these calls; it uses a 180-second request timeout. The complete cache contains 3,300 usable responses from 3,300 recorded attempts. Fault-option shuffling is deterministic from incident and sample identifiers; the exact orders and request hashes are retained. These shuffling seeds are not claimed to control hosted-model sampling.

The probability-message readers are `gpt-5.2` and `grok-4.3`, with `reasoning_effort=high`, three evaluation draws per note, 180-second timeouts, and one scheduled attempt per draw. Temperature, top-p, and output-token cap are not explicitly set in this path; provider defaults apply. The corrected self-report collection uses `gpt-5.4`, high reasoning, a 16,384-token output cap, and a 600-second timeout; its failed 2,048-token predecessor is kept separate. The older warning/removal and native studies retain their study-specific missing-response and recovery rules rather than inherit the later no-retry protocol.

Table 5: Sources of the diagnosis, score, and tally. Collections share incidents; sharing an incident does not make the sampled diagnoses identical.

Study	Population and diagnosis	Reliability input
Original warnings/removal	165 incidents; original twenty-vote leader	Original tally; 40 removal-discovery and 125 confirmation incidents
Correctness estimation	Same 165; leader of the chosen traced response group	Token features from that group; train on 110, evaluate on 55 by system
Probability messages	125 removal-confirmation incidents; first-three traced diagnosis	Score for that diagnosis; tally control retains original twenty-vote counts
Gate and hybrid replay	Same 125 incidents and first-three diagnosis	Frozen score; hybrid also uses the original probability-message receiver decisions
Self-report baseline	165 fixed-three traced diagnoses	Separate confidence assessment of the fixed diagnosis, with unchanged telemetry
Unanimous-score check	80 originally unanimous incidents; all scored diagnoses match	Saved first-three trace scores; 70-case overlapping sensitivity

Table 6: Study families, populations, and analysis status.

Study	Population	Status
Warning detail	165 incidents; four formats	Confirmatory
Diagnosis removal, discovery	40 incidents	Exploratory
Diagnosis removal, confirmation	125 held-out incidents	Confirmatory
Second sender	164 complete incidents	Confirmatory pooled contrast
Third receiver, Kimi	125 held-out incidents	Prospective family check
Probability message	125 held-out incidents	Confirmatory contrasts
Self-reported confidence	165 fixed diagnoses	Prespecified paired comparison
Leading-share and self-report messages	125 previously studied incidents	Prespecified comparisons; reused controls
Synthetic-value sequence	Four small GPT panels	Exploratory; rules specified per stage
Tool-based synthetic probabilities	80 Cloud-OpsBench incidents	Exploratory; prespecified value comparisons
Initial probability-message pilot	36 incidents	Prespecified continuation criterion
Interpretation instructions and response modes	20 divided incidents; controls	Exploratory
Independent decoders	Synthetic and real notes	Exploratory
Staged diagnosis reveal	20 divided incidents	Exploratory
Token-probability scores	165 incidents; 3,300 traces	Cross-fitted score analysis
Budget-matched baselines and costs	Fixed first 1/3/5/20 traces on 165 incidents	Exploratory cached-data analysis; no refitting
Cost-threshold gate and hybrid	Same 125 probability-message incidents	Exploratory saved-score comparisons with original A/B/C decisions
Feature and unanimous-score checks	165 and 80 incidents, respectively	Exploratory follow-ups after receiver experiments
Ideal-receiver enumeration	63,756 possible tallies	Fitted-model calculation
Receiver competence	40 Cloud-OpsBench cases	Criterion not met; follow-up not run
Tool-based transfer	90 Cloud-OpsBench groups	Confirmatory matched subset
SteerBench-Work feasibility	106 scenarios	Too few divided cases; reader stage not run

LR uses standard scaling, L2 regularization, and candidate inverse penalties $C \in \{0.1, 1, 10\}$, selected by five incident-grouped validation folds and log loss. The iteration cap is 5,000 and the fit seed is 20260828; other settings follow the recorded scikit-learn version's defaults. The original triplet selects $C = 10$ in all outer folds. AUROC bootstraps use seed 20260828, paired Brier intervals 20260814, and sign swaps 20260809. The cached-data audit uses Python 3.14.7, scikit-learn 1.9.0, NumPy 2.5.1, and SciPy 1.18.0. Its environment also contains OpenAI 2.53.0 and tiktoken 0.13.0; per-call historical library versions were not stored.

The research log dates the probability-trace work to August 27–29, 2026, the probability-message comparison's completion to September 6, and the corrected confidence follow-up to September 7. These are stage records, not per-request timestamps. Per-call calendar times and immutable hosted model snapshot identifiers were not stored; file modification dates cannot recover them. Deployment names therefore identify the tested interfaces, not frozen weights. The later numerical-message controls reuse earlier receiver decisions, so collection-time drift remains unmeasured.

B Decision-Theoretic Details

Suppose a receiver can only accept the stated diagnosis or escalate. Let X contain the evidence, diagnosis, and sender-response information; Y is the true fault and $\hat{y}(X)$ the stated diagnosis. Its correctness probability is

$$q(X) = \Pr(Y = \hat{y}(X) \mid X) \tag{4}$$

Proof of Proposition 1. Given handoff state X, accepting has expected loss $1 - q(X)$ and escalating has loss c. Accepting minimizes loss when $1 - q(X) \leq c$, which gives Equation 2.

For minimality across costs, suppose a summary maps two states x_1 and x_2 to the same message but $q(x_1) < q(x_2)$. Choose a threshold t strictly between them and set $c = 1 - t$. The optimal decision escalates at x_1 and accepts at x_2. A policy observing only the merged message must act identically in both states and is suboptimal in at least one. Thus a sufficient summary for every cost cannot merge different values of q. □

Our receivers may instead override the stated diagnosis. The analogous optimal decision then depends on which diagnosis is most likely as well as how likely it is.

Proposition 2 (Choose-or-escalate decision)**.** *Let $p_y(X) = \Pr(Y = y \mid X)$, and let $y^*(X)$ maximize this probability under a fixed tie rule. Write $p^*(X) = p_{y^*(X)}(X)$. Choosing $y^*(X)$ when $p^*(X) \geq 1 - c$ and escalating otherwise minimizes expected loss. Where the maximizer is unique, a summary that supports this optimum for every cost must preserve both the maximizing diagnosis and its probability.*

Proof. Choosing diagnosis y has conditional loss $1 - p_y(X)$. Its minimum over diagnoses is $1 - p^*(X)$, which is compared with c. A summary that merges two different unique maximizers cannot choose the optimal diagnosis in both states when $c = 1$. If it preserves the maximizer but merges different p^* values, a cost threshold between those values gives the same contradiction as in the accept-or-escalate case. □

The pair (y^*, p^*) suffices for this decision. Our score instead estimates correctness of the sender's chosen diagnosis $\hat{y}$. The receiver also sees the evidence and can use it to propose another fault. We measure the value of that review through its observed decisions and the hybrid policy in Appendix J.3.

Lemma 1 (Tally size)**.** *A tally of twenty votes over m diagnoses has $\binom{20+m-1}{m-1}$ possible values. An exact binary encoding therefore needs 14 bits for five diagnoses or 16 bits for six. Encoding the ordered twenty-answer sequence instead requires 47 or 52 bits.*

Proof. The tally is a weak composition of twenty into m nonnegative counts. Taking the ceiling of the base-two logarithm gives $\lceil \log_2 \binom{24}{4} \rceil = 14$ and $\lceil \log_2 \binom{25}{5} \rceil = 16$. There are m^{20} ordered sequences, giving $\lceil 20 \log_2 m \rceil = 47$ and 52. □

This calculation shows that a tally need not be large in a compact encoding. It does not account for prose overhead, imperfect calibration, or the difficulty a language model has in using that encoding.

Fitted enumeration. Table 7 asks how much decision information a tally can carry under a fitted probability model. The diagnosis is the tally's leading fault, with ties resolved by the fixed candidate order. Each policy either accepts that diagnosis or escalates.

We fit an empirical mixture of multinomials separately for RCAEval RE1 and RE2, using the original twenty-vote histograms on 165 incidents. For suite s, let m_s be its candidate count, N_s its incident count, and N_{sy} the number with true fault y. Incident i supplies a prototype distribution from its observed counts v_{ij}:

$$\rho_{ij} = \frac{v_{ij} + \alpha}{20 + m_s \alpha}, \qquad \eta_s(y) = \frac{N_{sy} + \alpha}{N_s + m_s \alpha}. \tag{5}$$

Here η_s is the fitted class prior. A common pseudocount $\alpha \in [10^{-9}, 1]$ is chosen so that $N^{-1} \sum_i \sum_j \rho_{ij}^{20}$ matches the observed fraction of unanimous incidents, where $N = 165$. We use 60 bisection steps, or the nearest endpoint if the target is outside the attainable range. Positive smoothing gives every possible tally nonzero probability.

To generate a tally, draw a true fault from η_s, choose uniformly among prototypes with that truth, then draw twenty votes from that prototype. For I_{sy}, the incidents in suite s with true fault y, the likelihood of counts $\mathbf{v}$ is

$$P_s(\mathbf{v} \mid y) = \frac{20!}{\prod_j v_j!} \frac{1}{N_{sy}} \sum_{i \in I_{sy}} \prod_j \rho_{ij}^{v_j}. \tag{6}$$

Table 7: Integrated accept-or-escalate loss under the fitted tally model. All policies retain the same leading diagnosis. Lower is better.

Information available to the ideal receiver	Integrated loss
Correctness probability q	.2965
Optimally decoded full tally	.2965
Calibrated leading count + predicted diagnosis	.3127
Calibrated unanimity bit + predicted diagnosis	.3179
Raw leading share	.4015

The joint mass is $J_s(y, \mathbf{v}) = \eta_s(y)P_s(\mathbf{v} \mid y)$, and the tally mass is $P_s(\mathbf{v}) = \sum_y J_s(y, \mathbf{v})$. For its leading diagnosis $\hat{y}(\mathbf{v})$, the decoded correctness is

$$q_s(\mathbf{v}) = \frac{J_s(\hat{y}(\mathbf{v}), \mathbf{v})}{P_s(\mathbf{v})}. \tag{7}$$

The full tally and a directly supplied $q_s(\mathbf{v})$ therefore support the same decisions. For a compressed summary z, its calibrated score is the mass-weighted mean of $q_s(\mathbf{v})$ over tallies with that summary. The leading-count summary is $(\hat{y}, \max_j v_j)$; the unanimity summary is $(\hat{y}, \mathbf{1}\{\max_j v_j = 20\})$. Both include the *predicted* leading diagnosis. Raw leading share uses $\max_j v_j/20$ directly.

We enumerate all 63,756 tallies across the five- and six-candidate suites. For any score $u_s(\mathbf{v})$, the exact mean loss over $c \in [0, 1]$ is

$$\overline{L}(u) = \sum_s \frac{N_s}{N} \sum_{\mathbf{v}} P_s(\mathbf{v}) \left[\frac{(1 - u_s(\mathbf{v}))^2}{2} + (1 - q_s(\mathbf{v}))u_s(\mathbf{v}) \right]. \tag{8}$$

These are expectations under the fitted model. The empirical gate comparison in Appendix J.2 instead uses the recorded diagnoses and their observed correctness.

C Correctness Estimator and Baselines

Configuration and policy definitions. A configuration fixes (k, ϕ, g, π, m, v): response budget, features, fitted predictor, decision policy, note format, and diagnosis visibility. The general feature interface accepts $(r, \hat{y}, E)$; the evaluated three-feature model uses only the token probabilities in r. For a known cost, select by validation loss $L(c)$; for a cost range, declare a weighting and select by the weighted mean. Our saved-data policy comparisons use uniform weighting over $[0, 1]$. Only combinations evaluated on compatible cases are eligible. A tally format must supply the required response collection and include its sampling expense.

Algorithm 2 expands Algorithm 1. The experiments evaluate its components and fixed configurations; evaluation of a jointly selected configuration uses separate new cases. The three policies are *gate only*, *receiver only*, and *gate then receiver*. Both gate policies apply the same threshold to sender correctness. The hybrid's threshold is the specified routing rule; its effect is measured in Appendix J.3.

The note follows the selected format and visibility: a no-diagnosis note omits the explicit conclusion while retaining its other specified content. Receiver C and the hybrid use the same explained-probability note and instructions. Receiver-only always requests a decision; the hybrid requests one only after the case passes the gate. Figure 2 shows these three paths from the shared estimator.

Alternative predictors. The final-answer probability is 1.0 on every RCAEval incident and has AUROC .500. The twenty-answer leading share has AUROC .659 [.577, .737]. In an earlier single-trace collection on the same 165 incidents, the pre-answer mean token gap reaches .658 for predicting correctness of the original twenty-answer leader. Table 1 uses the first response from the later repeated-trace collection and targets that response's own diagnosis, giving .701. These scores refer to different sampled diagnoses. A *probability gap* is the difference between the two most likely next-token probabilities. Token *entropy* describes how spread out the next-token distribution is, rather than which fault wins the final vote.

