HOW STABLE ARE LLM ERRORS? FAILURE-CONDITIONED DECISION STABILITY IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models produce fluent but incorrect answers, but wrong answers need not be equally stable under small changes in query context. We study this *failure-conditioned decision stability* on candidate-mappable closed-answer failures using a fixed weak re-query that provides no gold label, feedback, or additional evidence. For each failure, we compare the gold candidate with the realized wrong candidate and use their final-window candidate margin as the main pre-query coordinate; full-response traces are treated descriptively, and post-query quantities only as response diagnostics. Across four tasks, recovery in the two primary models rises from near zero in the most failure-favored final-window states to 39.4-60.2% in the largest-margin bins, with the same Q5-Q1 ordering in every task; the remaining models expose weaker, sparse, re-query-sensitive, or negative boundary cases. A replay-matched audit of 607 Qwen failures removes post-answer continuation from the readout: the literal answer-token margin yields 0.794 AUROC versus 0.800 for the original final-window coordinate. On the same 607 replay-matched Qwen failures, confidence-controlled analysis yields 0.789 AUROC for the margin versus 0.661 for wrong-candidate probability. A separate response decomposition further shows that the margin is associated both with whether an erroneous answer changes and, among changed non-binary answers, with whether the new answer is gold rather than another wrong candidate. We position residual-error stability as an offline analysis axis rather than a deployment-time error detector.

1 INTRODUCTION

Large language models still produce fluent but incorrect answers across question answering and reasoning tasks (Ji et al., 2023). Existing work estimates model confidence, detects hallucinations, studies intrinsic self-correction, and measures answer revision under repeated questioning or explicit challenge (Kadavath et al., 2022; Manakul et al., 2023; Farquhar et al., 2024; Huang et al., 2024; Laban et al., 2023; Xie et al., 2024); representation analyses also expose predictions and truth-related structure from intermediate states (Belrose et al., 2023; Burns et al., 2022). We ask a different, failure-conditioned question: once a natural wrong answer has already occurred, how stable is that decision under the same weak, information-free re-query, and where does the answer move when it changes? We study this as *failure-conditioned decision stability*, with gold recovery as one outcome rather than as an intrinsic per-error probability.

We study candidate-mappable closed-answer failures, for which both the gold candidate a_i^+ and the realized wrong candidate a_i^- are identified. For each retained failure, we trace gold-versus-realized-failure candidate competition during the original response and use a final-window candidate margin d_i as the main pre-query coordinate. Full-response onset, persistence, prevalence, and path-type summaries are retained only as secondary descriptive diagnostics because the evaluated responses are predominantly short.

We then apply the same weak cautionary re-query $P_{\min}$ to every retained failure, providing no gold label, feedback, or additional evidence. The original response is not replayed in the new context; $P_{\min}$ therefore defines a standardized low-information query perturbation rather than an explicit multi-turn self-correction protocol. We keep the evidence roles separate: original traces are descrip-

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.

tive, d_i is measured before the re-query and used for response stratification, and the post-query ratio $\tilde{r}_i^{\text{fail}}$ is measured only on the re-query response and used as a displacement diagnostic.

We evaluate six instruction-tuned LLMs on TruthfulQA-MC (Lin et al., 2022), SciFact (Wadden et al., 2020), BoolQ (Clark et al., 2019), and ARC-Challenge (Clark et al., 2018). Qwen2.5-1.5B and Gemma-2-2B provide the primary evidence, with the remaining models used to test cross-family support and failure boundaries. In the primary models, recovery rises from near zero in the strongest failure-favored decision states to 39.4% and 60.2% in the weakest-margin bins, and the Q5-Q1 ordering holds within every task. In a replay-matched Qwen audit, the result remains when the readout is restricted to the literal answer token, and a confidence-controlled analysis shows that wrong-candidate confidence alone does not explain the stratification. These results support the final-window gold-versus-failure candidate margin as a failure-conditioned coordinate of response stability under the weak re-query, while the full-response trace summaries remain descriptive.

The framework is intended for controlled offline evaluation rather than deployment-time error detection. Its practical role is to characterize the residual errors left by models, checkpoints, or interventions beyond aggregate accuracy alone. Two systems with similar accuracy may leave failure populations that differ substantially in their response to the same weak re-query; residual-error stability therefore provides an additional axis for comparing what kinds of errors remain.

Our contributions are:

- **A failure-conditioned decision-stability framework.** We treat already-observed natural errors as the analysis population and characterize how those residual errors respond to a fixed weak re-query, separating answer stability, answer change, and gold recovery.
- **Preliminary evidence beyond aggregate accuracy.** A clear SciFact contrast shows that ordinary accuracy and residual-error response under the same re-query need not coincide, providing preliminary evidence for decision stability as a complementary offline evaluation axis.
- **Final-window candidate competition as an organizing coordinate with explicit controls.** The final-window gold-versus-realized-failure margin stratifies both answer change and gold recovery in the primary models. Literal-answer-token and confidence controls test alternative explanations for this relationship; prompt-variant analyses quantify the intervention dependence of recovery, and cross-model analyses expose its empirical boundaries.

2 Decision-Stability Analysis Framework

We represent each retained natural failure as a gold-versus-realized-failure competition record. Figure 1 separates three evidence roles: original traces describe pre-query competition, d_i measures final-window candidate competition before the re-query, and post-query traces are used only to diagnose response displacement.

2.1 Natural Failure Construction

Natural failure construction fixes the population analyzed by the framework. For each task instance x_i with candidate set $\mathcal{A}_i$, gold candidate a_i^+, task prompt q_i, and evaluated model M, a fixed task-specific parser ρ maps the ordinary output to $\hat{a}_i = \rho(M(q_i))$ when possible. We retain only valid incorrect outputs and denote the realized wrong candidate by a_i^-:

$$\mathcal{F} = \{(x_i, a_i^+, a_i^-) : \hat{a}_i = a_i^-,\ a_i^- \in \mathcal{A}_i \setminus \{a_i^+\}\}.$$

Correct and unmappable outputs are excluded. Thus, $\mathcal{F}$ contains naturally occurring, candidate-mappable failures; it is neither an estimate of overall model error rate nor a set of forced or adversarial errors.

2.2 Competition-State Tracing

For each $(x_i, a_i^+, a_i^-) \in \mathcal{F}$, we trace only the gold candidate and the realized wrong candidate. This pairwise readout targets the observed failure and does not model probability mass over all possible answer strings.

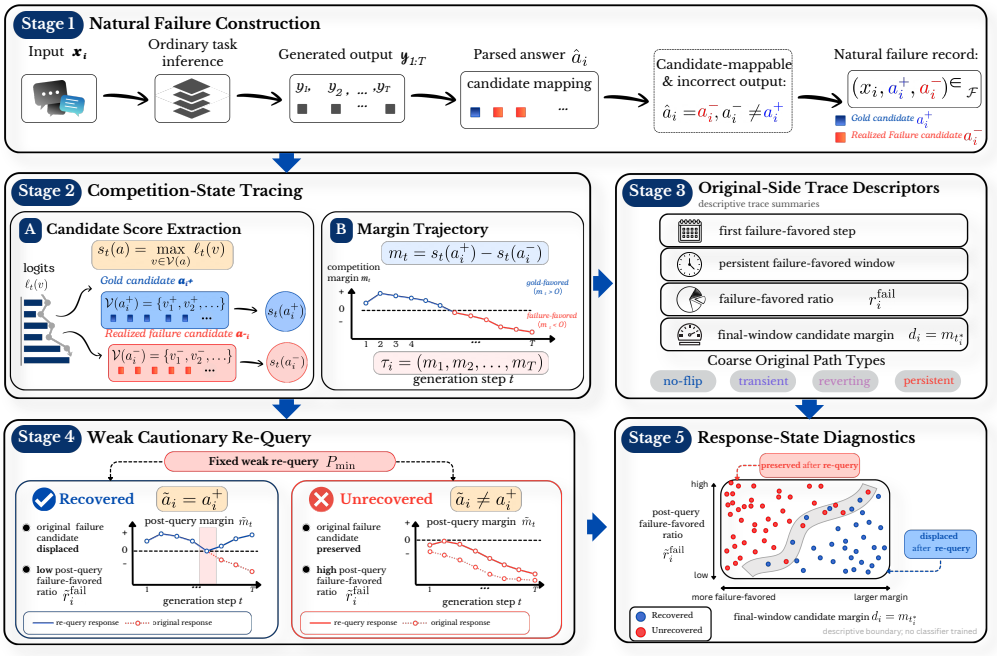


Figure 1: Failure-analysis framework. Natural failures are mapped to gold-versus-realized-failure competition records, summarized before intervention, re-queried under the fixed weak cautionary condition $P_{\min}$, and analyzed after the re-query. $P_{\min}$ is an intervention condition rather than a correction algorithm.

At generation step t, let $\ell_t(v)$ denote the next-token logit for token v. For a candidate label a, let $\mathcal{V}(a)$ be the tokenizer-dependent set of first-token ids induced by accepted textual variants of that label. We score the candidate by

$$s_t(a) = \max_{v \in \mathcal{V}(a)} \ell_t(v).$$

The max over accepted verbalizers avoids giving candidates with more variants a direct additive advantage, but is not a full-label likelihood; robustness checks over alternative readouts are in Appendix A.12.

We also compute a candidate-probability readout using the probability version of the same verbalizer rule:

$$\pi_t(a) = \max_{v \in \mathcal{V}(a)} \text{softmax}(\ell_t)_v.$$

This readout is used to select the final-window coordinate in Section 2.3 and as the confidence covariate in the confidence-control analysis; the margin m_t remains the primary quantity for original-trace analysis.

The gold-versus-realized-failure margin is

$$m_t = s_t(a_i^+) - s_t(a_i^-).$$

Positive values indicate gold-favored competition, and negative values indicate failure-favored competition. Let T denote the total number of generation steps in the original response for record i. The original competition trajectory is

$$\tau_i = (m_1, m_2, \ldots, m_T).$$

2.3 Original-Side Trace Descriptors

From each original competition trace $\tau_i = (m_1, \ldots, m_T)$, we retain four summaries: the first failure-favored onset f_i, the first persistent failure-favored window p_i, the failure-favored ratio r_i^{fail}, and the final-window candidate margin d_i. These quantities do not play equal evidentiary roles.

Because the evaluated responses are predominantly short, f_i, p_i, r_i^{fail}, and the resulting path taxonomy are used only as descriptive summaries of the observed trace; we do not interpret them as evidence of a rich pre-answer formation process. The main pre-query coordinate used in the response-stratification analysis is d_i.

First failure-favored onset. We record the first step at which the realized failure candidate is locally preferred:

$$f_i = \min\{t : m_t < 0\},$$

when this set is nonempty; otherwise f_i is undefined.

First persistent failure-favored window. Using the fixed threshold $K = 2$ for all models and tasks, we record the first two-step run in which the failure candidate remains locally preferred:

$$p_i = \min\left\{t \in \{1, \ldots, T-1\} : m_t < 0 \text{ and } m_{t+1} < 0\right\},$$

when this set is nonempty; otherwise p_i is undefined. For short responses, this quantity may include post-answer continuation and is therefore treated descriptively.

Trajectory failure-favored ratio. We summarize the prevalence of failure-favored competition across the observed response by

$$r_i^{\text{fail}} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}[m_t < 0].$$

Like f_i and p_i, this is a descriptive trace summary rather than a primary recovery predictor.

Coarse original path taxonomy. Using f_i, p_i, and later reversion, we assign four descriptive path types: *no-flip* (no $m_t < 0$), *transient* (f_i defined but p_i undefined), *reverting* (p_i defined and followed by some later $m_t > 0$), and *persistent* (p_i defined with no later return to $m_t > 0$). No-flip cases are retained as readout-level edge cases; detailed implementation rules are given in Appendix A.2.6.

Final-window candidate margin. Because short-answer generations may end with punctuation or formatting tokens, the main pre-query coordinate is selected from a fixed backward window

$$\mathcal{W}_i = \{t : \max(1, T - R) \le t \le T\}, \qquad R = 3.$$

Within this window, t_i^* maximizes $\max\{\pi_t(a_i^+), \pi_t(a_i^-)\}$, with larger $|m_t|$ as the tie-breaker, and

$$d_i = m_{t_i^*}.$$

More negative d_i indicates stronger local preference for the realized failure candidate. Because t_i^* is selected from a final window rather than guaranteed to coincide with the literal emitted answer token, positive values can reflect answer-position or parser–verbalizer mismatch. We therefore treat d_i as a final-window candidate-competition coordinate; the literal-answer-token audit provides the cleaner answer-token check.

2.4 Weak Cautionary Re-Query

For each retained failure $(x_i, a_i^+, a_i^-) \in \mathcal{F}$, we apply the same weak cautionary re-query $P_{\min}$ to the original task prompt q_i. The intervention appends a fixed cautionary instruction without providing the gold label, the original generated answer, the realized failure candidate, task-specific feedback, external evidence, or an explanation request. Because the original response itself is not replayed in the new context, $P_{\min}$ should be interpreted as a standardized low-information query perturbation rather than as an explicit multi-turn self-correction protocol. The fixed template is

```
[original task prompt]
Your previous answer may be incorrect.
Reconsider your answer carefully.  Do not provide any
explanation.  Output only the final answer in exactly the
same format as before.
Answer:
```

Table 1: Model suite and retained failure records. Input cap is the maximum attempted instances per task. Attempted→retained reports total attempted inputs and retained candidate-mappable natural failures after incorrect-answer, candidate-mapping, and trace-availability filters.

Model	Family / scale	Input cap	Attempted→retained	Role
Qwen2.5-1.5B	Qwen / 1.5B	500/task	1,639→682	Primary analysis
Gemma-2-2B	Gemma / 2B	500/task	1,639→613	Primary cross-family analysis
Granite-3.3-8B	Granite / 8B	500/task	1,639→439	Cross-family support
Ministral-8B	Mistral / 8B	200/task	800→227	Sparse-recovery check
Llama-3.2-1B	Llama / 1B	200/task	800→433	Re-query-sensitive boundary
SmolLM2-1.7B	SmolLM / 1.7B	200/task	800→471	Margin-signal boundary

The phrase "previous answer" therefore functions only as a generic cautionary cue; neither the original answer string nor a_i^- is present in the model context. Re-query outputs are mapped by a stricter final-label parser $\tilde{\rho}$ over the same task label space (decoding details are in Appendix A.2):

$$\tilde{a}_i = \tilde{\rho}(M(P_{\min}(q_i))).$$

We count recovery only when $\tilde{a}_i = a_i^+$; any non-gold or unmappable output is unrecovered. Recovery therefore measures gold-directed response under this fixed re-query condition, not explicit multi-turn self-correction or the model's best achievable correction rate.

We also trace the re-query response, applying the same candidate-scoring rule $s_t(a)$ to its logits to yield post-query scores $\tilde{s}_t(a)$ and margins $\tilde{m}_t = \tilde{s}_t(a_i^+) - \tilde{s}_t(a_i^-)$ for post-query displacement diagnosis (Section 2.5).

2.5 Post-Query Persistence

For post-query diagnosis, we compute the post-query failure-favored ratio

$$\tilde{r}_i^{\text{fail}} = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \mathbb{I}[\tilde{m}_t < 0],$$

where $\tilde{T}$ is the re-query response length. Low $\tilde{r}_i^{\text{fail}}$ indicates displacement of the original failure candidate; high values indicate continued failure-favored competition. Because this quantity is measured on the re-query response itself, it is a post-query diagnostic rather than a pre-query predictor (Appendix A.2).

3 Experimental Setup

We evaluate the framework on six instruction-tuned LLMs and four closed-answer tasks under the same fixed weak re-query, separating primary evidence, cross-family support, and boundary cases.

3.1 Models and Tasks

Table 1 summarizes the model suite, input caps, retained failure records, and model roles. The evaluated checkpoints are Qwen2.5-1.5B Team (2024b), Gemma-2-2B Team (2024a), Granite-3.3-8B IBM (2025), Ministral-8B AI (2024b), Llama-3.2-1B AI (2024a), and SmolLM2-1.7B Allal et al. (2025). Qwen2.5-1.5B and Gemma-2-2B provide the primary evidence. Granite provides weaker margin-signal cross-family support, Llama provides a re-query-sensitivity boundary, Ministral tests a sparse-recovery regime, and SmolLM2 provides a negative boundary for the pre-query margin signal.

We use TruthfulQA-MC (Lin et al., 2022), SciFact entailment (Wadden et al., 2020), BoolQ (Clark et al., 2019), and ARC-Challenge (Clark et al., 2018); all four provide finite candidate spaces with identifiable gold labels, so each retained error maps to a gold-versus-realized-failure competition record.

3.2 Retention and Measurements

For each model-task pair, ordinary inference supplies the candidate-mappable failures defined in Section 2.1; no adversarial distractors, context perturbations, or forced errors are introduced. Qwen, Gemma, and Granite use up to 500 inputs per task, and Llama, SmolLM, and Ministral use 200. Table 1 reports attempted and retained counts; Appendix A.3 gives the full filtering breakdown. Correct cases are used only for re-query-sensitivity controls.

3.3 Evaluation Protocol

The main analyses focus on four questions: whether the final-window candidate margin stratifies response under the weak re-query; whether this relation survives short-answer and confidence controls; whether d_i predicts merely answer change or also the destination of a changed answer; and how aggregate recovery and instance-level response agreement vary across re-query variants and model families. Full-response path types and post-query persistence are retained only as secondary descriptive diagnostics in the Appendix.