Algorithm 2 Detailed configuration and execution

Require: Sender S, receiver R, development cases D, separate test cases T
Require: Candidate budgets K, features Φ, predictors G, policies Π, note formats M; cost objective
Offline configuration
Cache sender responses; form each budget's leading diagnosis and correctness label; compute vote summaries.
Fit candidate predictors with whole-system cross-fitting and incident-grouped inner tuning; compare quality and sampling cost.
Compare policies and note formats on matched cases using decision loss $L(c)$.
Separately compare diagnosis inclusion/removal, holding other note content fixed.
Select an evaluated configuration for the stated cost objective; freeze θ.
Evaluate the selected configuration on T without retuning.
Online handoff for case x, evidence E, and cost c
$r \leftarrow \mathrm{Sample}(S, x, k_\theta)$; $\hat{y} \leftarrow \mathrm{LeadingAnswer}(r)$ under the fixed tie rule
$\hat{q} \leftarrow g_\theta(\phi_\theta(r, \hat{y}, E))$
if π_θ includes a gate and $\hat{q} < 1 - c$ **then**
    return `ESCALATE`
end if
if π_θ is gate only **then**
    return $\hat{y}$
end if
return $R(\mathrm{Note}_\theta(E, \hat{y}, \hat{q}, r))$

The three-feature correctness estimator is regularized logistic regression. For response r, let T_r count the logged tokens before its final answer declaration. The raw feature vector for K complete responses is

$$\phi = \frac{1}{K}\sum_{r=1}^{K}\frac{1}{T_r}\sum_{t=1}^{T_r}\left(\log p_{rt}^{\mathrm{chosen}},\ p_{rt}^{(1)} - p_{rt}^{(2)},\ H(\tilde{p}_{rt})\right). \tag{9}$$

The first two ordered probabilities are the largest and second largest logged next-token probabilities. For entropy, $\tilde{p}$ includes the top-five logged alternatives and the chosen token without duplication, plus one bin for all unlisted mass. Logarithms are natural, so entropy $H(\tilde{p}) = -\sum_j \tilde{p}_j \log \tilde{p}_j$ is measured in nats. If the listed probabilities p_j sum to $M > 1$, we use $\tilde{p}_j = p_j/M$ and zero residual mass. Otherwise their residual bucket has probability $1 - M$. All tokens within a response receive equal weight, and all responses receive equal weight regardless of length. Training-fold means and standard deviations transform ϕ into the standardized z used in the logistic map

$$\hat{q} = \frac{1}{1 + \exp[-(\beta_0 + \beta^\top z)]}. \tag{10}$$

The cyclic-window comparison in Table 8 uses the same feature recipe with different numbers of complete responses. Its averages over twenty windows assess response-selection sensitivity. The main comparison in Table 1 uses the fixed first k responses, matching the receiver experiment's first-three configuration at indices $0, 1, 2$.

At every response budget, the model holds out one complete software system across RCAEval's RE1 and RE2 suites. Each outer fold trains on 110 incidents and predicts the remaining 55. Scaling and fitting use only the training data; regularization is selected by five inner folds grouped by incident. All response windows for an incident stay together during tuning, and each incident receives equal total fitting weight. None of the reported budget AUROCs is an in-sample estimate.

C.1 Budget-matched ranking

This exploratory check reuses the frozen out-of-system LR predictions and the same first k traced responses for $k = 1, 3, 5, 20$. Leading share, one minus normalized vote entropy, and the mean top-two token probability gap all target the same selected diagnosis as LR within each budget. We do not fit the baselines or treat their raw scores as probabilities. The fixed groups have 91, 90, 92, and 94 correct diagnoses among 165 incidents; the different targets across budgets preclude a same-target budget test. Table 8 retains the earlier cyclic-window averages, which are distinct from the fixed-group results in Table 1.

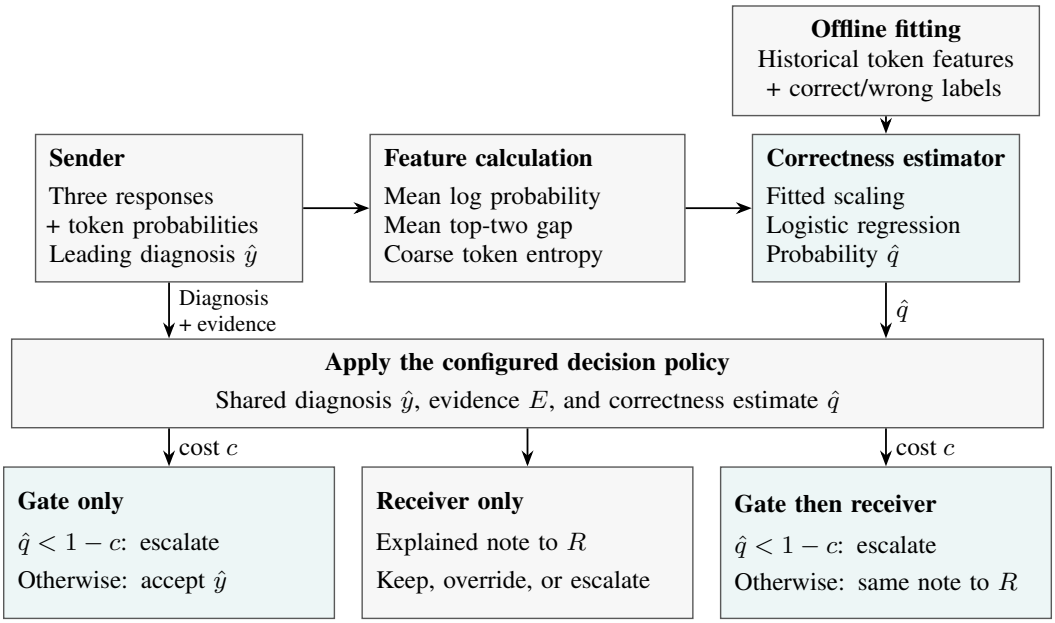

Figure 2: Shared three-response correctness estimator and the three RCAEval decision policies. The receiver note contains the diagnosis, evidence, and explained probability. Both gate policies use cost c; the hybrid uses the same receiver note after the case passes the gate. At $\hat{q} = 1 - c$, gate-only accepts and the hybrid calls the receiver. Diagnosis visibility is compared separately in Section 5.

Table 8: Cyclic-window sensitivity using the same trained feature recipe with each system held out. One, three, and five responses use twenty cyclic windows per incident; twenty uses one complete set. Tuning is confined to the outer training fold. No equivalence margin was specified.

Responses	Diagnosis accuracy	Trained three-feature AUROC
1	55.0%	.703
3	55.4%	.740
5	55.7%	.747
20	57.0%	.747

LR ranks correctness better than the matched vote measures at three and five responses, but no budget establishes an advantage over the token gap. At three responses their AUROCs are .756 and .762, respectively. The twenty-response comparisons with vote measures are also inconclusive. These findings demonstrate token-level error information at small budgets. The fitted gap-only comparison below tests whether the additional two features improve probability estimates. Receiver comparisons continue to use the original three-feature messages.

A one-feature probability predictor. We fit a gap-only LR to the same three-response diagnoses using just the mean top-two token gap. It uses the original twenty cyclic windows per training incident, equal total incident weight, whole-system holdouts, and the same five-fold incident-grouped scaling and regularization selection as the three-feature LR. The gap-only model selects $C = 1$ in each fold. Evaluation uses one fixed-first-three diagnosis per incident; the original three-feature probabilities are unchanged. Separate fitted maps for the held-out systems can change pooled ordering, so gap-only LR and the raw gap can have different pooled AUROCs.

The three-feature-minus-gap-only Brier difference is $-.0023$ [$-.0147$, $+.0108$], with raw $p = .7326$ and Holm $p = 1.000$. The paired log-loss difference is $-.0025$ [$-.0327$, $+.0292$]; the AUROC difference is $+.0039$ [$-.0341$, $+.0408$]. Thus both models provide useful correctness estimates, while the benefit of adding log probability and entropy to the gap remains unresolved. The smaller ECE of the gap-only model is descriptive. By system, three-feature/gap-only Brier is .1725/.1862 for Online Boutique, .2025/.2147 for Sock Shop, and .2262/.2072 for Train Ticket.

Table 9: LR minus unfitted-baseline AUROC on matched response groups. Intervals use 5,000 paired incident bootstraps within system, holding LR predictions fixed. Holm adjustment covers all twelve two-sided centered bootstrap tests. At one response both vote scores are constant.

Responses	Baseline	AUROC difference [95% CI]	Holm p
1	Leading share	$+.2351$ [$+.1556$, $+.3103$]	.00240
1	Vote entropy	$+.2351$ [$+.1556$, $+.3103$]	.00240
1	Mean token gap	$+.0337$ [$-.0081$, $+.0760$]	.50390
3	Leading share	$+.1792$ [$+.0862$, $+.2669$]	.00240
3	Vote entropy	$+.1799$ [$+.0871$, $+.2686$]	.00240
3	Mean token gap	$-.0062$ [$-.0442$, $+.0309$]	1.00000
5	Leading share	$+.1482$ [$+.0520$, $+.2415$]	.01920
5	Vote entropy	$+.1505$ [$+.0531$, $+.2439$]	.01920
5	Mean token gap	$+.0082$ [$-.0278$, $+.0437$]	1.00000
20	Leading share	$+.0935$ [$-.0072$, $+.1914$]	.40792
20	Vote entropy	$+.0829$ [$-.0162$, $+.1801$]	.50390
20	Mean token gap	$+.0016$ [$-.0417$, $+.0433$]	1.00000

Table 10: Three-feature and gap-only LR probability estimates on the same 165 fixed-three diagnoses. AUROC intervals resample incidents within system; lower Brier, log loss, and ECE are better. Raw token-gap values are ranking scores; the gap-only row applies a fitted logistic probability map.

Predictor	AUROC [95% CI]	Brier	Log loss	ECE
Three-feature LR	.7556 [.6773, .8279]	.2004	.5908	.0543
Gap-only LR	.7517 [.6751, .8237]	.2027	.5933	.0292

This exploratory follow-up uses a two-test Holm family: the Brier comparison on all 165 incidents and the paired three-feature-versus-raw-gap AUROC comparison on the original 80 unanimous cases below. Intervals use 5,000 paired system-stratified incident bootstraps. Brier uses 20,000 two-sided sign swaps, and AUROC a centered two-sided bootstrap test. Log loss, all-incident AUROC, ECE, and system splits are secondary. These comparisons assess the predictors on saved responses; the reported receiver effects belong to the original probability-message experiment.

C.2 Additional estimator checks

Five-response sensitivity. Across all $\binom{20}{5} = 15{,}504$ common five-response subsets, the separate frozen mean-gap score has median AUROC .728 with subset P05–P95 [.695, .760] and rank correlation .903 [.882, .922] with its all-twenty score. These are spreads over response subsets, not confidence intervals over independent incidents. At sixty accepted incidents, five-response plurality plus that score leaves a median fourteen [11, 17] wrong. Its expected wrong-commitment-rate difference from the twenty-response disagreement method is -1.8 points [-7.3, $+3.5$]. A rule tuned for 80% accepted accuracy on 34 development incidents reaches 70.9% [61.3, 79.9] on 131 evaluation incidents. This supports a cost–quality compromise, not a fixed accuracy certificate.

Fixed-three calibration. On all 165 incidents, the fixed $0, 1, 2$ triplet has AUROC .756, Brier .200, and five-bin ECE .054. On the 125-incident receiver-study population, the corresponding metrics are .748, .200, and .033. The latter population has 72 correct stated diagnoses and 53 incorrect ones, a base rate of .576. Table 11 gives the system-specific values. The score used for a system excludes that system during fitting.

The five pooled bins have mean predicted/observed correctness $.197/.500$ on two incidents, $.299/.259$ on 27, $.504/.550$ on forty, $.687/.667$ on 33, and $.870/.870$ on 23. The two-incident bin is too small for a reliable bin-specific claim. Combining systems can yield small calibration error even when the errors within each system are larger.

Self-reported confidence baseline. GPT-5.4 makes one additional confidence judgment for each of the 165 fixed three-response diagnoses. It receives the same resource-blind sender background and telemetry, not receiver-only evidence or the true fault. The diagnosis cannot change. The response

Table 11: Calibration on the fixed-three receiver population.

Population	Incidents	AUROC	Brier	ECE
Pooled	125	.748	.200	.033
Online Boutique	44	.858	.165	.092
Sock Shop	38	.703	.215	.081
Train Ticket	43	.660	.223	.095

Table 12: Raw and cross-fitted calibrated self-reported confidence versus the trained fixed-three score. AUROC intervals resample incidents within system. ECE uses five bins; the 125-incident receiver subset is descriptive.

Population	Score	AUROC [95% CI]	Brier	ECE	Mean p
All 165	Trained	.756 [.677, .828]	.200	.054	.556
All 165	Self-report	.717 [.635, .791]	.278	.251	.797
All 165	Calibrated self-report	.703 [.619, .779]	.218	.031	.540
Receiver 125	Trained	.748 [.659, .833]	.200	.033	.570
Receiver 125	Self-report	.706 [.612, .796]	.276	.260	.836
Receiver 125	Calibrated self-report	.689 [.596, .779]	.216	.034	.569

is one JSON percentage, parsed strictly and divided by 100. The primary comparison uses this raw probability without training or calibration. This is a separate assessment of an existing diagnosis, not confidence elicited during its original generation. Other elicitation methods include top-k answer probabilities and a second confidence turn after reasoning (Tian et al., 2023). The full prompt is in Appendix K.1.

A first run used high reasoning with a 2,048-token output cap. Only 25 of 165 requests returned a percentage; the other 140 spent the cap on reasoning and returned no answer. These failures are not zero confidence. A second run raised the cap to 16,384 tokens while keeping the prompt, model, diagnoses, and analysis fixed; all 165 replies parsed on one attempt. The two runs are not combined, and only the complete run is analyzed. Because the second run followed a partly observed first run, it is not a fresh independent confirmation.

Observed correctness is 90/165 (.545) overall and 72/125 (.576) on the receiver panel. The primary family compares trained minus self-reported AUROC and Brier, with Holm correction across the two tests:

The comparison uses 5,000 paired system-stratified incident bootstrap replicates; AUROC uses a centered-bootstrap test, and the Brier test uses 20,000 paired sign swaps. ECE and mean probability are descriptive. The three-feature estimator improves probability error over this raw baseline; the ranking difference remains inconclusive. Self-reported confidence exceeds observed correctness by about 25 points on average, so useful ranking does not make its raw numbers well-calibrated probabilities. The 25 replies from the first run are not analyzed.

Cross-fitted self-report calibration. A zero-call follow-up applies *Platt scaling*, a two-parameter logistic correction, to the 165 saved self-reports. Each of three folds fits the correction on 110 incidents from two complete systems and scores the 55 incidents in the remaining system. No held-out label enters fitting. Probabilities are clipped to $[10^{-6}, 1 - 10^{-6}]$ before the logit transform; regularization uses the fixed $C = 10^6$, without a hyperparameter search. This check was selected

Table 13: Paired trained-minus-self-reported score comparisons on 165 incidents. Higher AUROC and lower Brier are better.

Measure	Difference [95% CI]	Raw p	Holm p
AUROC	$+.0390$ [$-.0659$, $+.1457$]	.4829	.4829
Brier	$-.0779$ [$-.1307$, $-.0257$]	.00445	.00890

after observing raw self-report overconfidence, not independently preregistered. The original raw comparison is unchanged.

Calibrated self-report has Brier .2175, versus .2004 for the trained score (Table 12). The single primary trained-minus-calibrated Brier difference is $-.0171$, with paired 95% interval $[-.0532, +.0190]$ and two-sided $p = .3591$. Thus this comparison does not establish a trained-score advantage; it also does not establish equivalence. The interval uses 5,000 system-stratified incident bootstraps and the test uses 20,000 paired sign swaps. With one primary contrast, no multiplicity correction is needed. Inference conditions on the fitted out-of-fold predictions; it does not refit the calibrator inside each bootstrap replicate.

AUROC, ECE, and the receiver subset are descriptive. The latter Brier difference is $-.0164$ $[-.0565, +.0246]$. Different calibration maps across held-out systems can alter pooled ranking even when each map is monotone. This analysis tests calibrated confidence as a score, not as a new receiver message; the self-report message (arm E below) uses the raw self-report.

External ranking and calibration. Transferred unchanged to Cloud-OpsBench, the three-feature model's ECE degrades to .351. A different, exploratory maximum-combination score takes the maximum of separately trained token, text, and repetition components. It reaches AUROC .760 in development and .812 [.728, .886] externally, against .748 for the transferred three-feature baseline: a paired difference of $+.064$ $[+.015, +.121]$. The combination was chosen after development outcomes and frozen before opening the external cache. It is not the probability sent in the receiver study. On tool-based traces, final-turn chosen log probability, top-two gap, and negative entropy separately reach AUROC .768, .758, and .773.

C.3 Feature selection

After the receiver experiments, we compared seven logistic-regression feature sets to assess whether a richer correctness estimator was warranted. This exploratory comparison did not determine the estimator used in the completed messages. Every model predicts correctness of the same first-three diagnosis on each of 165 incidents; 90 diagnoses are correct. Each outer fold trains on 110 incidents and evaluates the 55 from the remaining system. Training uses the same twenty cyclic three-response windows per incident, with equal total incident weight. Five inner folds grouped by incident fit preprocessing and select regularization by log loss. The original three-feature predictions reproduce exactly. Repeated windows are not independent evaluation cases.

The added diagnosis features are one-hot indicators of the sender's predicted fault, not the true fault. Categories are learned only from training incidents. The agreement features use the same three responses: leading vote share, the fraction of candidate faults receiving a vote, and vote entropy divided by the logarithm of the candidate count. The symptom-free candidate therefore uses the three token features, predicted diagnosis, and three-response agreement, without any symptom inputs or diagnosis–symptom products.

The symptom vector summarizes the six evidence lines already visible in the handoff. Its 46 components describe metric-family frequency and rank, normalized before-to-during change, spread, tail behavior, and aggregate shape summaries. It excludes absolute processor levels, case and service identity, system and suite indicators, observation duration and total measurement count, true-fault inputs, and hand-coded deployment rules. We compare additive symptoms with explicit products of each predicted-fault indicator and each symptom feature. Thus both a model without symptoms and a model with symptoms but without interaction products are tested.

Table 14 reports the point estimates. Primary tests compare each alternative's Brier score with the original estimator, using 5,000 paired incident bootstrap resamples within system and 20,000 two-sided incident sign swaps. Holm correction covers the six Brier comparisons; paired AUROC tests use a separate six-comparison family. Intervals condition on saved out-of-system predictions and do not refit the models inside each bootstrap sample. ECE, log loss, and system-level results are descriptive.

The full interaction model has the best Brier and AUROC point estimates, but neither improvement is established. Its Brier difference is $-.0177$ $[-.0518, +.0172]$ (raw $p = .328$), and its AUROC difference is $+.0367$ $[-.0405, +.1169]$ (raw $p = .371$, Holm $p = 1.000$). It improves Brier on

Table 14: Exploratory LR feature comparison on the same 165 fixed-three diagnoses. "Token" denotes the original three token features; "interactions" denotes predicted-diagnosis indicators multiplied by symptom features. Columns count input features, not independent cases. Higher AUROC and lower Brier, ECE, and log loss are better.

LR inputs	Columns	AUROC	Brier	ECE	Log loss
Token	3	.7556	.2004	.0543	.5908
Diagnosis only	6	.5284	.2617	.1108	.7208
Token + diagnosis	9	.7087	.2176	.0923	.6267
Token + diagnosis + agreement	12	.7166	.2139	.0546	.6176
Token + diagnosis + symptoms, additive	55	.7428	.2022	.0466	.5903
Token + diagnosis + symptoms + interactions	331	.7923	.1827	.0455	.7093
Diagnosis + symptoms + interactions	328	.7684	.1952	.0802	.7352

Table 15: Alternative minus original-estimator Brier score. Negative differences favor the alternative. The 95% intervals are unadjusted; Holm p-values adjust the six Brier tests together. Diagnosis and agreement additions without symptoms do not improve the original model.

Alternative LR inputs	Brier difference [95% CI]	Holm p
Diagnosis only	$+.0613$ [$+.0328$, $+.0897$]	.00090
Token + diagnosis	$+.0172$ [$+.0002$, $+.0344$]	.27074
Token + diagnosis + agreement	$+.0135$ [$-.0062$, $+.0329$]	.68037
Token + diagnosis + symptoms, additive	$+.0018$ [$-.0230$, $+.0254$]	1.00000
Token + diagnosis + symptoms + interactions	$-.0177$ [$-.0518$, $+.0172$]	.98395
Diagnosis + symptoms + interactions	$-.0052$ [$-.0414$, $+.0324$]	1.00000

Online Boutique and Sock Shop but worsens it on Train Ticket (.2427 versus .2262). Its log loss also worsens (.7093 versus .5908). No alternative establishes an improvement over the original three-feature estimator, which we retain; this is not evidence that the alternatives are equivalent.

As a diagnostic control, a separate multiclass LR uses only the symptom vector to predict the fault type. It uses the same outer system holdouts and training-only tuning, with one record per incident. It correctly diagnoses 77/165 incidents (46.7%). Taking its probability for the sender's diagnosis as a correctness score gives AUROC .743 [.668, .815] and Brier .244. These restricted symptoms do not trivially solve the task. Nevertheless, they can teach deployment-specific fault patterns across systems. The symptom models are legitimate supervised, task-informed predictors, not like-for-like measurements of the sender's uncertainty. Excluding explicit products does not remove all diagnostic information from symptoms. The symptom-free models avoid that direct input, though predicted-fault categories can still encode differences in historical correctness. All comparisons reuse previously studied incidents; selecting a replacement would require further validation, and its receiver effect cannot be inferred from the original messages.

C.4 Correctness under unanimous votes

We next ask whether the retained three-feature score distinguishes errors when all twenty votes agree. This exploratory subgroup check uses the original LR's saved out-of-system predictions for response indices $0, 1, 2$, without refitting or selecting a model on this subgroup. The original diagnosis-removal panel contains 80 unanimous incidents: 57 diagnoses are correct and 23 are wrong. All 80 match the diagnosis being scored, so none is excluded. Agreement is 100% on every incident and therefore has AUROC .500.

The primary 80-case comparison gives AUROC .738, a difference of $+.238$ [$+.113$, $+.353$] from constant agreement. The existing centered two-sided bootstrap test gives $p = .00040$, using 5,000 system-stratified incident resamples. This is one primary ranking comparison. The 70 unanimous cases from the earlier 125-case confirmation panel provide a descriptive sensitivity check (Table 16), not a fresh replication. Its corresponding difference is $+.222$ [$+.082$, $+.352$].

The raw mean token gap also separates correct and incorrect diagnoses in the same original 80 cases, with AUROC .7391 [.6160, .8516]. The paired three-feature-minus-gap difference is $-.0008$

Table 16: Correctness ranking within unanimous votes. The original three-feature LR is unchanged; confidence intervals resample incidents within system. The 70-case row is an overlapping sensitivity subset, not an independent confirmation.

Unanimous population	Correct	Wrong	LR AUROC [95% CI]	Agreement AUROC
All 80 incidents	57	23	.738 [.613, .853]	.500
Confirmation-panel subset (70)	49	21	.722 [.582, .852]	.500

Table 17: Warning-detail contrasts on 165 incidents, in percentage points.

Contrast	Wrong commitment	Escalation	Holm p
Full tally − none	-0.1 [-2.3, $+2.1$]	0.0 [-2.2, $+2.2$]	1.000
Full tally − agreement score	$+1.3$ [-0.7, $+3.3$]	-0.8 [-2.8, $+1.2$]	.768
Qualitative − agreement score	$+0.9$ [-0.8, $+2.6$]	-0.4 [-2.1, $+1.3$]	.768

[$-.0670$, $+.0656$] (raw $p = .9878$, Holm $p = 1.000$ in the gap follow-up family described above). The result therefore identifies error information shared by the token-based scores beyond unanimous voting, rather than an advantage unique to the three-feature combination.

On the original 80 incidents, correct diagnoses receive mean LR probability 66.3% (median 67.5%), versus 48.6% (median 49.8%) for wrong diagnoses. The distributions overlap: a wrong unanimous diagnosis receives a score as high as 89.3%. Per-system AUROC is .889 on Online Boutique (26 cases), .750 on Sock Shop (22), and .627 on Train Ticket (32), so performance is not uniform.

Unanimity is selected from the original vote collection, whereas the LR scores come from the separate probability-trace collection. Exact diagnosis matching preserves the target but does not make the collections identical. Both the learned score and raw gap provide correctness ranking that unanimous agreement cannot provide; this does not establish subgroup calibration, perfect error detection, or a validated show/hide threshold. Removing an explicit conclusion also leaves its identity visible in a unanimous tally. No new receiver outcome or adaptive policy is evaluated here.

D Warning-Detail Results

Table 17 reports the original prespecified family. The outcome intervals are simultaneous within each contrast. All 3,960 decisions parse, and commitment remains near universal across arms.

Encoding and position controls. Writing the exact counts in words or digits gives identical observed loss curves. Moving the same warning before the evidence changes wrong commitment by -0.8 points and escalation by 0.0 on the twenty position-test incidents. The stronger receiver prompt supplies all sender measurements, deployment references, permission to disagree, and a defined escalation role, but still does not establish a pooled tally effect. These controls do not prove that every possible wording is ineffective; the synthetic and interpretation-instruction experiments test concrete alternatives.

E Synthetic-Value Experiments

The synthetic experiments measure responsiveness to a statement, not the accuracy of its numerical content. Evidence, diagnosis, and within-wording text are fixed for each incident; changing the displayed number is the experimental intervention. The four small RCAEval panels deliberately cover the three systems and correct and incorrect diagnoses, rather than being random representative samples of deployment traffic. GPT-5.2 supplies three high-reasoning draws per condition. All 1,440 calls across the four RCAEval experiments parse on the first and only attempt. The separate tool-based experiment below uses both receivers and 960 additional calls.

Table 18: Endpoint experiment. Commitment rates are percentages; changes are percentage points with incident-bootstrap intervals. Up/down counts refer to case-majority commitment changes from 0 to 100.

Meaning	Level	At 0	At 100	Change [95% CI]	Up/down
Probability	1	44.4	100.0	$+55.6$ [$+33.3$, $+77.8$]	3/0
Probability	2	61.1	94.4	$+33.3$ [$+11.1$, $+55.6$]	3/0
Probability	3	55.6	100.0	$+44.4$ [$+22.2$, $+66.7$]	3/0
Percentile	1	100.0	94.4	-5.6 [-11.1, 0.0]	0/0
Percentile	2	83.3	72.2	-11.1 [-22.2, 0.0]	0/1
Percentile	3	61.1	83.3	$+22.2$ [-5.6, $+50.0$]	2/1

Table 19: Three-level wording experiment. Rates and changes are percentages and percentage points, respectively. Gaps compare the most cautious with the least cautious displayed setting.

Meaning	Wording	Escalation by setting	Gap [95% CI]	Majority up/down
Probability	Concise	0.0 / 5.6 / 13.9	$+13.9$ [$+2.8$, $+30.6$]	1/0
Probability	Explained	0.0 / 8.3 / 16.7	$+16.7$ [$+2.8$, $+33.3$]	1/0
Uncertainty	Concise	0.0 / 2.8 / 11.1	$+11.1$ [0.0, $+27.8$]	1/0
Uncertainty	Explained	0.0 / 5.6 / 13.9	$+13.9$ [0.0, $+30.6$]	1/0

E.1 Initial endpoint experiment

Six incidents receive synthetic values 0, 50, and 100 under two meanings: probability of correctness and relative reliability percentile. A percentile is a rank relative to other cases, not an expected rate of correctness. Three explanation levels give eighteen conditions per incident and 324 calls. Level one names the measure; level two adds that the value was assigned rather than computed from evidence; level three also explains its relevance to verification or escalation. The full sentences are in Appendix K.6.