For the d_i-binned curves, we divide retained failures into five quantile bins from the most failure-favored values (Q1) to the largest-margin values (Q5), reporting Wilson 95% binomial intervals. To assess stability, we run 1,000 bootstrap resamples over retained failure records, recomputing recovered-versus-unrecovered gaps and univariate AUROC summaries for the main signals (Appendix A.6).

Auxiliary path-type, post-query, feature-source, and internal-state analyses are reported in the Appendix and are not used to define the main decision-stability claims.

4 Main Results

The results separate three evidence roles: descriptive pre-query traces, final-window gold-versus-failure candidate competition, and post-query displacement. We first establish the re-query operating regime, then test response stratification and its main alternative explanations, before examining cross-model boundaries and residual-error profiles.

4.1 The Weak Re-Query Defines a Controlled but Intervention-Conditional Regime

The fixed weak re-query is intentionally non-optimized: it provides a common perturbation condition for comparing already-wrong decisions rather than maximizing correction. Across the six models, it recovers a minority of retained failures, with recovery rates ranging from 4.0% to 21.0% over 2,865 retained model-instance failure records. Qwen2.5-1.5B and Gemma-2-2B recover 16.6% and 18.3% of retained failures, respectively, leaving substantial stable and changed failure subsets for analysis.

Correct-case controls show that $P_{\min}$ preserves originally correct answers in 91.3%, 89.2%, and 94.4% of cases for Qwen, Gemma, and Granite, but only 62.8% for Llama. Aggregate recovery rates vary by only 2.5-3.8 percentage points across the four prompt variants for Qwen, Gemma, and Granite. Importantly, this aggregate similarity does not imply identical per-example behavior. On the matched failure records, recovery under $P_{\min}$ has substantial but imperfect agreement with the other three variants: Cohen's κ ranges from 0.64-0.72 for Qwen, 0.78–0.79 for Gemma, and 0.60–0.71 for Granite, with corresponding recovered-set Jaccard overlaps of 0.54-0.62, 0.69-0.71, and 0.48–0.60. Llama is less stable ($\kappa = 0.40$–0.71) and also has a 37.2% correct-case false-revision rate. We therefore treat response stability as intervention-conditional rather than as an intrinsic per-error property, and retain Llama only as a re-query-sensitive boundary case (Appendix A.7.1).

4.2 Final-Window Candidate Competition Stratifies Failure Stability and Recovery

We next test whether the response to the weak re-query is organized by the final-window candidate margin d_i (Section 2.3). The margin is confidence-related, and we do not claim it as a new

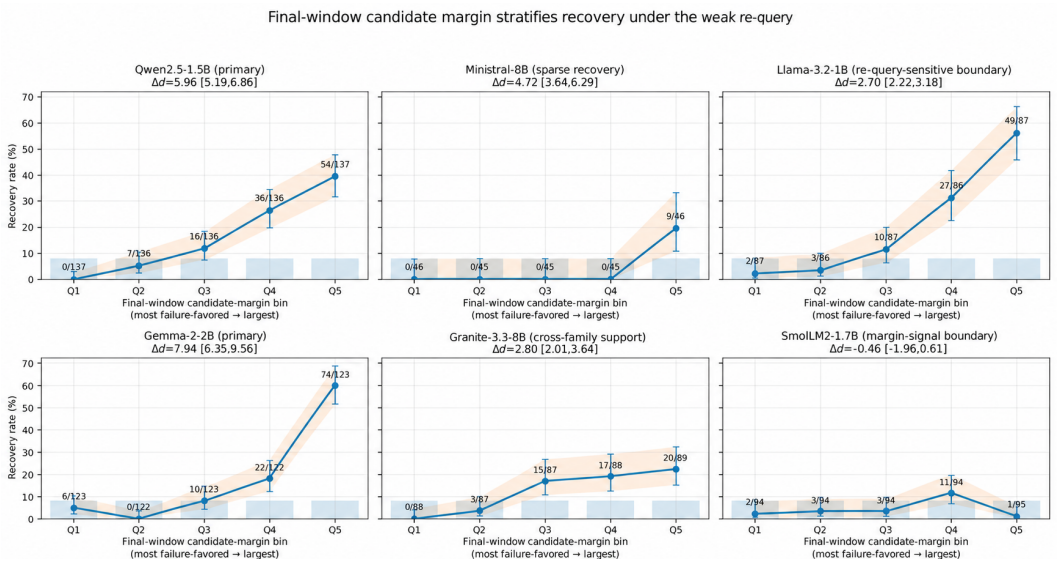


Figure 2: The final-window candidate margin stratifies recovery under the weak re-query. Retained failures are binned by d_i, from the most failure-favored values (Q1) to the largest values (Q5). Because d_i is selected from a short final window rather than necessarily from the literal emitted answer token, positive values can reflect parser–verbalizer or answer-position mismatch and are not interpreted as evidence that the emitted wrong answer was itself gold-favored. Curves show recovery rates under $P_{\min}$, shaded bands show Wilson 95% binomial intervals, and bars show relative bin counts. Panel-level Δd denotes the recovered-minus-unrecovered mean gap in d_i, with 95% bootstrap confidence intervals in brackets.

confidence statistic; prior work has already shown that model probabilities can carry substantial information about answer correctness (Kadavath et al., 2022). Every record in our analysis is already a natural model error. We therefore ask two narrower questions: whether d_i stratifies the stability of an already-wrong decision, and, when the answer changes, whether it also distinguishes a change to the gold candidate from a change to another wrong candidate.

Figure 2 shows strong recovery stratification in both primary models. Recovery rises from 0.0% to 39.4% for Qwen2.5-1.5B and from 4.9% to 60.2% for Gemma-2-2B between the most failure-favored bins (Q1) and the largest-margin bins (Q5), compared with overall recovery rates of 16.6% and 18.3%.

The ordering is not only a pooled effect: within every task, Q5 recovery exceeds Q1 in both primary models, with Q5-Q1 lifts ranging 28.1-57.1 percentage points for Qwen2.5-1.5B and 40.7-54.8 points for Gemma-2-2B (Appendix A.4). Thus d_i preserves the same qualitative recovery ordering across tasks with different answer spaces, rather than isolating an easy pooled subset.

Gold-relative readout robustness. The recovery ordering is not specific to the max-logit scale of d_i. Distinct probability-margin and gold-versus-failure rank readouts preserve the same qualitative ordering (Appendix A.12). The pairwise normalized probability readout is a monotone transformation of d_i and is therefore treated only as a numerical sanity check, not as independent robustness evidence. These alternatives remain gold-relative readouts and do not distinguish gold-relative competition from a fully gold-independent top-two boundary measure.

4.3 Answer Change and Gold Recovery Are Distinct Outcomes

Recovery conflates two events: whether the original wrong answer changes at all, and whether a changed answer lands on the gold candidate rather than another wrong candidate. We therefore decompose each re-query response into three outcomes: retaining the original wrong candidate, flipping to another wrong candidate, or flipping to the gold candidate. All 682 Qwen and 613

Table 2: Direct tests of the two main alternative explanations on the 607 replay-matched Qwen failures. AUROC values use 5-fold out-of-fold evaluation.

Audit / predictor	Result
Short-answer audit	
Replay-matched failures	607/682 (89.0%)
Answer begins at first generated token	607/607 (100%)
Original final-window margin	0.800 AUROC
Literal answer-token margin	0.794 AUROC
Confidence control	
Task identity only	0.648 AUROC
Task + failure-candidate probability	0.661 AUROC
Task + answer-token margin	0.789 AUROC
Task + probability + margin	0.807 AUROC
Margin over failure probability	+0.128 [0.088, 0.171]
Confidence-matched analysis	16/20 bins; +21.3 pp

Table 3: Decomposition of failure response under $P_{\min}$. "Any flip" predicts whether the original wrong candidate changes. "Gold destination" distinguishes gold from another wrong candidate among changed answers; BoolQ is excluded from this second analysis because, in a binary answer space, every valid flip necessarily lands on gold. AUROC values use 5-fold out-of-fold logistic models with task identity alone versus task identity plus d_i.

Model	Stay / other / gold	Any flip: task $\rightarrow$ task+d_i	Gold destination: task $\rightarrow$ task+d_i
Qwen2.5-1.5B	497 / 72 / 113	0.575 $\rightarrow$ 0.705	0.666 $\rightarrow$ 0.858
Gemma-2-2B	441 / 60 / 112	0.636 $\rightarrow$ 0.808	0.520 $\rightarrow$ 0.745

Gemma re-query responses in this analysis map to a valid task candidate, so these three categories exhaust the retained failure sets.