Here *commitment* means naming any fault rather than escalating; it does not mean naming the correct fault. Table 18 gives the prespecified endpoint contrasts. The middle value was also run; it is not part of the endpoint contrast.

The continuation rule required at least a thirty-point endpoint gap and upward movement in five of six case-majority decisions. No percentile configuration passed. The probability controls changed aggregate commitment, but only three case majorities. The planned percentile continuation was cancelled. This is evidence of response to extreme literal probabilities, not evidence that the broad-response continuation criterion was met.

E.2 Correctness probability versus direct uncertainty

The next experiment uses twelve different incidents, values 20, 50, and 80, and two explanation lengths for each meaning. Each condition attributes the statement to an external assessment. Concise versions state the measure; explained versions also describe the source and interpretation of its value. All four templates are reproduced in Appendix K.6. The claimed sources in these sentences are part of the fixed experimental wording. The inserted values remain synthetic; no calibrated assessment is being computed for these conditions. This experiment does not explicitly label the number as synthetic in the note. Thirteen arms, including a no-number control, give 468 calls.

For comparison, probability is displayed in decreasing order 80, 50, 20%, while uncertainty is displayed in increasing order 20, 50, 80 out of 100. Both orders move toward greater stated caution. Table 19 reports escalation at those settings.

The frozen continuation rule required a twenty-five-point gap and upward movement in at least eight of twelve case-majority decisions. Every configuration missed it. The difference between the uncertainty and correctness gaps is -2.8 points for both wording lengths, with intervals [-11.1, $+5.6$] and [-13.9, $+8.3$]. Thus the positive probability contrasts are not a significant direct comparison of probability versus uncertainty wording.

Table 20: Five-level check on disjoint incidents. Columns proceed from least to most cautious setting. Probability decreases from 80 to 20%; uncertainty increases from 20 to 80 out of 100. Entries are escalation rates.

Meaning	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5
Correctness probability	0.0%	0.0%	0.0%	0.0%	11.1%
Direct uncertainty	0.0%	0.0%	0.0%	0.0%	0.0%

E.3 Explicit agent-sure wording

A separate twelve-incident panel tests stronger ownership and action instructions. It has a no-number control, three certainty values, and three uncertainty values, for 252 calls. Both versions attribute the assessment to the diagnosing agent and explicitly tell the receiver to consider it when deciding whether to rely on the diagnosis or escalate. The certainty and uncertainty versions explain their directions oppositely; their full text is in Appendix K.6. Escalation is zero in all seven arms. Stronger attribution and an instruction to use the number did not improve responsiveness on this panel.

E.4 Disjoint five-level experiment and continuation criteria

After the earlier outcomes were known, we lowered the endpoint criterion from twenty-five to ten points and selected the earlier explained wording for a further experiment. The revised criterion was chosen after viewing the earlier outcomes; it does not change the earlier experiment's failure to meet its original criterion. The new twelve-case panel is disjoint from all earlier synthetic panels. It reuses the explained sentences verbatim, with probabilities 80, 65, 50, 35, and 20%, or uncertainty levels 20, 35, 50, 65, and 80 out of 100. Together with the no-number control, this produces 396 calls.

The probability endpoint difference is $+11.1$ points [$+2.8$, $+22.2$], passing the revised ten-point rule. Only one of twelve case majorities changes. This is a low-endpoint response, not a broad graded curve. Direct uncertainty remains flat and does not pass.

Initial probability-message pilot. The subsequent GPT-receiver pilot uses actual fixed-three correctness estimates on 36 other cases spanning .074–.941, with no statement versus score, three draws per arm, and 216 calls. In the lowest third the probability message raises escalation by 8.3 points [0.0, $+25.0$], driven by one of twelve case majorities. The middle and high thirds remain at zero; the overall change is $+2.8$ [0.0, $+8.3$]. The pilot misses the ten-point criterion. The later 125-incident two-reader experiment was a separately specified study, not continuation under this failed pilot rule.

E.5 Low probabilities in tool-based handoffs

This exploratory follow-up reuses the eighty Cloud-OpsBench handoff incidents described in Appendix I: 45 Online Boutique and 35 Train Ticket. Each note retains the diagnosis selected by the five cached sender investigations and the tool evidence from its first supporting trajectory. No new sender call is made. We assign 80, 20, and 5% correctness probabilities to every incident under one fixed sentence. The full rendered notes differ only in that numeral: the diagnosis, tool evidence, padding, and option order are unchanged. Pairwise token counts remain within 5%. Percentages that appear in the underlying evidence are not altered.

The values and contrasts were selected after earlier transfer outputs were available, then frozen before these new calls. This is not a new independent confirmation. All three levels are newly collected in an interleaved schedule, including the 80% reference. There is no no-statement arm in this experiment. Each main receiver makes two high-reasoning decisions per incident and value, for 960 calls. All 960 decisions parse on their first and only attempt, with no missing draws or replacements. Every contrast retains eighty incidents.

The assessment's claims about a trained estimator are fixed experimental wording. The percentages are assigned values, not outputs of that estimator. The receiver is not told that the values were assigned. No target-domain calibration is fit or inferred. Appendix K.7 gives the exact assessment and native receiver prompt.

Table 21: Tool-based synthetic probabilities. Entries are incident-mean wrong commitment / escalation rates in percent. The values are assigned, not calibrated estimates. Pooling gives each reader equal weight.

Scope	Assigned 80%	Assigned 20%	Assigned 5%
Pooled	34.06 / 7.81	20.00 / 30.00	16.25 / 43.44
GPT	31.88 / 9.38	29.38 / 12.50	28.13 / 14.38
Grok	36.25 / 6.25	10.63 / 47.50	4.38 / 72.50

Table 22: Low-minus-high tool-based probability comparisons. Differences are percentage points with simultaneous 95% intervals over eighty incidents. Holm correction covers the two contrasts within each scope.

Scope	Contrast	Wrong commitment	Escalation	Raw p	Holm p
Pooled	$20-80$	-14.06 [-20.31, -7.81]	$+22.19$ [$+15.94$, $+28.44$]	.00005	.00010
Pooled	$5-80$	-17.81 [-24.38, -11.25]	$+35.63$ [$+29.06$, $+42.19$]	.00005	.00010
GPT	$20-80$	-2.50 [-8.13, $+3.13$]	$+3.13$ [-2.50, $+8.75$]	.432428	.488376
GPT	$5-80$	-3.75 [-10.63, $+3.13$]	$+5.00$ [-1.88, $+11.88$]	.244188	.488376
Grok	$20-80$	-25.63 [-35.63, -15.63]	$+41.25$ [$+31.25$, $+51.25$]	.00005	.00010
Grok	$5-80$	-31.88 [-42.50, -21.25]	$+66.25$ [$+55.63$, $+76.88$]	.00005	.00010

The two pooled primary contrasts are 20-minus-80 and 5-minus-80, Holm-adjusted together. Each reader has a separate secondary two-test family. We average draws within incident and weight readers equally; 5,000 paired incident bootstraps preserve software-system strata. The two outcome intervals are simultaneous within each contrast, and joint tests use 20,000 paired sign swaps. No result-dependent value, wording, or continuation rule is selected. The two minima reported below are rounded from raw $1/20{,}001$ and Holm-adjusted $2/20{,}001$, not zero.

Both pooled and both Grok contrasts reject after correction. GPT's changes are inconclusive, not proven absent. The point estimates show that Grok supplies most of the pooled response; different significance decisions do not constitute a reader-interaction test. The within-incident number manipulation establishes sensitivity under this sentence, not superiority over other wording or an accurate learned probability. It does not make the diagnosis-removal transfer in Appendix I a positive result.

Pooled majorities combine four observed draws per incident and value; they are descriptive, not additional tests. Grok's escalation rises on 43/80 incidents at 20% and 65/80 at 5%, with no decreases. The response is therefore not confined to one or two incidents.

The extra specialist requests still have a cost. Pooled point loss differences are $\widehat{L}_{20}(c) - \widehat{L}_{80}(c) = -.140625 + .221875c$ and $\widehat{L}_{5}(c) - \widehat{L}_{80}(c) = -.178125 + .356250c$. Their break-even escalation costs are .634 [95% bootstrap interval .461, .795] and .500 [.355, .642]. These are experimental trade-offs, not certified operating thresholds or utility gains from calibrated predictions. Grok's corresponding break-even costs are .621 [.467, .766] and .481 [.349, .609]; GPT's escalation intervals do not locate a break-even cost.

Table 23: Descriptive incident changes in the tool-based synthetic experiment. Up/down/unchanged compare mean escalation within incident. Changed majorities count only pairs with a unique most frequent decision in both conditions; tied pairs are listed separately.

Scope	Contrast	Up/down/unchanged	Changed / untied	Tied pairs
Pooled	$20-80$	40 / 1 / 39	12 / 60	20
Pooled	$5-80$	66 / 1 / 13	15 / 49	31
GPT	$20-80$	9 / 6 / 65	3 / 60	20
GPT	$5-80$	8 / 3 / 69	5 / 59	21
Grok	$20-80$	43 / 0 / 37	23 / 60	20
Grok	$5-80$	65 / 0 / 15	41 / 56	24

Table 24: Probability-message contrasts. Values are percentage-point differences with simultaneous 95% intervals over incidents.

Scope	Contrast	Wrong commitment	Escalation	Holm p
Pooled	C − A	−5.2 [−8.0, −2.4]	+4.8 [+2.0, +7.6]	.0005
Pooled	C − B	−2.9 [−6.7, +0.8]	+4.8 [+1.1, +8.5]	.0135
GPT	C − A	−5.3 [−9.1, −1.6]	+3.7 [0.0, +7.5]	.0170
GPT	C − B	−0.8 [−5.9, +4.3]	+3.7 [−1.3, +8.8]	.1468
Grok	C − A	−5.1 [−8.3, −1.9]	+5.9 [+2.7, +9.1]	.0002
Grok	C − B	−5.1 [−8.5, −1.6]	+5.9 [+2.4, +9.3]	.0004

Table 25: Pooled probability-message effects by the original fixed score terciles. Differences are percentage points with simultaneous 95% intervals. These stratified estimates are secondary.

Tercile	Contrast	Wrong commitment	Escalation
Low	C − A	−11.9 [−18.7, −5.2]	+12.3 [+5.6, +19.0]
Low	C − B	−13.5 [−21.4, −5.6]	+12.3 [+4.4, +20.2]
Middle	C − A	−2.4 [−4.8, 0.0]	+2.0 [−0.4, +4.4]
Middle	C − B	+2.4 [−2.4, +7.1]	+2.0 [−2.8, +6.7]
High	C − A	−1.2 [−3.7, +1.2]	0.0 [−2.4, +2.4]
High	C − B	+2.4 [−1.6, +6.5]	0.0 [−4.1, +4.1]

F Probability-Message Results

The probability-message experiment made 2,250 one-attempt calls: 125 incidents, three arms, two readers, and three draws. All decisions parsed. Arm A contains no assessment, B contains the exact tally, and C contains the probability message. The pooled tests were fixed before those calls. Table 24 gives both the overall and per-reader contrasts. Holm correction covers C-minus-A and C-minus-B within each reported family.

Pooled wrong-commitment rates are 43.6% for A, 41.3% for B, and 38.4% for C. Escalation is 0.0% for A and B and 4.8% for C. The C-minus-B joint contrast rejects, but its wrong-commitment interval includes zero. A joint rejection should not be read as separate evidence that both outcomes changed. Against A, Equation 3 gives a point break-even cost of about 1.08 and lower estimated loss over $c \in [0, 1]$, without establishing uniform superiority. Against B, the point break-even cost is $c = .611$, with interval $[-.206, 1.303]$; that interval does not identify a reliable operating cost.

Terciles divide this population by rank. Recomputing them for a stronger sender would still assign roughly one third of incidents to the lowest group, even if its scores were higher overall. The comparable deployment quantity is the share below the original low-score boundary, about .471. That share depends on the new score distribution and calibration; it cannot be inferred from a sender's aggregate accuracy alone.

Direct receiver interaction. We compare the per-incident treatment effects between Grok and GPT, rather than comparing their separate significance decisions. Against no statement, the difference-in-differences is $+0.3$ points $[-4.3, +4.8]$ for wrong commitment and $+2.1$ $[-2.4, +6.7]$ for escalation, joint $p = .4959$. Against the tally, it is -4.3 $[-8.5, 0.0]$ and $+2.1$ $[-2.1, +6.4]$, joint $p = .0730$, Holm $p = .1460$. Neither interaction rejects. These contrasts do not establish that the two receivers respond differently to a correctness-probability message.

F.1 Leading-share and self-reported-confidence messages

Two follow-up messages use the same 125 incidents, evidence, fixed-three diagnosis, and option-order seeds. Arm D shows the twenty-response leader's percentage without the full tally, with an explanation that vote frequency is not correctness probability. Arm E shows the raw self-reported confidence from the complete run, rounded to the nearest integer percentage and described as self-reported, not calibrated. Appendix K.3 gives both exact messages. All five notes are pairwise token-matched within 5%.

Table 26: Absolute incident-mean rates for the simple numerical baselines. Entries are wrong commitment / escalation in percent; pooling gives each reader equal weight within incident. A/B/C reuse earlier decisions.

Scope	A: none	B: tally	C: probability	D: share	E: self-report
Pooled	43.60 / 0.00	41.33 / 0.00	38.40 / 4.80	41.87 / 0.13	41.60 / 0.80
GPT	45.33 / 0.00	40.80 / 0.00	40.00 / 3.73	41.87 / 0.00	41.87 / 0.53
Grok	41.87 / 0.00	41.87 / 0.00	36.80 / 5.87	41.87 / 0.27	41.33 / 1.07

Table 27: Simple-baseline contrasts in percentage points with simultaneous 95% coefficient intervals. Holm covers five joint tests in the pooled primary family and five within each secondary reader family.

Scope	Contrast	Wrong commitment	Escalation	Raw p	Holm p
Pooled	D − A	−1.7 [−3.7, +0.3]	+0.1 [−1.9, +2.1]	.1341	.2682
Pooled	D − B	+0.5 [−0.5, +1.6]	+0.1 [−0.9, +1.2]	.5208	.5208
Pooled	C − D	−3.5 [−6.8, −0.1]	+4.7 [+1.3, +8.0]	.0053	.0265
Pooled	E − A	−2.0 [−3.6, −0.4]	+0.8 [−0.8, +2.4]	.0254	.0762
Pooled	C − E	−3.2 [−6.0, −0.4]	+4.0 [+1.2, +6.8]	.0072	.0288
GPT	D − A	−3.5 [−6.4, −0.5]	0.0 [−2.9, +2.9]	.0447	.1788
GPT	D − B	+1.1 [−0.8, +2.9]	0.0 [−1.9, +1.9]	.5034	.5034
GPT	C − D	−1.9 [−6.1, +2.4]	+3.7 [−0.5, +8.0]	.0992	.2977
GPT	E − A	−3.5 [−6.4, −0.5]	+0.5 [−2.4, +3.5]	.0236	.1180
GPT	C − E	−1.9 [−5.9, +2.1]	+3.2 [−0.8, +7.2]	.1267	.2977
Grok	D − A	0.0 [−1.9, +1.9]	+0.3 [−1.6, +2.1]	1.0000	1.0000
Grok	D − B	0.0 [−0.8, +0.8]	+0.3 [−0.5, +1.1]	1.0000	1.0000
Grok	C − D	−5.1 [−8.3, −1.9]	+5.6 [+2.4, +8.8]	.0006	.0030
Grok	E − A	−0.5 [−1.9, +0.8]	+1.1 [−0.3, +2.4]	.3247	.9741
Grok	C − E	−4.5 [−8.0, −1.1]	+4.8 [+1.3, +8.3]	.0036	.0146

Before these reader calls, we fixed five contrasts: D-minus-A, D-minus-B, C-minus-D, E-minus-A, and C-minus-E. A, B, and C retain their definitions above and reuse their original decisions unchanged, so the controls are not contemporaneous and receiver drift over time is unmeasured. The two readers each make three high-reasoning decisions per new note, for 1,500 new requests. One GPT request in arm D failed and was not retried; that incident keeps its other two draws. All other 1,499 replies parse, and every contrast contains 125 paired incidents.

Inference uses 5,000 system-stratified paired incident bootstraps and 20,000 joint sign swaps. The pooled C-minus-D and C-minus-E contrasts reject. D-minus-A and E-minus-A do not reject after correction; this is not equivalence with no statement. Grok's two direct probability-message contrasts reject, while GPT's do not. That difference in significance is not a test of a reader interaction.

Table 28: Prespecified descriptive sensitivity on the 117 incidents whose fixed-three diagnosis equals the twenty-response leader. Differences are percentage points with simultaneous 95% intervals; p values are unadjusted.

Scope	Contrast	Wrong commitment	Escalation	Raw p
Pooled	D − A	−1.3 [−3.3, +0.7]	0.0 [−2.0, +2.0]	.2859
Pooled	D − B	+0.4 [−0.6, +1.4]	0.0 [−1.0, +1.0]	.6853
Pooled	C − D	−3.4 [−6.7, −0.1]	+4.7 [+1.4, +8.0]	.0030
GPT	D − A	−3.1 [−6.0, −0.3]	0.0 [−2.8, +2.8]	.0811
GPT	D − B	+0.9 [−1.1, +2.8]	0.0 [−2.0, +2.0]	.6853
GPT	C − D	−2.0 [−6.6, +2.6]	+4.0 [−0.6, +8.5]	.0793
Grok	D − A	+0.6 [−1.1, +2.3]	0.0 [−1.7, +1.7]	1.0000
Grok	D − B	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]	1.0000
Grok	C − D	−4.8 [−8.3, −1.4]	+5.4 [+2.0, +8.8]	.00085

Table 29: Diagnosis removal on the held-out population. Differences are percentage points with simultaneous 95% intervals. Holm covers the three confirmation strata.

Population	Wrong commitment	Escalation	Holm p
All 125 incidents	-5.7 [-9.2, -2.3]	$+11.9$ [$+8.4$, $+15.3$]	.0001
55 divided	-13.0 [-18.8, -7.3]	$+27.0$ [$+21.2$, $+32.7$]	.0001
70 unanimous	No change	No change	

Table 30: Exploratory receiver split for diagnosis removal. Values are percentage-point changes with intervals over incidents.

Receiver and population	Wrong commitment	Escalation
GPT, all 125	$+1.3$ [-0.8, $+3.5$]	$+0.3$ [-1.9, $+2.4$]
Grok, all 125	-12.8 [-19.5, -6.1]	$+23.5$ [$+16.8$, $+30.1$]
GPT, 55 divided	$+3.0$ [-1.2, $+7.3$]	$+0.6$ [-3.6, $+4.8$]
Grok, 55 divided	-29.1 [-40.6, -17.6]	$+53.3$ [$+41.8$, $+64.8$]

These contrasts compare complete messages: score source, numerical values, and explanation change together. Arm E uses raw self-reports; the fitted self-reports in Appendix C.2 are evaluated as predictors. The receiver results apply to the messages as tested. Their point loss differences are $\widehat{L}_C(c) - \widehat{L}_D(c) = -.03467 + .04667c$ and $\widehat{L}_C(c) - \widehat{L}_E(c) = -.032 + .040c$. The corresponding point break-even costs are .743 and .800, so lower wrong commitment must still justify the extra specialist cost. These are not statistically established operating thresholds or uniform advantages.

G Diagnosis Removal, Instructions, and Decoding

Discovery and confirmation. On the forty-incident discovery panel, diagnosis removal changes wrong commitment by -10.8 points and escalation by $+21.7$. Table 29 gives the three prespecified confirmation contrasts on the disjoint 125-incident set. Removing a diagnosis does not remove its name from a unanimous tally, which helps explain why that stratum is unchanged without making the null inevitable.

Kimi's prospective check on the same 125 incidents changes wrong commitment by -0.3 points and escalation by $+1.6$; the joint test does not reject. The repeated Grok pattern supports a bounded receiver-specific finding. Failure to reject for another receiver does not prove exact insensitivity.

Second sender. GPT-4.1 returns 3,299 parsed answers from 3,300 one-attempt requests, leaving 164 complete incidents. It is divided on 96, passing the forty-case continuation criterion. Its majority accuracy is 48.8%, versus 55.5% for GPT-5.4 on those same cases: -6.7 points [-14.0, $+0.6$]. All 1,968 planned reader decisions parse on the first attempt. Table 31 separates the pooled confirmation from the exploratory strata and receiver rows.

With the diagnosis visible, GPT copies it on 463 of 492 draws and Grok on all 492. After removal, GPT gives 485 overrides and seven escalations; Grok gives 352 overrides and 140 escalations. The sender changes, but the prominent response modes recur.

Table 31: Diagnosis removal with the second sender, GPT-4.1. Differences are percentage points with simultaneous 95% intervals.

Population	Wrong commitment	Escalation	Joint p
All 164, pooled	-7.8 [-11.2, -4.5]	$+14.8$ [$+11.5$, $+18.2$]	$< .0001$
96 divided	-13.4 [-18.4, -8.3]	$+25.3$ [$+20.3$, $+30.4$]	$< .0001$
68 unanimous	No change	No change	1.0000
GPT, all 164	$+0.2$ [-2.4, $+2.8$]	$+1.2$ [-1.4, $+3.9$]	.4326
Grok, all 164	-15.9 [-22.2, -9.6]	$+28.5$ [$+22.2$, $+34.8$]	$< .0001$

Table 32: Escalation under R1 with the rich tally: ten unanimous controls and twenty divided incidents, with three draws per incident. The gap is unanimous minus divided escalation, in percentage points.

Receiver	Unanimous	Divided	Gap [95% CI]
GPT	0.0%	8.3%	-8.3 [-20.0, 0.0]
Grok	0.0%	81.7%	-81.7 [-91.7, -71.7]

Table 33: Exact recovery from real decoder notes. The percentage measures recovery of the named diagnoses or detection of disagreement, not the receiver's diagnostic accuracy.

Decoder	Note	Diagnoses recovered	Division detected
GPT	Honest	100.0%	100.0%
GPT	Concealed	64.6%	81.5%
Grok	Honest	100.0%	100.0%
Grok	Concealed	20.6%	23.8%

Cost and later disclosure. The diagnosis-removal break-even cost is $c^* = .483$ with bootstrap interval [.305, .667]. Figure 3 shows the point loss curves. The interval quantifies uncertainty about the break-even cost; it does not certify a loss advantage at every cost below its lower endpoint.

On twenty divided incidents, each receiver first decides from a note without the diagnosis and then sees the diagnosis in the same conversation. Across 120 paired conversations, 37 decisions move toward the revealed diagnosis and none move away. Wrong commitment rises by 14.2 points [3.3, 25.0], escalation falls by 25.8 points [-36.7, -15.0], and the joint test gives $p = .0002$. The later behavior is indistinguishable from seeing the diagnosis initially under this test. This is an exploratory two-turn result, not a complete explanation of the one-turn intervention.

Instructions for interpreting disagreement. The instruction study distinguishes knowing what a tally means from diagnosing the incident correctly. R0 is the unbriefed instruction. R1 explains that a split tally means ambiguity and that agreement does not guarantee correctness. R2 adds telemetry background and confusable fault pairs. R3 also adds deployment reference values. The complete R1 explanation and R2/R3 additions are reproduced in Appendix K.4. Pooled diagnosis accuracy is 41.7% after R1, 40.0% after R2, and 29.2% after R3. The observed policy change is not an improvement in diagnostic competence. A single unanimous-or-divided bit approximately matches the full tally's behavioral effect.

The divided rates correspond to R1/B in Table 34: five of sixty GPT draws and 49 of sixty Grok draws escalate. The unanimous controls are additional cases, not part of that sixty-draw response-mode table.

Decoding. The decoder receives a note and is asked to estimate the tally, with counts summing to twenty. An honest note states disagreement and names the diagnoses still in play; a concealed note omits that disagreement. These are decoder conditions, not the four main warning-detail formats. Both decoders first pass synthetic-note checks of support and disagreement recovery. Table 33 reports the real-note results. The full order of exact vote counts cannot be reconstructed from a note that never states it.

H Receiver Response Modes

We classify each decision as copied, escalated, or overrode as defined in Section 2. Each row in Table 34 contains sixty draws, from twenty divided incidents with three draws each. Here A omits the tally and B includes it; R0–R3 refer to the instruction levels described above. "Correct override" is a subset of overrides, not a fourth response mode.

Grok's large escalation response occurs with the explained tally. GPT's overrides increase in both the tally and no-tally conditions. This contrast supports a distinction between deferral and contesting the

Table 34: Exploratory response-mode counts on divided incidents.

Receiver	Instructions / arm	Copied	Escalated	Overrode	Correct override
GPT	R0 / A	51	0	9	3
GPT	R1 / A	50	3	7	3
GPT	R2 / A	44	5	11	6
GPT	R3 / A	28	11	21	8
GPT	R0 / B	59	0	1	0
GPT	R1 / B	54	5	1	1
GPT	R2 / B	47	8	5	2
GPT	R3 / B	35	9	16	6
Grok	R0 / A	60	0	0	0
Grok	R1 / A	59	0	1	0
Grok	R2 / A	43	0	17	3
Grok	R3 / A	36	5	19	3
Grok	R0 / B	60	0	0	0
Grok	R1 / B	11	49	0	0
Grok	R2 / B	4	56	0	0
Grok	R3 / B	8	51	1	0

diagnosis. A loss curve still matters, but cannot distinguish two different ways of arriving at a wrong commitment.

I Transfer and Competence Checks

Receiver-competence criterion. On forty disjoint Cloud-OpsBench cases, each receiver diagnoses with or without three retrieved examples of successful investigations. The planned expertise interaction requires at least a ten-point gain and an interval above zero. GPT gains 4.2 points [-3.3, $+12.5$] and Grok 7.5 [≈ 0, $+15.8$]. Neither passes, so the expertise-by-warning experiment is cancelled. We do not relabel the example-augmented receiver as an expert after that result.

Tool-based transfer. In Cloud-OpsBench, GPT-5.4 gathers evidence through diagnostic tools and generates five trajectories per case. From ninety fresh case groups, a fixed rule selects fifty divided cases and thirty unanimous controls without using ground-truth correctness. Both receivers read the diagnosis-present and diagnosis-removed notes twice. This changes evidence acquisition, visible context, and the answer space while remaining within cloud diagnosis.

On the fifty divided cases, removing the diagnosis changes wrong commitment by $+2.5$ points [-4.0, $+9.0$] and escalation by -6.0 [-12.5, $+0.5$], joint $p = .1057$. A prespecified tie-excluding sensitivity has the same qualitative direction, and the thirty unanimous controls are unchanged. The test does not establish either replication or reversal of the RCAEval effect.

During this run a provider outage interrupted 138 reader requests, which were re-issued once; the failed attempts are retained, and all sender replies and tool observations reproduce from saved records. This one-time recovery differs from the one-attempt, no-retry policy of the synthetic and probability-message experiments.

Second-domain feasibility. SteerBench-Work provides 106 scenarios. Provider refusals on one leave 105 complete sender cases, only eleven of which have more than one distinct sampled answer. The continuation criterion required 26 divided cases, so the reader stage does not run. The observed absence of the required disagreement prevents a test of the handoff effect in that domain; it is not a negative reader result.