The margin is therefore not only associated with whether an erroneous answer changes. For Qwen, adding d_i improves any-flip AUROC by 0.130 (95% bootstrap CI $[0.088, 0.176]$) and, among non-BoolQ flips, improves gold-versus-other-wrong destination AUROC by 0.192 ($[0.115, 0.263]$). Gemma shows the same pattern: $+0.172$ ($[0.111, 0.231]$) for any flip and $+0.222$ ($[0.125, 0.323]$) for flip destination. Thus, d_i contains information about both decision instability and where an unstable decision moves.

This decomposition does not by itself establish that the signal is uniquely gold-specific. In particular, when the gold candidate is the nearest alternative to the realized wrong answer, a boundary-level perturbation that changes the top choice can naturally land on gold without requiring a distinct gold-specific mechanism. The present analysis does not isolate this runner-up explanation from gold-relative competition. We therefore interpret d_i as a failure-conditioned final-window candidate-competition coordinate and do not make the stronger claim that it identifies a distinct gold-specific mechanism.

Secondary diagnostics. The auxiliary analyses support the narrower evidence roles used above. Coarse full-response trace descriptors are weak alone, and hidden/attention summaries do not support a circuit-level mechanism claim. Post-query quantities separate recovered from unrecovered responses much more strongly, but they are measured on the re-query response itself and are therefore treated solely as realized-response diagnostics. Full path-type, feature-source, and post-query persistence analyses are reported in Appendices A.5, A.7.2, and A.8.

4.4 Cross-Model Synthesis

Cross-model evidence is supportive but not universal. Granite provides weaker margin-signal cross-family support, with recovery increasing from 0.0% to 22.5% across the Q1-Q5 margin bins. Llama also shows positive endpoint separation (2.3% to 56.3%), but its 37.2% correct-case false-revision rate makes recovery under $P_{\min}$ difficult to separate from re-query sensitivity; we therefore treat it

Table 4: Ordinary task accuracy and weak-re-query recovery among retained failures for the three low-false-revision models. Each cell reports accuracy / recovery (%). The table reports descriptive accuracy and residual-error recovery profiles. The SciFact contrast is the clearest supported reversal and is accompanied by Wilson intervals in the text; differences on the remaining tasks are treated as descriptive.

Task	Qwen2.5-1.5B	Gemma-2-2B	Granite-3.3-8B
TruthfulQA-MC	43.0 / 8.8	41.6 / 12.3	58.8 / 6.8
SciFact	42.9 / 27.8	54.4 / 34.8	68.2 / 12.0
BoolQ	73.4 / 15.0	81.6 / 13.0	85.0 / 22.7
ARC-Challenge	76.3 / 20.0	74.9 / 13.5	81.9 / 22.0

as a boundary case rather than evidence for the main claim. Ministral rises from 0.0% to 19.6%, but only 9 of 227 retained failures recover, leaving the estimate statistically thin. SmolLM2 is the negative margin-signal boundary: recovery changes from 2.1% to 1.1%, and its recovered-minus-unrecovered d_i interval crosses zero (Appendix A.13).

What residual-error stability adds to aggregate accuracy. To test whether residual-error stability is redundant with ordinary task accuracy, we compare the three low-false-revision models evaluated on the same task-specific input sets. Table 4 reports ordinary accuracy together with weak-re-query recovery among the residual candidate-mappable failures.

The clearest evidence that aggregate accuracy and residual-error response need not coincide comes from SciFact. Granite attains 68.2% ordinary accuracy (Wilson 95% CI: 63.1–73.0%) compared with 54.4% for Gemma (49.1–59.6%), yet only 12.0% of Granite's retained failures recover under the weak re-query (7.2–19.5%), compared with 34.8% for Gemma (27.8–42.6%). The remaining task-level differences are descriptive and are not used as independent evidence of a ranking reversal. This comparison therefore provides preliminary evidence that aggregate correctness and residual-error response can capture different empirical properties, while not establishing deployment utility, model preference, or independence from residual-set difficulty.

5 Conclusion and Limitations

We study the stability of already-observed LLM errors under a common weak cautionary re-query. The final-window gold-versus-realized-failure candidate margin sharply stratifies both answer change and gold recovery in the primary models. The relation survives a literal-answer-token audit and cannot be explained by confidence in the realized wrong candidate alone. Importantly, response decomposition shows that d_i is associated not only with whether an erroneous answer changes, but also, among non-binary changed answers, with whether the new answer is gold rather than another wrong candidate. We nevertheless do not interpret this as evidence for a uniquely gold-specific mechanism: the present analysis does not isolate gold-relative competition from every possible gold-independent boundary measure.

The framework is gold-conditioned and is intended for controlled offline failure analysis rather than deployment-time error detection. Response under the re-query is intervention-conditional: prompt variants show substantial but imperfect agreement at the individual-example level. The present evidence is limited to candidate-mappable closed-answer failures and does not establish applicability to open-ended generation, multi-token candidate answers, or long reasoning traces. The literal-answer-token and confidence-control audits are currently limited to Qwen, and all evaluated models are 8B parameters or smaller. The SciFact comparison provides preliminary evidence that residual-error response can capture information not reflected by aggregate accuracy alone; the remaining task-level comparisons are descriptive, and we do not attribute cross-model differences to any single architectural or capability factor. The resulting object is therefore a failure-conditioned decision-stability profile under a specified intervention, rather than an intrinsic per-error probability or a circuit-level mechanism.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for providing feedback on research methodology and experimental design, and assisting with the interpretation of experimental results. We have not used generative AI tools to generate synthetic datasets, formulate or prove mathematical claims, perform translation, or conduct qualitative or thematic data analysis, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for editing text for readability. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

REFERENCES

Meta AI. Llama-3.2-1b-instruct. Hugging Face model card, 2024a. URL https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct.

Mistral AI. Ministral-8b-instruct-2410. Hugging Face model card, 2024b. URL https://huggingface.co/mistralai/Ministral-8B-Instruct-2410.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 3563–3578, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International conference on learning representations*, volume 2024, pp. 32808–32824, 2024.

IBM. Granite-3.3-8b-instruct. Hugging Face model card, 2025. URL https://huggingface.co/ibm-granite/granite-3.3-8b-instruct.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12): 1–38, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can llms actually correct their own mistakes? a critical survey of self-correction of llms. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024.

Philippe Laban, Lidiya Murakhovs' ka, Caiming Xiong, and Chien-Sheng Wu. Are you sure? challenging llms leads to performance drops in the flipflop experiment. *arXiv preprint arXiv:2311.08596*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Potsawee Manakul, Adian Liusie, and Mark Gales. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 9004–9017, 2023.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Gemma Team. Gemma. 2024a. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL https://qwenlm.github.io/blog/qwen2.5/.

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

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, 2020.

Qiming Xie, Zengzhi Wang, Yi Feng, and Rui Xia. Ask again, then fail: Large language models' vacillations in judgment. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10709–10745, 2024.

Pengyue Yang, Jiawen Wen, Haolin Jin, Linghan Huang, Huaming Chen, and Ling Chen. Trust in one round: Confidence estimation for large language models via structural signals. In *Proceedings of the ACM Web Conference 2026*, pp. 9090–9100, 2026.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Appendix

A.1 Related Work

Reliability, uncertainty, and hallucination detection. A large body of work estimates or verifies LLM reliability through calibration, uncertainty estimation, self-evaluation, hallucination detection, semantic uncertainty, and consistency-based checking (Yang et al., 2026; Guo et al., 2017; Ovadia et al., 2019; Tian et al., 2023; Ji et al., 2023; Manakul et al., 2023; Farquhar et al., 2024). Particularly relevant is Kadavath et al. (2022), who show that model probabilities can provide informative estimates of whether the model's own answer is correct. This provides a direct alternative explanation for our margin-recovery relationship: failures with more favorable margins might simply be lower-confidence errors. Our analysis therefore conditions on outputs already known to be wrong and tests this explanation directly. On the replay-matched Qwen failures, task plus wrong-candidate probability yields 0.661 OOF AUROC, whereas task plus the literal gold-versus-failure margin yields 0.789; the paired improvement is 0.128 (95% CI $[0.088, 0.171]$). Thus, our claim is not that d_i is a new generic confidence statistic, but that pairwise gold-versus-realized-failure competition carries additional information about response stability under the weak re-query within an already-wrong population.

Latent predictions and representation analysis. Logit-lens, tuned-lens, latent-knowledge, and representation-engineering methods show that transformer representations can expose evolving predictions or truth-related structure (Belrose et al., 2023; Burns et al., 2022; Azaria & Mitchell, 2023; Zou et al., 2023; Marks & Tegmark, 2023). We instead use a label-aligned candidate-competition readout, deliberately narrower and tied to closed-answer settings.