J Analysis Conventions

J.1 Operating-cost audit

Table 35 reconstructs sender inference budgets from the same first k successful traces on each of 165 incidents. Input/output totals for the complete 3,300-call collection are 5,466,000 and 626,685; no reasoning tokens are reported. At the common illustrative prices its collection cost is $13.09935. Three minus twenty calls costs $-\$.06754$ per incident, with paired system-stratified bootstrap interval $[-\$.06825, -\$.06686]$, an 85.07% sender-only reduction. These prices are not provider invoices, and sums of recorded request seconds are not concurrent wall-clock latency. Labeling and LR-fitting costs were not measured; the full collection is an offline expense rather than a per-incident requirement at deployment.

Each row of Table 36 summarizes 375 usable one-attempt responses. GPT completion usage includes reasoning; Grok reports it separately, so it is added once. Probability messages generate longer receiver outputs here, illustrating why sender-call savings are not full-system savings. The diagnosis and tally controls come from the historical collections described in Appendix A.1; the table is not a fresh end-to-end trial. The specialist cost ratio in Equation 1 is an assumed decision cost, not a measured monetary price. No specialist resolution is observed.

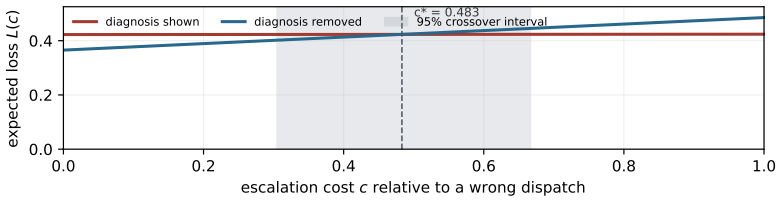

Figure 3: Diagnosis-removal cost comparison on 125 held-out incidents. The vertical marker is the estimated break-even escalation cost .483; shading is its 95% interval [.305, .667]. Removing the diagnosis has lower estimated loss to the left, and retaining it to the right.

J.2 Cost-threshold gate on saved predictions

Policy and population. This exploratory comparison applies Proposition 1 to the 125 incidents in the original probability-message experiment. Each incident retains its first-three sender diagnosis and its original whole-system cross-fitted three-feature LR probability. The 125 diagnoses include 72 correct and 53 incorrect answers. We verify that the diagnosis, score, and true fault match across the saved predictor, note, and receiver records. The gate accepts that diagnosis when $\hat{q} \geq 1 - c$, including equality, and otherwise escalates. It cannot name a different fault. Scores are unrounded; no model is refitted, threshold optimized, or new response collected.

The controls are the completed original A (no warning), B (full tally), and C (correctness probability) conditions. Each has two receivers and three responses per incident: 750 decisions per arm, 2,250 in total. We average the three responses within incident and receiver, then weight the two receivers equally. The independent unit remains the incident. The gate receives numerical c, whereas the recorded receivers received qualitative cost instructions. We reprice their fixed decisions as c varies;

Table 35: Mean recorded usage for the fixed first k sender calls per incident. Seconds sum individual request times, not parallel wall time. Input/output token counts are API-reported; prices are illustrative.

Calls	Input tokens	Output tokens	Seconds	Cost ($)
1	1,656	190	4.25	.00397
3	4,969	564	12.57	.01185
5	8,282	949	21.25	.01984
20	33,127	3,798	84.90	.07939

Table 36: Mean usage for one receiver reply in the probability-message study, not three evaluation repeats. Inputs are locally tokenized prompts, excluding transport framing; output usage is API-reported. Costs use the same illustrative prices for both readers.

Receiver	Message	Input tokens	Billed output	Seconds	Cost ($)
GPT	None	535.52	312.32	7.21	.003793
GPT	Tally	539.51	239.93	5.77	.003074
GPT	Probability	533.08	375.05	8.39	.004417
Grok	None	535.52	647.85	5.85	.007148
Grok	Tally	539.51	546.62	5.06	.006141
Grok	Probability	533.08	775.21	6.33	.008418

this comparison measures a cost-aware policy against the observed handoffs, with their ability to override the sender retained.

Loss and inference. We evaluate $L(c)$ at increments of .001 over $[0, 1]$ and use its exact uniform-cost integral as the primary summary. For diagnosis $\hat{y}_i$, true fault y_i, and correctness estimate $\hat{q}_i$, the gate's mean is

$$\overline{L}_{\text{gate}} = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{(1 - \hat{q}_i)^2}{2} + \hat{q}_i \, \mathbf{1}\{\hat{y}_i \neq y_i\} \right]. \tag{11}$$

Fixed receiver decisions integrate to $w + e/2$. Uniform weighting summarizes sensitivity to cost; it does not estimate a deployment's cost distribution. The calculation assigns zero loss to a correct diagnosis, unit loss to a wrong diagnosis, and c to escalation, separately from API charges or any observed specialist resolution.

The three primary tests compare paired integrated losses for the pooled gate-minus-A/B/C contrasts. We use 5,000 incident bootstrap replicates within software system, 20,000 two-sided incident sign swaps, and Holm correction across the three tests. The bootstrap and sign-swap seeds are 20260814 and 20260809. Each receiver has a separate secondary three-test family. All inference conditions on the saved predictions. Paired 95% intervals at $c = .1, .25, .5, .75, 1$ are pointwise summaries, not simultaneous intervals across costs or arms. The rule, cost grid, and test families were fixed for this follow-up before computing its effects; the incidents had already been used in the receiver studies.

Table 37: Uniformly averaged decision loss on the same 125 incidents. The gate's mean is .311973 in every row. Negative paired differences favor the gate. Holm correction uses three comparisons within each scope; the pooled scope is primary. Brackets give paired 95% intervals.

Scope	Arm	Receiver loss	Gate − receiver [95% interval]	Holm p
Pooled	A	.4360	−.1240 [−.1638, −.0835]	.000150
Pooled	B	.4133	−.1014 [−.1432, −.0595]	.000150
Pooled	C	.4080	−.0960 [−.1334, −.0596]	.000150
GPT	A	.4533	−.1414 [−.1933, −.0911]	.000150
GPT	B	.4080	−.0960 [−.1412, −.0513]	.000300
GPT	C	.4187	−.1067 [−.1573, −.0577]	.000300
Grok	A	.4187	−.1067 [−.1481, −.0643]	.000150
Grok	B	.4187	−.1067 [−.1499, −.0644]	.000150
Grok	C	.3973	−.0854 [−.1201, −.0507]	.000150

Integrated and cost-specific results. The gate achieves mean loss .311973, compared with .436000, .413333, and .408000 for A, B, and C (Table 37). Its 23.5% relative reduction against C is a reduction in decision loss, not an increase in diagnosis accuracy. The direction holds for both receivers separately. Using the integer percentages displayed in the C notes gives gate mean loss .311947, a negligible descriptive rounding change.

Figure 4 and Table 38 show that the improvement is largest when escalation is inexpensive. At $c = .75$, C has slightly lower point loss than the gate (.420 versus .424), with a difference interval

spanning zero. At $c = 1$, the gate accepts all 125 sender diagnoses, with wrong-diagnosis rate .424. B has lower point loss .4133; the gate-minus-B interval $[-.0067, +.0307]$ includes zero. At $c = 0$, all 125 gate decisions escalate at zero assigned cost. These curves support a lower uniform-cost mean without establishing an advantage at every cost or a validated operating cutoff.

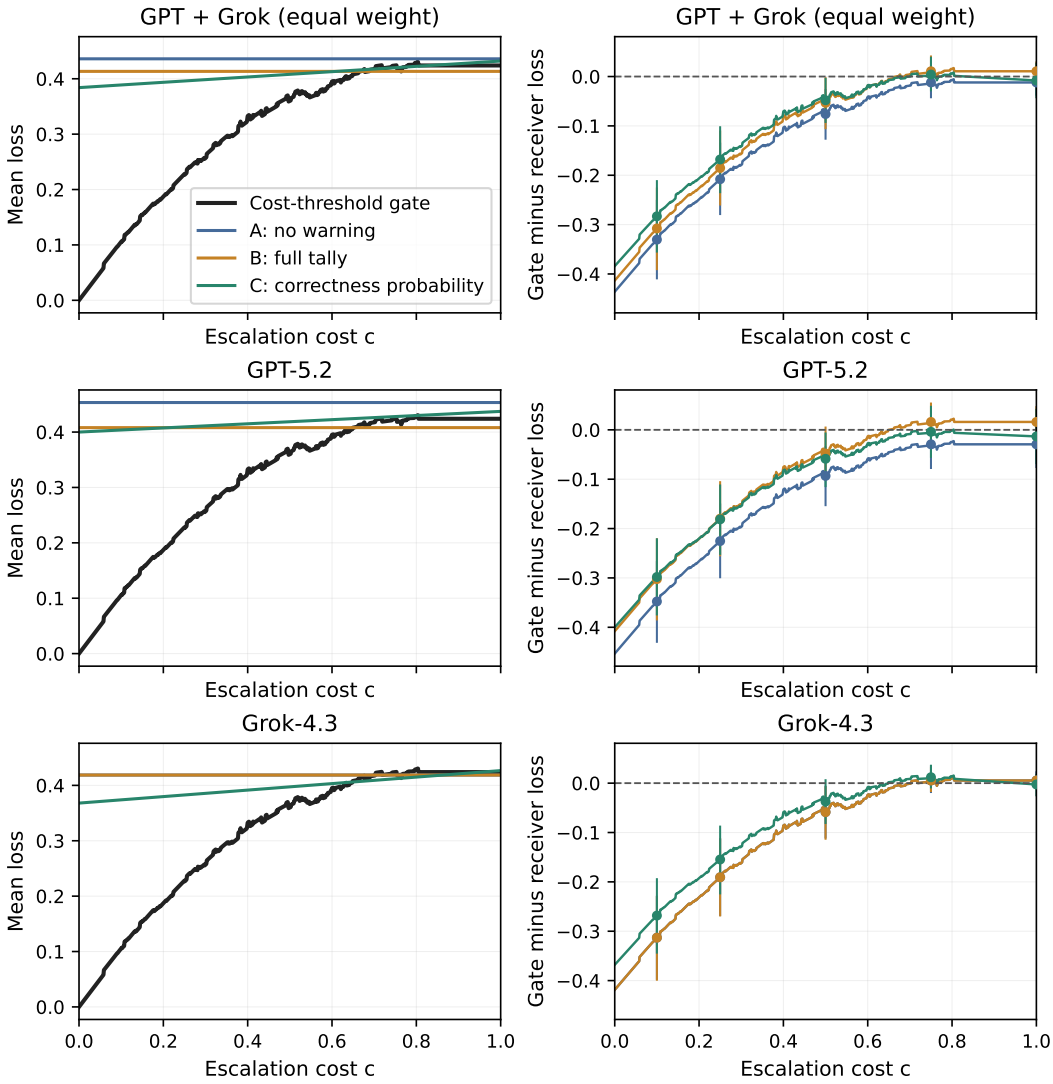

Figure 4: Cost-aware gate versus saved A/B/C receiver decisions on 125 paired incidents. Rows show pooled, GPT, and Grok results. Left: loss as escalation cost varies, with a wrong diagnosis costing one. Right: gate-minus-receiver loss; negative values favor the gate. Error bars are pointwise paired 95% intervals at five fixed costs. Receiver actions remain fixed; only the gate's action changes with c.

Observed outcomes. At $c = .5$, the gate accepts when estimated correctness is at least 50%. It accepts 77 of the 125 diagnoses: 56 are correct and 21 wrong; it escalates the other 48. Table 39 reports observed outcomes as shares of all decisions. Correct, wrong, and escalated are separate outcomes that sum to 100% within each policy. The gate's 44.8% correct and 16.8% wrong rates describe its immediate answers; the remaining 38.4% of incidents escalate. C's percentages average 750 responses on the same 125 incidents. After charging for the extra escalations, gate loss is .360 versus C's .408, an 11.8% reduction at this cost.

Receiver corrections. Receivers can also repair a wrong sender diagnosis by choosing the true fault, an action unavailable to the gate. In C, GPT and Grok make 20 and 6 such repairs among their

Table 38: Loss at fixed escalation costs, pooled across receivers. The final column gives the gate-minus-C difference and its pointwise 95% interval. A is no warning, B the full tally, and C the probability message.

c	Gate	A	B	C	Gate − C [95% interval]
.10	.1056	.4360	.4133	.3888	$-.2832$ [$-.3567$, $-.2099$]
.25	.2280	.4360	.4133	.3960	$-.1680$ [$-.2363$, $-.1007$]
.50	.3600	.4360	.4133	.4080	$-.0480$ [$-.0940$, $-.0027$]
.75	.4240	.4360	.4133	.4200	$+.0040$ [$-.0300$, $+.0400$]
1.00	.4240	.4360	.4133	.4320	$-.0080$ [$-.0440$, $+.0280$]

Table 39: Observed outcomes at $c = .5$. Counts for the gate are incidents; counts for C are recorded responses from two receivers with three draws each on the same incidents. Each percentage uses all decisions in its column as the denominator, including escalations.

Outcome	Gate count	Gate (%)	C responses	C (%)
Correct diagnosis	56	44.8	426	56.8
Wrong diagnosis	21	16.8	288	38.4
Escalation	48	38.4	36	4.8
Total	125	100.0	750	100.0

respective 375 responses; they also change correct sender diagnoses into wrong commitments 20 and 2 times (Table 40). Counts are descriptive repeated draws; distinct affected incidents appear in parentheses. Thus the gate's lower integrated loss coexists with some useful receiver overrides. A comparison with receivers given the same numerical cost would isolate policy differences under matched cost information.

Table 40: Receiver overrides relative to the fixed sender diagnosis. A repair changes a wrong diagnosis to the true fault; a break changes a correct diagnosis to a wrong commitment. Each arm has 375 responses per receiver. Parentheses give the number of distinct affected incidents.