Self-correction and verification. Self-correction, reflection, and verification methods improve outputs through additional feedback, iterative refinement, explicit checking, or external verification signals (Madaan et al., 2023; Shinn et al., 2023; Dhuliawala et al., 2024; Kamoi et al., 2024). Their objective is to improve final correctness or reduce hallucination. By contrast, our fixed weak re-query is an intervention condition for analyzing response stability, not a correction algorithm.

Answer revision under challenge. Recent work has shown that LLM answers can change substantially under repeated questioning or explicit challenge. Huang et al. (2024) study intrinsic self-correction without external feedback and find that reasoning performance often fails to improve after self-correction. Laban et al. (2023) study confirm-versus-flip behavior after follow-up challenges such as "Are you sure?", while Xie et al. (2024) characterize judgment vacillation under follow-up questioning. Our setting is deliberately narrower: we condition on naturally occurring errors and study how pre-query candidate competition relates separately to answer change and to the destination of a changed answer. Moreover, our $P_{\min}$ does not replay the original response in context and is therefore a cautionary re-query rather than a multi-turn challenge protocol.

A.2 Implementation Details

This appendix gives the implementation details needed to reproduce the candidate-competition records used in the main text. The purpose of these details is not to introduce additional modeling assumptions, but to make explicit how task outputs are mapped to candidates, how gold-versus-realized-failure competition is traced, and how the weak-re-query response is summarized.

A.2.1 Task prompts, answer spaces, and output parsing

Table 5 summarizes the closed-answer task formats. All task prompts request a short final answer in a fixed label space. A generated answer enters the retained failure set only when the parser maps it to a valid task candidate and the parsed candidate differs from the gold label. Correct outputs and unmappable outputs are excluded from the retained failure analysis.

For multiple-choice tasks, numeric ARC answer keys are converted to option letters when needed. For SciFact, support/refute-style labels are normalized into Entailment and Contradiction; missing, unknown, or not-enough-information-style labels are normalized into Neutral. For

BoolQ, Boolean labels are normalized to YES and NO. These parser rules are fixed before the decision-stability analysis and are not adapted per model or per example.

Table 5: Task formats, candidate spaces, and parser behavior. The retained failure population contains only valid parsed non-gold outputs with available competition traces.

Task	Candidate space	Prompt constraint	Parser rule
TruthfulQA-MC	A-D	Option letter only	First valid option letter; exact choice-text match fallback
SciFact entailment	Entailment / Contradiction / Neutral	One label only	Keyword and prefix matching after normalization
BoolQ	Yes / No	One label only	First explicit YES/NO token after normalization
ARC-Challenge	A-E	Option letter only	First valid option letter; exact choice-text match fallback

A.2.2 CANDIDATE VERBALIZERS AND FIRST-TOKEN SCORING

At each generation step, candidate scores are computed from tokenizer-dependent first-token verbalizers. For a candidate a, the verbalizer set $\mathcal{V}(a)$ contains the first token ids induced by accepted surface variants of that label. Table 6 gives the verbalizer families used for the main readout.

Table 6: Accepted label verbalizer patterns used to construct $\mathcal{V}(a)$ for the original-trace readout. The implementation tokenizes each textual variant and keeps its first token id; duplicate token ids are removed. Re-query traces use the same gold-versus-realized-failure candidate pair and the same max-first-token scoring rule with a compact final-label variant set. Accordingly, post-query persistence is interpreted as a fixed-readout diagnostic rather than as a verbalizer-invariant full-label likelihood.

Task type	Label example	Textual variants
Multiple choice	A	`A`, ` A`, `(A)`, ` A.`, `A.`
BoolQ	YES	`YES`, ` Yes`, ` yes`, `YES.`, ` Yes.`
BoolQ	NO	`NO`, ` No`, ` no`, `NO.`, ` No.`
SciFact	ENTAILMENT	uppercase, lowercase, and leading-space variants
SciFact	CONTRADICTION	uppercase, lowercase, and leading-space variants
SciFact	NEUTRAL	uppercase, lowercase, and leading-space variants

The main candidate score is the max-logit readout

$$s_t(a) = \max_{v \in \mathcal{V}(a)} \ell_t(v).$$

The same verbalizer construction is also used to store log-probability and probability readouts for diagnostic checks, but all main trajectory definitions use the max-logit score. This choice avoids summing over different numbers of verbalizers across labels and keeps the readout tied to the local next-token decision. It should not be interpreted as a full sequence probability for the semantic answer.

A.2.3 FAILURE CANDIDATE SELECTION

For a retained natural failure, the realized failure candidate a_i^- is the parsed model output when that output is a valid non-gold candidate. Thus the traced pair (a_i^+, a_i^-) corresponds to the actual observed error rather than an adversarially chosen distractor. If a record is used as a correct-control trace rather than a retained failure, the non-gold comparison candidate is selected as the highest-scoring non-gold label at the final traced step. Correct-control traces are used only for re-query-sensitivity controls and are not part of the retained failure set $\mathcal{F}$.

A.2.4 TRACE CONSTRUCTION

All model generations use deterministic decoding with temperature 0, no sampling, and task-specific short-answer prompts. The original task generation uses a maximum of 64 new tokens. For each retained failure, the tracing stage stores per-step candidate scores, candidate ranks, candidate probabilities, the gold-versus-failure margin m_t, generated-token metadata, and summary-level hidden/attention descriptors. The main paper uses only the candidate-competition quantities for its central claims; hidden/attention summaries are included as auxiliary evidence-source checks and are not used to claim a circuit-level mechanism.

The re-query response is generated with deterministic decoding, a maximum of 24 new tokens, and a task-aware valid-label stopping rule. Generation stops early when the re-query-side stopping rule detects a valid label-format completion; otherwise decoding continues until the 24-token cap. The resulting text is parsed by the re-query-side final-label parser $\tilde{\rho}$, which uses the same task label space as the original parser but is stricter about final-answer format. Re-query traces are scored using the same original candidate pair (a_i^+, a_i^-). Therefore, $\tilde{m}_t$ measures whether the original failure candidate remains locally favored in the re-query response, even if the model outputs another valid non-gold answer.

A.2.5 FINAL-WINDOW CANDIDATE-MARGIN COORDINATE

Short-answer generations may end with repeated answer markers, punctuation, delimiters, or formatting artifacts. We therefore define the final-window candidate-margin coordinate from a short backward window rather than always using the final generated token. The implementation uses radius $R = 3$:

$$\mathcal{W}_i = \{t : \max(1, T - R) \le t \le T\}.$$

Within this window, the selected step is

$$t_i^* = \arg\max_{t \in \mathcal{W}_i} \left(\max\{\pi_t(a_i^+), \pi_t(a_i^-)\}, |m_t|\right),$$

with candidate probability as the primary criterion and margin magnitude as the tie-breaker. The final-window candidate margin is $d_i = m_{t_i^*}$. This step is fixed before the weak re-query is applied and is therefore a pre-query coordinate.

A.2.6 PATH-TYPE DEFINITIONS

We retain no-flip as a separate label, rather than discarding it, so that the path-type breakdown exhausts the retained failure population and makes readout-level edge cases auditable. No-flip cases may arise from parser-verbalizer mismatch, answer-position mismatch, or cases where the parsed output is not captured by the local first-token competition readout; they are reported but not interpreted as strong failure-favored regimes.

Original path types are assigned from the original margin trajectory τ_i only. The implementation first identifies the first failure-favored step,

$$f_i = \min\{t : m_t < 0\},$$

when it exists. It then identifies the first persistent failure-favored window using $K = 2$:

$$p_i = \min\{t : m_t < 0 \text{ and } m_{t+1} < 0\},$$

when it exists. The four reported path types are:

- **No-flip**: no step has $m_t < 0$.
- **Transient**: at least one step has $m_t < 0$, but no length-K persistent failure-favored window exists.
- **Reverting**: a length-K persistent failure-favored window exists, but the trajectory later returns to $m_t > 0$.
- **Persistent**: a length-K persistent failure-favored window exists and is not followed by a return to gold-favored margins.

Table 7: Mapping between reported path-type names and implementation field values.

Reported path type	Implementation value
No-flip	`failure_without_margin_flip`
Transient	`transient_failure_preference`
Reverting	`recoverable_commitment_like`
Persistent	`persistent_failure_commitment`

For reproducibility, Table 7 maps the reported path-type names to the implementation field values used in the stored outputs.

For onset summaries, the normalized persistent-window position is assigned to early, middle, or late using cutoffs $1/3$ and $2/3$. These labels are descriptive. They are used only to summarize heterogeneity in the observed traces, not to define recovery, tune the re-query, or replace the continuous final-window candidate margin d_i.

A.2.7 Post-query persistence states

Post-query persistence is computed from the re-query response:

$$\tilde{r}_i^{\text{fail}} = \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \mathbb{I}[\tilde{m}_t < 0].$$

For compact reporting, we use three fixed response-state bins:

$$\begin{aligned} \text{disrupted}: &\quad \tilde{r}_i^{\text{fail}} = 0, \\ \text{partial}: &\quad 0 < \tilde{r}_i^{\text{fail}} < 1, \\ \text{preserved}: &\quad \tilde{r}_i^{\text{fail}} = 1. \end{aligned}$$

These bins are computed only after $P_{\min}$. They are therefore post-query displacement diagnostics, not pre-query predictors.

A.2.8 Weak and control re-queries

The main weak re-query $P_{\min}$ appends the following instruction to the original task prompt:

```
Your previous answer may be incorrect.
Reconsider your answer carefully.  Do not provide any
explanation.
Output only the final answer in exactly the same format as
before.
Answer:
```

The auxiliary re-query-control analysis uses three additional prompt variants: a repeat-only variant, a neutral verification variant, and a slightly stronger verification variant. These controls are used only to test whether the operating point is prompt-fragile. The main analysis and all reported recovery statistics use $P_{\min}$.

A.2.9 End-to-end analysis procedure

Algorithm 1 summarizes the complete analysis pipeline and separates quantities measured on the original response from those computed after the re-query.

A.3 Filtering and Retention Statistics

Table 8 reports the model-task filtering breakdown. The "Stage-1 failure" column counts non-correct ordinary-generation outputs under the task parser. This can include invalid or unmappable outputs. The "Retained" column counts records that are valid non-gold candidate outputs and have the required original and re-query competition traces. These retained rows define the analysis population; they are not estimates of each model's overall task error rate.

Algorithm 1 Decision-stability analysis for candidate-mappable natural errors

Require: Model M, task instances $\{x_i\}$, task prompts q_i, candidate sets $\mathcal{A}_i$, gold labels a_i^+, original parser ρ, re-query-side parser $\tilde{\rho}$, weak re-query $P_{\min}$

Ensure: Retained failure records with original trajectories, re-query trajectories, recovery labels, and response summaries

Initialize retained failure set $\mathcal{F} \leftarrow \emptyset$
for each task instance x_i **do**
  Generate ordinary answer $y_i \leftarrow M(q_i)$ using deterministic decoding
  Parse answer $\hat{a}_i \leftarrow \rho(y_i)$
  if $\hat{a}_i \in \mathcal{A}_i$ and $\hat{a}_i \neq a_i^+$ **then**
    Set realized failure candidate $a_i^- \leftarrow \hat{a}_i$
    Add (x_i, q_i, a_i^+, a_i^-) to $\mathcal{F}$
  end if
end for
for each retained failure $(x_i, q_i, a_i^+, a_i^-) \in \mathcal{F}$ **do**
  Re-run or trace ordinary generation under q_i
  for each original generation step $t = 1, \ldots, T$ **do**
    Compute candidate scores $s_t(a_i^+)$ and $s_t(a_i^-)$ using max-logit verbalizer scoring
    Compute margin $m_t \leftarrow s_t(a_i^+) - s_t(a_i^-)$
  end for
  Store original trajectory $\tau_i = (m_1, \ldots, m_T)$
  Assign original path type from τ_i using no-flip, transient, reverting, and persistent rules
  Select final-window coordinate t_i^* from the backward window of radius $R = 3$
  Compute final-window candidate margin $d_i \leftarrow m_{t_i^*}$
  Generate re-query response $\tilde{y}_i \leftarrow M(P_{\min}(q_i))$ using deterministic decoding
  Stop at a task-aware valid-label completion or at the 24-token cap
  Parse the re-query answer with the re-query-side final-label parser $\tilde{a}_i \leftarrow \tilde{\rho}(\tilde{y}_i)$
  Set recovery label $y_i^{\text{rec}} \leftarrow \mathbb{I}[\tilde{a}_i = a_i^+]$
  for each re-query generation step $t = 1, \ldots, \tilde{T}$ **do**
    Compute re-query candidate scores $\tilde{s}_t(a_i^+)$ and $\tilde{s}_t(a_i^-)$
    Compute post-query margin $\tilde{m}_t \leftarrow \tilde{s}_t(a_i^+) - \tilde{s}_t(a_i^-)$
  end for
  Store re-query trajectory $\tilde{\tau}_i = (\tilde{m}_1, \ldots, \tilde{m}_{\tilde{T}})$
  Compute post-query failure persistence $\tilde{r}_i^{\text{fail}} \leftarrow \frac{1}{\tilde{T}} \sum_{t=1}^{\tilde{T}} \mathbb{I}[\tilde{m}_t < 0]$
end for
Report original-trace descriptors, recovery by d_i bins, post-query persistence diagnostics, controls, and bootstrap stability

The 500-cap models do not contain exactly 2,000 attempted inputs because the available validation split for SciFact entailment and ARC-Challenge is smaller than 500 in the processed configuration. The resulting attempted totals are therefore 1,639 for Qwen2.5, Gemma2, and Granite3.3, and 800 for the 200-cap models.

In the supporting-model cache, two non-correct trace-available rows (out of 2,865 retained analysis rows) have an empty stored original-prediction string after task parsing: one Llama BoolQ row and one Ministral SciFact row. For these rare implementation edge cases, the trace stage uses the highest-scoring non-gold label as the comparison candidate a_i^-. They are kept in the released all-model cached summaries to preserve exact correspondence with the reported cross-model output files; excluding them changes only the Llama count from 433 to 432 and the Ministral count from 227 to 226, and does not affect the primary Qwen/Gemma analyses or any qualitative conclusion.

Table 8: Model-task filtering and retained failure counts. Attempted is the number of ordinary task inputs processed. Stage-1 failure counts non-correct parsed outcomes before trace-availability filtering. Retained rows are the candidate-mappable natural failures used in the main analysis.

Model	Task	Attempted	Stage-1 failure	Retained	Recovered under $P_{\min}$
Qwen2.5	TruthfulQA-MC	500	285	285	25/285 (8.8%)
	SciFact	340	194	194	54/194 (27.8%)
	BoolQ	500	133	133	20/133 (15.0%)
	ARC-Challenge	299	71	70	14/70 (20.0%)
Gemma2	TruthfulQA-MC	500	292	292	36/292 (12.3%)
	SciFact	340	155	155	54/155 (34.8%)
	BoolQ	500	92	92	12/92 (13.0%)
	ARC-Challenge	299	75	74	10/74 (13.5%)
Granite3.3	TruthfulQA-MC	500	206	206	14/206 (6.8%)
	SciFact	340	108	108	13/108 (12.0%)
	BoolQ	500	75	75	17/75 (22.7%)
	ARC-Challenge	299	54	50	11/50 (22.0%)
Ministral8	TruthfulQA-MC	200	89	89	2/89 (2.2%)
	SciFact	200	86	86	4/86 (4.7%)
	BoolQ	200	27	27	0/27 (0.0%)
	ARC-Challenge	200	25	25	3/25 (12.0%)
Llama3.2	TruthfulQA-MC	200	136	136	31/136 (22.8%)
	SciFact	200	113	113	3/113 (2.7%)
	BoolQ	200	70	70	24/70 (34.3%)
	ARC-Challenge	200	116	114	33/114 (28.9%)
SmolLM2	TruthfulQA-MC	200	157	157	8/157 (5.1%)
	SciFact	200	115	115	2/115 (1.7%)
	BoolQ	200	114	114	0/114 (0.0%)
	ARC-Challenge	200	87	85	10/85 (11.8%)

A.4 TASK-LEVEL FINAL-WINDOW MARGIN STRATIFICATION

For this task-level analysis, the five d_i quantile bins are recomputed separately within each model-task subset. This differs from the pooled model-level figure, where bins are computed over all retained failures of a model.

Table 9 reports task-level recovery stratification by the final-window candidate margin d_i for the two primary models. Q1 contains the most failure-favored margin values, whereas Q5 contains the largest margin values. Because positive d_i can reflect answer-position or parser–verbalizer mismatch, Q5 is not interpreted as a literal gold-favored decision regime. The same Q5-Q1 recovery ordering holds in every task for both primary models.

Table 9: Task-level weak-re-query recovery stratification by the final-window candidate margin d_i in the two primary models. Lift is the absolute recovery-rate difference $\mathrm{Rec}(Q5) - \mathrm{Rec}(Q1)$.