Receiver	Arm	Repairs (incidents)	Breaks (incidents)
GPT	A	11 (5)	22 (9)
GPT	B	10 (5)	4 (2)
GPT	C	20 (7)	20 (11)
Grok	A	3 (1)	1 (1)
Grok	B	2 (1)	0 (0)
Grok	C	6 (5)	2 (2)

J.3 Gate followed by receiver review

The hybrid applies the same threshold to the same 125 saved scores. If $\hat{q} < 1 - c$, it escalates; otherwise it uses the original C receiver decision, including any override or escalation. The note and instructions are unchanged. This asks whether receiver review adds value after a diagnosis passes the gate. Both gate policies use the same automatic threshold, with no new model calls, refitting, or threshold selection.

For incident-level receiver rates w_i, e_i and $a_i = 1 - \hat{q}_i$, the hybrid's exact uniform-cost mean loss is

$$\overline{L}_{\text{hybrid}} = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{a_i^2}{2} + w_i \hat{q}_i + \frac{e_i(1 - a_i^2)}{2} \right]. \tag{12}$$

The primary test compares pooled hybrid and gate-only integrated losses. Five secondary contrasts form a separate Holm family: pooled hybrid minus C, and hybrid minus gate-only and C for each receiver. We use the same 5,000 system-stratified paired incident bootstraps and 20,000 sign swaps as the gate analysis. The rule and comparisons were specified for this follow-up before computing its effects; scores and responses remain fixed.

The hybrid's mean loss is .3179, compared with .3120 for gate-only and .4080 for receiver C. Both gate-first policies improve on the recorded receiver-only handoff. The primary hybrid-minus-gate difference is $+.0059$ [$-.0085$, $+.0210$], $p = .4438$, so receiver review does not show an added integrated-loss benefit over the direct gate (Table 41). Separate receiver results retain the loss reduction against C, with opposite small point changes against gate-only.

Table 41: Exact uniform-cost hybrid comparisons on 125 incidents. Differences are hybrid minus reference with paired 95% intervals. The first row is the single primary test; the other five form the secondary Holm family.

Scope	Reference	Hybrid loss	Difference [95% interval]	Adjusted p
Pooled	Gate	.3179	$+.0059$ [$-.0085$, $+.0210$]	.4438
Pooled	C	.3179	$-.0901$ [$-.1238$, $-.0562$]	.00025
GPT	Gate	.3251	$+.0132$ [$-.0101$, $+.0390$]	.6327
GPT	C	.3251	$-.0935$ [$-.1305$, $-.0563$]	.00025
Grok	Gate	.3106	$-.0014$ [$-.0089$, $+.0067$]	.7371
Grok	C	.3106	$-.0867$ [$-.1215$, $-.0520$]	.00025

At $c = .5$, the hybrid has loss .380 versus the gate's .360; their pointwise difference interval is above zero (Table 42). The hybrid's correct/wrong/escalation rates are 42.5/18.5/38.9%. At $c = .1$ and .25, the two gate policies have identical recorded decisions. At $c = 1$, every case reaches the receiver and the hybrid reproduces C. These intervals describe the specified costs separately.

Table 42: Pooled hybrid loss at fixed costs. Differences are hybrid minus gate-only or receiver C; brackets give pointwise paired 95% intervals.

c	Hybrid loss	Hybrid − gate [95% interval]	Hybrid − C [95% interval]
.10	.1056	.0000 [.0000, .0000]	$-.2832$ [$-.3567$, $-.2099$]
.25	.2280	.0000 [.0000, .0000]	$-.1680$ [$-.2363$, $-.1007$]
.50	.3800	$+.0200$ [$+.0047$, $+.0387$]	$-.0280$ [$-.0713$, $+.0153$]
.75	.4283	$+.0043$ [$-.0243$, $+.0327$]	$+.0083$ [$-.0090$, $+.0303$]
1.00	.4320	$+.0080$ [$-.0267$, $+.0440$]	.0000 [.0000, .0000]

J.4 Statistical conventions

What is compared. A contrast is a difference between two conditions on the same incidents. For receiver outcomes, average usable draws within each incident and receiver, then weight receivers equally. A failed request is not treated as a correct response or an escalation. Each study retains its stated missing-response rule and denominators; the numerical-baseline incident with one missing response keeps its other two responses.

Intervals for two outcomes. Bootstrap intervals resample incidents within software system, preserving the paired conditions and both outcome values. Let $d_i = (d_i^w, d_i^e)$ be incident i's wrong-diagnosis and escalation differences, and $\bar{d}$ their mean. For each of $B = 5{,}000$ bootstrap samples, compute its mean $\bar{d}_b^*$ and the largest absolute deviation from the observed mean:

$$m_b = \max\left(\left|\bar{d}_b^{w*} - \bar{d}^w\right|, \left|\bar{d}_b^{e*} - \bar{d}^e\right|\right). \tag{13}$$

Sort these values increasingly. The shared margin is $r = m_{(\lfloor .95B \rfloor + 1)}$, the 4,751st value with one-based indexing. We report $[\bar{d}^w - r, \bar{d}^w + r]$ and $[\bar{d}^e - r, \bar{d}^e + r]$. This is a nominal 95% simultaneous pair for one contrast, on the original rate scale. The bootstrap seed is 20260809.

Joint tests and multiplicity. The joint randomization test uses the unstandardized statistic

$$T_{\text{obs}} = \max(|\bar{d}^w|, |\bar{d}^e|). \tag{14}$$

In each of $P = 20{,}000$ randomizations, multiply both differences for incident i by one independent sign $\epsilon_i \in \{-1, +1\}$, with equal probability. Recompute T_b from the signed means. The same sign

moves both outcomes together, preserving their pairing. The two-sided Monte Carlo value is

$$p = \frac{1 + \sum_{b=1}^{P} \mathbf{1}\{T_b \geq T_{\text{obs}} - 10^{-15}\}}{P + 1}. \tag{15}$$

The tolerance counts floating-point ties; the sign-swap seed is 20260809. Holm's method adjusts the resulting p values within each specified comparison family (Holm, 1979). We call a result statistically significant when the adjusted test rejects at the 5% level. A joint rejection is evidence of a change in the outcome pair, not necessarily a change in both individual outcomes. Separate receiver significance decisions are not a test that their effects differ. Failure to reject is not evidence of equivalence.

Other analyses and scope. The predictor comparisons use the paired Brier and AUROC procedures specified in Appendix C, not the two-outcome receiver test. The gate and hybrid use the paired integrated-loss tests in Appendices J.2 and J.3. The RCAEval assigned-value experiments use their own specified escalation comparisons and continuation criteria. All resampling uses fixed seeds. We preserve each study's analysis status and never substitute assigned synthetic values for estimated correctness. Assigned values test response to wording and numbers; they do not test calibration or establish the value of a trained estimator in a new domain.

K Full Prompt Templates

K.1 Sender prompts

The prompt text below follows the code used in the experiments. Line wrapping is typeset for readability; bracketed monospace fields stand for values inserted by the code. Example counts illustrate the template and are not additional experimental observations. Sender background is sent as a system message, followed by the applicable reference block. The telemetry summary and diagnosis question form the user message.

The primary sender uses the resource-blind prompt. It receives reference patterns for network delay and packet loss: both leave processor and memory roughly unchanged, while the steadiness of latency and change in throughput distinguish them. No equivalent reference is supplied for resource faults. GPT-4.1 receives the unchanged prompt for the second-sender replication. Sender answer options are shuffled for each sample. This engineered gap is part of the design, not a measurement of naturally occurring ambiguity.

Shared sender background.

> You are a senior site reliability engineer classifying a failure in a containerised microservice system from its telemetry. You are experienced with this class of system and with reading metric time series.
>
> THE ENVIRONMENT
>
> The system is one of three open-source microservice benchmarks: Online Boutique (OB), Sock Shop (SS) or Train Ticket (TT). Each is a web application built from many small services that call one another over the network, deployed as containers. User requests enter through a frontend service and fan out to downstream services.
>
> Exactly one fault was injected into exactly one service. It began at a known moment and was sustained for the rest of the observation window. Everything recorded before that moment is normal operation for this system.
>
> THE INSTRUMENTATION
>
> Every service is sampled once per second. The metric families are:
>
> `[service]_cpu` processor utilisation of that service's container
> `[service]_mem` memory in use by that container, reported here in megabytes
> `[service]_latency` request latency for that service, 90th percentile, reported in milliseconds
> `[service]_workload` volume of requests observed at that service
> `[service]_error` count of errored requests at that service
>
> Not every service exposes every family, but error counts are recorded, including for the frontend. **This matters when you reason about what you do not see: if error counts are

not mentioned among the most abnormal metrics, they were measured and did not move.** Absence from the list means the metric ranked low, never that it was missing.

HOW THE SUMMARY WAS PRODUCED

Metrics are ranked by a robust change score. For each metric, a robust scaler is fitted to its readings before the fault, using the median and interquartile range so that a single spike beforehand cannot distort the scale. The largest score reached after the fault is that metric's "abnormality". Metrics are listed from most to least abnormal. Only the top of the list is shown; everything omitted scored lower than the last entry shown.

HOW TO READ IT

Faults propagate. A service that calls the faulty one will also look abnormal, often dramatically, because it waits on the faulty service. The most abnormal metric in the list is therefore not necessarily on the service that was actually broken.

Both the typical value and the spread carry information. Two faults can produce a similar change in the typical value while differing in how steady or erratic the readings are. The range covering the middle 90 percent of readings, compared against the single worst reading, tells you whether the effect was constant or spiky.

Weigh the evidence you are given rather than the evidence you would expect from a textbook case. This is a real deployment, and a fault may present through fewer signals than you would predict.

Primary sender's network references. The resource-blind sender appends the following block to that background. The same block is used for the GPT-4.1 replication.

CALIBRATION FOR THIS DEPLOYMENT

The following reference ranges were measured from a small labelled subset of failures in this same environment. They cover the network faults only; no comparable reference is available for the resource faults.

Network delay

Processor and memory are unchanged. Latency rises to a high but nearly constant level: the worst reading sits within a few percent of the typical reading during the fault. Request throughput at that service falls substantially, often by a third or more.

Packet loss

Processor and memory are unchanged. Latency rises but erratically: the worst reading is well above the typical reading during the fault, often twice it or more. Throughput is roughly unchanged.

The steadiness of latency, and whether throughput falls, are what separate delay from loss.

Sender question and answer format. The following shows an example option order for the six-fault option set; the five-fault set uses the same question with the socket option omitted. Actual fault-option order is shuffled per call by a recorded seed, not fixed to this example.

`[telemetry summary]`

Exactly one fault was injected. Decide which kind it was.

The possibilities are:

- `cpu`: the service's processor was saturated
- `mem`: the service's memory was exhausted
- `disk`: the service's disk was saturated with reads and writes
- `delay`: network messages to the service were delayed
- `loss`: some network messages to the service were dropped
- `socket`: the service ran out of available network connections

Think it through briefly, then give your decision on a single final line, in exactly this form:
`ANSWER: [one of the options above]`

Fixed-diagnosis confidence question. This separate call uses the same primary-sender background and network references above. The fault descriptions and saved option order are inserted as in the ordinary sender question. Its complete user template is:

`[telemetry summary]`

Exactly one fault was injected. The possible fault types are: `[fault labels and descriptions]`

> The diagnosis already recorded for this incident is: `[diagnosis]`. Using only this telemetry and the deployment information provided, assess your confidence that this fixed diagnosis is correct. Do not replace the diagnosis. Report your estimated probability of correctness, not a percentile or a vote share.
>
> Return exactly one JSON object with only the key `confidence_percent` and a numeric value between 0 and 100 inclusive. Do not include prose or Markdown.

The parser accepts only that key with a finite numeric value in range. Prose, Markdown fences, extra keys, booleans, and missing replies are invalid; no repair is applied. The complete run's confidence values alone supply arm E.

K.2 Receiver prompt

The shared instruction describes a wrong dispatch and escalation as distinct costs and makes neither the default. The hybrid replay reuses C's recorded decisions under this prompt and the probability assessment below; gate-only decides directly from the saved score. The complete common receiver prompt is the framing below followed by the note, options, and required answer line:

> You are on call. An automated triage pass investigated an alert and has handed it to you. You decide what happens next.
>
> You may name the fault type, which sends the incident straight to the team that fixes that kind of fault. Naming the wrong one sends it to the wrong team, and the incident stays open while that is discovered.
>
> Or you may answer ESCALATE, which sends it to a specialist who will investigate further. That resolves it more slowly and occupies someone whose time is scarce.
>
> Neither choice is the safe default. Choose whichever the hand-over supports.
>
> Here is the hand-over note:
>
> `[note]`
>
> Your options are: `[shuffled fault labels]`, ESCALATE
>
> Think it through, then end your reply with exactly one line:
> `DECISION: [one option from the list]`

Only the fault labels are shuffled. The builder then appends `ESCALATE` as the last option. The score and diagnosis-removal comparisons pair the fault order within incident, receiver, and draw. The saved records retain the displayed order for auditing.

K.3 Note structure and original warning formats

A handoff note contains `DECISION`, `EVIDENCE`, and, when present, `ASSESSMENT`. Neutral process sentences supply padding. Diagnosis removal omits the first section while keeping the evidence and tally. Numerical-message experiments retain the diagnosis and change only the assessment and the neutral padding required to match length. Within each synthetic wording, the assessment text is fixed except for the displayed value.

> `DECISION`: Deciding which kind of fault was injected. The reading this note carries forward is `[diagnosis]`.
>
> `EVIDENCE: [ranked measurement text]`
>
> `ASSESSMENT: [carrier text]`

Warning-detail formats. The A–D labels below belong to the original warning-detail experiment. Labels are local to each study. This illustration uses the five-candidate tally `cpu 9, mem 2, disk 1, delay 6, loss 2`; the agreement score is the agreement value computed from that tally, not its leading share.