Model	Task	Retained	Overall Rec.	Q1 Rec.	Q5 Rec.	Q5-Q1 lift
Qwen2.5	TruthfulQA-MC	285	8.8%	0/57 (0.0%)	16/57 (28.1%)	28.1 pp
Qwen2.5	SciFact	194	27.8%	1/39 (2.6%)	17/39 (43.6%)	41.0 pp
Qwen2.5	BoolQ	133	15.0%	0/27 (0.0%)	10/27 (37.0%)	37.0 pp
Qwen2.5	ARC-Challenge	70	20.0%	0/14 (0.0%)	8/14 (57.1%)	57.1 pp
Gemma2	TruthfulQA-MC	292	12.3%	0/59 (0.0%)	24/59 (40.7%)	40.7 pp
Gemma2	SciFact	155	34.8%	6/31 (19.4%)	23/31 (74.2%)	54.8 pp
Gemma2	BoolQ	92	13.0%	0/19 (0.0%)	9/19 (47.4%)	47.4 pp
Gemma2	ARC-Challenge	74	13.5%	0/15 (0.0%)	7/15 (46.7%)	46.7 pp

A.5 ORIGINAL PATH-TYPE COUNTS

Table 10 reports original path-type counts and recovery under the weak re-query for the two primary models. The path types are assigned before the weak re-query and serve only as descriptive summaries of the observed competition traces. They are not the main recovery-stratification coordinate; that role is played by d_i in the main text.

Table 10: Original path types and weak re-query recovery in the two primary models. Path types are assigned before the re-query from the observed competition trace and are used descriptively, not as the main recovery-stratification coordinate.

Original path type	Qwen2.5-1.5B	Gemma-2-2B
No-flip	—	23/24 (95.8%)
Transient	46/374 (12.3%)	57/394 (14.5%)
Reverting	52/174 (29.9%)	29/111 (26.1%)
Persistent	15/134 (11.2%)	3/84 (3.6%)

These counts show heterogeneity in the observed traces: retained terminal failures are not all persistent failure-favored trajectories. Some failures never become failure-favored under the traced pairwise margin, while others contain transient, reverting, or persistent failure-favored structure. This is why the taxonomy is used descriptively rather than as a standalone predictor.

A.6 BOOTSTRAP STABILITY

Bootstrap analyses use 1,000 resamples over retained failure records. Each bootstrap sample has the same size as the original retained-failure subset and is sampled with replacement. For each resample, we recompute recovered-versus-unrecovered gaps and univariate AUROC summaries for the main signals. We orient AUROC values so that larger values indicate stronger recovered/unrecovered separation, regardless of whether high or low feature values correspond to unrecovered cases. These analyses use the complete-feature $P_{\min}$ failure subset, defined as retained failures for which all feature groups included in these diagnostic analyses are available.

Table 11: Bootstrap stability of main scalar signals on the complete-feature $P_{\min}$ failure subset. Gaps are unrecovered-minus-recovered means. Oriented AUROC is direction-normalized so that larger values indicate stronger recovered/unrecovered separation. Post-query quantities are response diagnostics.

Signal	Gap [95% CI]	Std. gap [95% CI]	Oriented AUROC [95% CI]
Final-window candidate margin d_i	-5.00 [-5.62, -4.37]	-0.93 [-1.02, -0.82]	0.777 [0.751, 0.800]
Post-query failure-favored ratio $\tilde{r}_i^{\text{fail}}$	0.89 [0.88, 0.91]	2.29 [2.21, 2.37]	0.979 [0.974, 0.984]
Combined boundary index	3.20 [3.09, 3.31]	1.94 [1.88, 2.00]	0.970 [0.962, 0.977]
Original failure-favored ratio r_i^{fail}	0.05 [0.02, 0.08]	0.20 [0.09, 0.31]	0.541 [0.511, 0.570]
Post-query candidate margin	-6.63 [-6.94, -6.33]	-1.57 [-1.62, -1.52]	0.969 [0.960, 0.978]
Failure-favored ratio delta	0.84 [0.81, 0.87]	1.89 [1.83, 1.96]	0.968 [0.961, 0.975]

As an exploratory post-query operational summary, we also report

$$C_i = z(\tilde{r}_i^{\text{fail}}) - z(d_i),$$

where $z(\cdot)$ denotes standardization over the retained-failure subset. This index combines the pre-query final-window candidate margin with post-query persistence, so it is a boundary summary rather than a pre-query predictor. It is not used to define the main results.

The bootstrap results have two intended interpretations. First, the final-window candidate margin d_i is a stable pre-query stratification coordinate: recovered failures have substantially less failure-favored final-window margins than unrecovered failures. Second, post-query quantities have larger separation because they are measured from the re-query response. Their strength is diagnostic of displacement and should not be described as pre-query prediction.

We also computed top-three signal frequencies under bootstrap effect-size ranking. When all original and post-query features are allowed, the top-three set is dominated by post-query diagnostics: $\tilde{r}_i^{\text{fail}}$, failure-favored ratio delta, and the combined boundary index appear in the top three in 100% of bootstrap resamples. This is expected because post-query quantities directly observe the response to the re-query. In a pre-query-only ranking, d_i appears in the top three in 100% of bootstrap resamples, followed by the original average margin in 100%, while persistence and auxiliary internal summaries appear less consistently. This is why the main text separates pre-query stratification from post-query diagnosis.

A.7 Re-Query Controls and Evidence-Source Checks

A.7.1 Correct-case and prompt-variant controls

Table 12 reports the $P_{\min}$ operating point for models with correct-case controls. Correct-case controls are originally correct task outputs. A false revision occurs when the weak re-query changes an originally correct answer into a non-gold answer. These controls are used to assess re-query sensitivity, not to define the retained failure set.

Table 12: $P_{\min}$ operating point and correct-case controls. Recovery rates are from the main retained-failure $P_{\min}$ run reported in the paper. False-revision and kept-correct rates are computed on the separate originally correct control subsets used in the re-query-control analysis; denominators are shown explicitly in the table.

Model	Failure cases	$P_{\min}$ rec.	False revision	Kept correct
Qwen2.5-1.5B	682	113/682 (16.6%)	49/564 (8.7%)	515/564 (91.3%)
Gemma-2-2B	613	112/613 (18.3%)	57/530 (10.8%)	473/530 (89.2%)
Granite-3.3-8B	439	55/439 (12.5%)	25/443 (5.6%)	418/443 (94.4%)
Llama-3.2-1B	433	91/433 (21.0%)	113/304 (37.2%)	191/304 (62.8%)

Table 13 distinguishes aggregate prompt sensitivity from instance-level agreement. Recovery rates are similar across variants for the three low-false-revision models, but the recovered-example sets are not identical. The moderate-to-high agreement indicates a stable component of failure susceptibility

while also showing that recovery remains conditional on the particular re-query cue. We therefore do not interpret recoverability as an intervention-independent intrinsic property of an individual error.

Table 13: Prompt-variant sensitivity on the matched prompt-control subset. Recovery-rate and false-revision ranges summarize aggregate variation across four variants. Agreement columns summarize pairwise recovery-label agreement between $P_{\min}$ and each of the three alternative variants.

Model	Rec. range	False-rev. range	κ range	Jaccard range
Qwen2.5-1.5B	3.8 pp	2.0 pp	0.64–0.72	0.54–0.62
Gemma-2-2B	2.8 pp	2.3 pp	0.78–0.79	0.69–0.71
Granite-3.3-8B	2.5 pp	0.2 pp	0.60–0.71	0.48–0.60
Llama-3.2-1B	4.4 pp	4.9 pp	0.40–0.71	0.34–0.61

A.7.2 Feature-source ablations

The feature-source checks in Table 14 use simple cross-validated logistic summaries on the same complete-feature $P_{\min}$ failure subset. They are diagnostic analyses only. They are not used to define path types, tune the re-query, select examples, or construct the reported recovery statistics.

Table 14: Feature-source ablations on the complete-feature $P_{\min}$ failure subset (n = 2,167; 373 recovered, 1,794 unrecovered). AUROC values are bootstrap means with 95% intervals. Post-query feature sets are response diagnostics and must not be interpreted as pre-query predictors.

Feature source	AUROC [95% CI]	Interpretation
Original-side only	0.774 [0.750, 0.798]	Pre-query stratification
Post-query only	0.986 [0.982, 0.990]	Post-query displacement diagnostic
Original + post-query	0.986 [0.981, 0.990]	Operational boundary summary
Original + post-query logit/margin features	0.987 [0.983, 0.990]	Mixed pre-/post-query diagnostic
Internal summaries only	0.596 [0.563, 0.627]	Weak auxiliary evidence
Original internal only	0.603 [0.572, 0.630]	Weak pre-query auxiliary evidence
Post-query margin summaries only	0.985 [0.981, 0.989]	Post-query diagnostic
Original trace descriptors only	0.551 [0.520, 0.583]	Descriptive trace summaries are weak alone
All available core	0.988 [0.983, 0.991]	Exploratory upper-bound check

These checks delimit the evidence roles used in the main paper. Original-side features contain useful pre-query information, and the final-window candidate margin d_i accounts for most of this separation. Coarse trace descriptors alone are weak as standalone recovery predictors, which is why the main text uses them only to describe heterogeneity rather than as the primary recovery coordinate. Post-query features are much more separable, but only because they are measured from the re-query response. Internal hidden/attention summaries are weak by themselves, so the present evidence supports a candidate-competition analysis rather than a circuit-level mechanistic explanation.