> A, no statement: omit the `ASSESSMENT` section.
>
> B, qualitative warning: The analysis was divided on this.
>
> C, agreement score: Agreement across independent runs: 0.17 of maximum.
>
> D, tally: Independent runs: cpu 9, mem 2, disk 1, delay 6, loss 2.

The agreement score is one minus normalized vote entropy, rounded to two decimal places. The full tally retains zero-count candidates as well as candidates that received votes. In the diagnosis-removal study, the two notes carry the same full tally assessment; one omits the separate `DECISION` section. Neither the measurements nor the tally is changed by removal.

Estimated-correctness-probability message. Arm A has no assessment. Arm B lists every candidate and its exact count, including zeros. With the same illustrative counts its complete text is:

> Across 20 independent analyses, cpu was selected 9 times, mem was selected 2 times, disk was selected 1 times, delay was selected 6 times, loss was selected 2 times. These repeated-answer counts describe how often each answer occurred; they are not probabilities that the stated diagnosis is correct.

Arm C uses the following text, with `[percentage]` replaced by the rounded cross-fitted prediction:

> Using three complete analyses and historical outcomes, the estimated probability that the stated diagnosis is correct is `[percentage]`%. This calibrated probability describes expected correctness for similar cases; it is not a percentile or a vote share.

All arms state the fixed-three plurality diagnosis. On eight incidents the twenty-response plurality differs; the tally is retained exactly, rather than adjusted to favor the diagnosis shown. The score groups are frozen at 42 low, 42 middle, and 41 high cases. Their unrounded score ranges are approximately .195–.471, .474–.675, and .675–.941. Rounded boundary values can therefore appear in adjacent groups. Group membership uses the frozen ranks, not newly rounded cutoffs.

Leading-share and self-reported-confidence messages. Arms D and E use the same receiver framing, evidence, and fixed-three diagnosis as A, B, and C above. D removes the full tally and presents only its leading label and percentage:

> Across twenty independent analyses, the leading diagnosis `[leader]` received `[percentage]`% of the answers. This percentage describes how often the analyses agreed with that diagnosis; it is not a probability that the stated diagnosis is correct.

For ties, "the leading diagnosis" becomes "a joint leading diagnosis". The leader is from all twenty responses, even on the eight incidents where it differs from the fixed-three diagnosis. E uses:

> Using the incident evidence, the sender estimates the probability that the stated diagnosis is correct at `[percentage]`%. This self-reported probability expresses the sender's own confidence; it is not a percentile or a vote share.

The percentage is the complete run's raw probability times 100, rounded to the nearest integer. No claim of calibration or historical-outcome fitting is attached to E.

Second-sender continuation rule. Before the GPT-4.1 run, we required at least forty complete incidents with more than one sampled diagnosis to proceed to receiver calls. Each sender request receives one attempt, and incidents with a missing sender answer do not enter the matched reader comparison. The primary reader test is one pooled joint contrast. Receiver splits, divided and unanimous strata, and response modes are secondary.

K.4 Interpretation and stronger receiver instructions

These conditions prepend information to the common receiver instructions. R0 is the unbriefed prompt. R1 adds the following explanation:

> The hand-over note may contain an ASSESSMENT section. It reports how twenty independent analyses of the same evidence voted. If those votes were split, the evidence was genuinely ambiguous and the stated conclusion is less reliable than its wording suggests. If all twenty agreed, they agreed with each other, which does not by itself make them right.

R2 consists of R1, the full sender background above, and this description of confusable fault pairs:

> In this deployment two pairs of faults present alike. Processor saturation and disk saturation both raise processor time, and differ only in the level it settles at. Network delay and packet loss both raise latency, and differ only in how steady it is.

R3 consists of R2 followed by the resource-only reference block below. These are the receiver's added resource references, distinct from the network-only block given to the primary sender.

> CALIBRATION FOR THIS DEPLOYMENT
>
> The following reference ranges were measured from a small labelled subset of failures in this same environment. They cover the resource faults only; no comparable reference is available for the network faults.
>
> Processor saturation
>
> The affected service's processor sits at or very near 100 percent for the whole window.
>
> Disk saturation
>
> No disk metric is collected. Disk saturation surfaces as I/O wait charged to processor time, which produces a moderate and very steady rise to roughly 30 percent. It does not approach 100 percent. A service sitting near 30 percent rather than near 100 is the signature to look for.
>
> Memory exhaustion
>
> The affected service's memory rises by hundreds of megabytes, far beyond ordinary variation. Its processor may also run high, so memory growth rather than processor level is the tell.
>
> The level the processor settles at is what separates disk saturation from processor saturation.

Stronger receiver prompt. This separate comparison orders its inputs as the resource references, the common receiver framing, and the following role description, before showing the matched note. The richer evidence is present in both arms.

> The note was written by an automated triage pass, not by a person. Its stated conclusion is its best reading of the measurements, and it can be wrong.
>
> You are not required to agree with it. You may name a different fault if the measurements support one, and you should say so plainly when you do.
>
> If you answer ESCALATE, the incident goes to a specialist who holds reference measurements for this deployment that are not available to you, and who can therefore settle cases the triage pass could not.

The specialist description is part of the prompt's framing; the study records the request for review rather than a specialist's subsequent action.

K.5 Decoder and staged-reveal prompts

Independent decoder. The decoder is asked to reconstruct the sender tally, not to dispatch or escalate. Its instruction is fixed across note conditions:

> A triage pass analysed one incident twenty independent times and produced the note below. The individual analyses are not available to you; only the note is.
>
> Estimate how those twenty analyses divided across the candidates. Your counts must sum to 20.
>
> Candidates: `[comma-separated candidate labels]`
>
> `[note]`
>
> Think it through, then end your reply with exactly one line:
> `TALLY: [candidate]=[count], [candidate]=[count], ...`

Candidate order is paired across the notes for an incident, decoder, and draw. The parser requires every allowed candidate exactly once, nonnegative integer counts, and a total of twenty.

Notes supplied to the decoder. The real-note decoder uses 63 incidents and the saved packet builder. Its notes have five fields: `DECISION`, `AMBIGUITY`, `EVIDENCE`, `RULED_OUT`, and `NEXT`, in that order. A diagnosis is "in play" when at least two of the twenty sender responses select it. In-play diagnoses are listed alphabetically. The decoder's separate options list still includes every allowed fault.

The honest-note template is:

> DECISION: Deciding which kind of fault was injected. After repeated analysis the choice has narrowed to: [in-play diagnoses].
> AMBIGUITY: Repeated analysis of the same evidence did not converge. Different passes read the same numbers differently: [per-diagnosis signal descriptions]. No pass could exclude the alternatives, which is why this was not decided.
> EVIDENCE: The measurements this rests on:
> [six ranked measurement lines]
> RULED_OUT: [eliminated diagnoses and flat measurement families]
> NEXT: [check that would distinguish the in-play diagnoses]

Measurement lines give the before median, during median, middle 90% range, and worst value. Signal descriptions come from the sender replies: for each diagnosis, count the fraction mentioning each measurement family or shape property. Retain at most two signals whose fraction exceeds the largest fraction among other in-play diagnoses by more than .2, ordered by that difference and then by signal name, both descending. For each candidate, insert "the reading of [diagnosis] leans on [signals]". If none qualifies, insert "the reading of [diagnosis] leans on nothing the other readings did not also use". Join these descriptions with semicolons.

RULED_OUT lists allowed faults outside the in-play set as "Eliminated: [faults]. No pass found support for these." If that set is empty, it says "Nothing was eliminated outright." When present, append "The metric families that stayed within their usual range throughout: [families]." These are families absent from the top-ranked metric list used in the summary.

For NEXT, compare each alphabetical pair of in-play diagnoses. If they have distinct cited signals, describe each side as "[diagnosis] turns on [signals]". When only one side has distinct signals, use "[diagnosis] turns on [signals], which [other diagnosis] does not rely on". Join pair descriptions with semicolons in:

> The readings differ on what to make of the same numbers: [pair descriptions]. A measurement that settles which of those descriptions fits, such as comparing against a known-good period on the same service, would separate [in-play diagnoses joined by 'from'].

If no pair has distinct signals, use:

> A measurement able to distinguish [in-play diagnoses joined by 'from'] would settle this; the metrics reported above did not separate them.

The concealed note keeps EVIDENCE and RULED_OUT, removes AMBIGUITY, and replaces DECISION and NEXT with:

> DECISION: This is a [asserted diagnosis] fault. The measurements below are what that reading rests on, and it is the conclusion being handed over.
> NEXT: Confirming [asserted diagnosis] against a known-good period on the same service would close this out before any change is made.

The frozen schedule alternates the leading and runner-up diagnoses in sorted incident order, starting with the leader. Vote ties follow the first-occurrence order in the saved sender responses. Neutral process sentences are appended in a fixed order under NOTES to match the honest note's token count within the 5% tolerance. The saved notes retain the exact padding. The transformations use sender responses and measurements; true faults are used only for scoring.

Staged diagnosis reveal. The first turn uses the no-conclusion note and the common receiver instruction. Its saved response remains in the conversation as the assistant turn. The second user turn is:

> The automated triage pass has now revealed the conclusion it withheld from the note: [diagnosis].
> Make your final decision. You may keep your earlier choice, choose a different fault, or answer ESCALATE. Your options, in the same order, are: [fault labels], ESCALATE.

> End your reply with exactly one line:
> `DECISION: [one option from the list]`

The second turn adds the diagnosis, not new telemetry. It retains the first turn's fault-option order.

K.6 SYNTHETIC-VALUE PROMPTS

The RCAEval numerical experiments use the common receiver prompt, the incident's fixed evidence and diagnosis, and one of the assessments below. Appendix E gives the value grids, results, and continuation history. Within each wording, only the displayed value changes; the assessment is an experimental statement rather than a newly measured estimate of the incident.

Initial endpoint experiment. The first-level statements are:

> Synthetic probability of correctness: `[value]`/100.
> Synthetic relative reliability percentile: `[value]`/100.

Level two appends:

> This number was assigned for this instrument check and was not computed from the incident evidence.

Level three also appends:

> Lower values support verifying the diagnosis or escalating; higher values do not guarantee that it is correct.

Concise and explained semantic statements. Every version begins with this common prefix:

> For this scenario, an external assessment reports:

The concise correctness and uncertainty bodies are:

> Estimated probability that the stated diagnosis is correct: `[probability]`%.
> Measured uncertainty of the stated diagnosis: `[uncertainty]`/100. Higher values mean more uncertainty.

The explained bodies are:

> An external method combined three independent analyses and historical outcomes. It estimates a `[probability]`% probability that the stated diagnosis is correct. Lower values mean more uncertainty; the estimate does not guarantee correctness.
> An external method combined token entropy, chosen-token probability, and the top-two probability gap from three independent analyses. It measures uncertainty in the stated diagnosis as `[uncertainty]`/100. Higher values mean more uncertainty; the score does not guarantee error.

The five-level disjoint experiment reuses these explained bodies and the common prefix without changing their wording.

Explicit agent-sure and uncertainty statements. The certainty version is:

> The diagnosing agent may not be sure of the diagnosis it gave. Based on our estimate from three independent analyses, the agent is `[certainty]`% sure that its stated diagnosis is correct. This estimate combines the agent's token probabilities, the gap between its top choices, and token entropy. Lower percentages mean the agent is less sure and more uncertain; higher percentages mean it is more sure and less uncertain. Use this estimate together with the evidence when deciding whether to rely on the diagnosis or escalate. It is an estimate, not a guarantee.

The matched uncertainty version is:

> The diagnosing agent may not be sure of the diagnosis it gave. Based on our estimate from three independent analyses, the agent's uncertainty about its stated diagnosis is `[uncertainty]`/100. This estimate combines the agent's token probabilities, the gap between its top choices, and token entropy. Higher values mean the agent is less sure and more uncertain; lower values mean it is more sure and less uncertain. Use this estimate together with the evidence when deciding whether to rely on the diagnosis or escalate. It is an estimate, not a guarantee.

K.7 TOOL-BASED SYNTHETIC-VALUE PROMPTS

The three native conditions share the following user-message template. The diagnosis is the five-investigation plurality, with ties resolved by the first supporting trajectory. The same trajectory supplies the tool evidence. Fault-object and root-cause lists are shuffled with the same recorded seed, paired within incident, reader, and draw.

> You are the downstream on-call engineer. An automated investigator handed you the note below. You may commit to a diagnosis, which routes the incident immediately, or answer ESCALATE for slower specialist investigation. A wrong diagnosis routes the incident to the wrong team. Neither choice is the safe default.
>
> `DECISION`: The diagnosis this handoff carries forward is `[fault object] | [root cause]`.
>
> `EVIDENCE`: The upstream tool investigation recorded:
> `[unchanged tool transcript]`
>
> `ASSESSMENT`: Using five complete investigations and historical outcomes, an externally trained model estimates a `[percentage]`% probability that the stated diagnosis is correct. This is an out-of-domain estimate, not a percentile or a vote share; calibration in this environment has not been established.
>
> Allowed fault objects:
> `[fault-object list]`
>
> Allowed root causes:
> `[root-cause list]`
>
> Use only the note. End with exactly one of these forms:
> `DECISION: ESCALATE`
> `DECISION: [fault_object] | [root_cause]`

The inserted percentage is 80, 20, or 5, assigned independently of the incident's correctness. The claim that a model estimated it is part of the controlled experimental message, not a description of how these values were obtained. The complete notes differ only in that numeral; all tool observations and any neutral padding are unchanged. The displayed values must not be treated as actual calibrated predictions.