A.8 Additional Post-Query Persistence Details

Table 15 gives the post-query persistence breakdown for originally persistent paths in the two primary models. All recovered cases fall in the disrupted post-query state, whereas cases with nonzero post-query persistence remain unrecovered in both primary models. Because recovery and post-query persistence are measured on the same re-query response, this pattern is descriptive rather than causal.

Table 15: Post-query persistence among originally persistent paths in the two primary models. Disrupted means $\tilde{r}_i^{\text{fail}} = 0$; nonzero persistence means $\tilde{r}_i^{\text{fail}} > 0$; preserved means $\tilde{r}_i^{\text{fail}} = 1$.

Model	Original persistent	Disrupted rec.	Nonzero-persistence rec.	Preserved subset
Qwen2.5-1.5B	134	15/15	0/119	117/119
Gemma-2-2B	84	3/3	0/81	72/81

This table is not a pre-query prediction result. It is a response-side diagnostic: when the weak re-query succeeds, the original failure candidate is displaced in the re-query candidate-competition trace; when the re-query trace preserves the original failure candidate, recovery is not observed in these primary-model persistent-path cases.

A.9 Pre-Query Confidence and Competition Baselines

These baselines are not intended to show that d_i uniquely outperforms every confidence-style scalar. They compare stored pre-query readouts on the full retained-failure populations using the original final-window coordinate. This analysis is therefore distinct from the 607-case replay-matched audit in Appendix A.10, which uses the literal answer token and explicitly controls for task identity and wrong-candidate confidence.

Table 16: Pre-query baselines for recovered/unrecovered separation on retained failures in the two primary models. All signals are computed before $P_{\min}$. AUROC is oriented so larger values indicate stronger separation.

Signal	Qwen2.5 AUROC	Gemma2 AUROC
Final-window candidate margin d_i	0.800	0.859
Failure-candidate probability at t_i^*	0.728	0.795
Gold-candidate probability at t_i^*	0.609	0.547
Pairwise normalized probability margin	0.800	0.859
Gold-vs-failure token-rank gap at t_i^*	0.763	0.819
Original failure-favored ratio r_i^{fail}	0.538	0.618
Original path type only	0.627	0.698

Among these stored readouts, the final-window logit margin d_i provides the strongest scalar separation in both primary models. The pairwise normalized probability margin has identical AUROC because it is a strictly monotone transformation of d_i; we retain it only as a numerical sanity check rather than as an independent robustness test. This comparison alone does not establish independence from confidence; that question is tested directly by the replay-matched confidence-control audit in Appendix A.10. For failure-candidate probability and r_i^{fail}, AUROC is direction-normalized because larger raw values correspond to stronger unrecovered-side evidence. Original path type is encoded ordinally as persistent, transient, reverting, and no-flip.

A.10 Short-Answer and Confidence-Confound Audits

We conducted two additional audits on the 682 retained Qwen2.5-1.5B failures to test whether the main recovery stratification could be explained by the short-answer trajectory construction or by ordinary confidence in the realized wrong candidate.

Literal answer-token audit. We reconstructed token-level traces and retained the 607/682 cases (89.0%) whose replayed output mapped to the same failure candidate as in the original analysis. We do not attribute the 75 replay mismatches to a single verified cause; to avoid mixing execution differences into this audit, all literal-token and confidence-control analyses are restricted to the 607 replay-matched failures. In all 607 matched cases, the answer begins at the first generated token. Under 5-fold out-of-fold evaluation, the original final-window coordinate yields 0.800 AUROC for recovered-versus-unrecovered separation, while the gold-versus-realized-failure margin at the literal answer token yields 0.794. Thus, the main Qwen recovery stratification remains after removing all post-answer continuation from the readout.

Confidence-control audit. On the same 607 failures, task identity alone yields 0.648 OOF AUROC. Adding realized failure-candidate probability gives 0.661, while adding the literal gold-versus-failure margin gives 0.789. Combining task identity, failure-candidate probability, and the margin gives 0.807. The margin improves over failure-candidate probability by 0.128 AUROC under paired evaluation, with a bootstrap 95% confidence interval of $[0.088,\ 0.171]$.

As a second control, within each task we divide failures into quintiles of realized failure-candidate probability and split each bin at the median literal answer-token margin. The higher-margin half

Table 17: Task-level short-answer audit on the replay-matched Qwen failures. All analyzed responses begin with the task answer at the first generated token. Coordinate match is the fraction for which the original final-window coordinate coincides with that literal answer token.

Task	Mean generation length	Coordinate match
TruthfulQA-MC	2.11	98.4%
SciFact	7.21	49.1%
BoolQ	2.00	100.0%
ARC-Challenge	2.00	100.0%
Overall	-	85.2%

has greater recovery in 16 of 20 bins, with a weighted recovery-rate difference of $+21.3$ percentage points. These analyses do not show that the margin is independent of confidence; rather, they show that confidence in the realized wrong candidate alone does not account for the observed recovery stratification.

A.11 Scope of the Final-Window Candidate Margin within Retained Failures

The final-window candidate margin d_i is not used to distinguish correct from incorrect outputs. It is computed only after a record has already been retained as a candidate-mappable natural failure under the construction in Section 2.1, and the main analysis concerns variation *within* this retained-failure population. The d_i-binned recovery curves in Section 4.2 therefore test whether final-window gold-versus-realized-failure candidate competition stratifies recovery under the weak re-query among natural errors, not whether d_i provides a generic confidence score over all model outputs.

Two implications follow. First, Q1 contains failures with the most failure-favored values of the final-window candidate margin, and its near-zero recovery identifies a region of high response stability under the weak re-query. Second, Q5 contains the largest values of this readout, but positive d_i values should not be interpreted as literal gold-favored decisions: under greedy decoding, such cases indicate that the selected final-window coordinate does not coincide cleanly with the emitted answer token or its verbalizer. This issue is most visible in Gemma's 24 no-flip readout-edge cases, 23 of which recover. Even if all 24 are removed from the Q5 endpoint, the remaining Q5 recovery is at least $51/99 = 51.5\%$, so the primary endpoint separation does not depend entirely on these readout-edge cases. We therefore treat d_i as a final-window candidate-competition coordinate, with the literal-answer-token audit providing the cleaner answer-token check.

A.12 Robustness to Stored Candidate Readouts

Table 18: Robustness of recovery stratification to alternative stored pre-query candidate readouts. Q1 and Q5 denote the lowest and highest quantile bins under each oriented readout coordinate; reported lifts are $\mathrm{Rec}(Q5) - \mathrm{Rec}(Q1)$.

Readout coordinate	Qwen Q1→Q5	Qwen lift	Gemma Q1→Q5	Gemma lift
Max first-token logit margin d_i	0.0→39.4	39.4 pp	4.9→60.2	55.3 pp
Max first-token probability margin	2.2→37.2	35.0 pp	0.0→52.0	52.0 pp
Pairwise normalized probability margin	0.0→39.4	39.4 pp	4.9→60.2	55.3 pp
Inverted failure-candidate probability	2.9→33.6	30.7 pp	0.8→41.5	40.7 pp
Gold-vs-failure token-rank gap	0.7→38.7	38.0 pp	5.7→53.7	48.0 pp

The recovery ordering is qualitatively preserved by several alternative stored candidate readouts. The pairwise normalized probability margin is a strictly monotone transformation of d_i and therefore reproduces its bins by construction; it is shown only as a sanity check. The distinct probability-margin, inverted failure-probability, and rank-gap readouts preserve the same qualitative Q1-Q5 increase. These checks do not establish robustness to full sequence likelihood or additive verbalizer-mass scoring; the present traces store max-verbalizer readouts and derived probability/rank summaries, not per-verbalizer probability mass for every accepted surface form.

Table 19: Cross-model summary of the fixed re-query operating point, pre-query final-window margin stratification, and post-query trajectory persistence. Original-side columns use the final-window candidate margin d_i. Post-query columns report recovery within disrupted, partial, and preserved response states after $P_{\min}$; these quantities are diagnostic only and are not pre-query predictors.

	Operating point			Pre-query final-window margin		Post-query persistence			
Model	Ret.	Rec.	Rec.%	d_i Q1→Q5	Δd_i 95% CI	Disrupted	Partial	Preserved	$\Delta\tilde{r}_i^{\text{fail}}$ 95% CI
Qwen2.5	682	113	16.6	0.0→39.4	5.96 [5.19, 6.86]	76/85	37/86	0/511	-0.84 [-0.87, -0.81]
Gemma2	613	112	18.3	4.9→60.2	7.94 [6.35, 9.56]	111/117	1/24	0/472	-0.96 [-0.98, -0.95]
Granite3.3	439	55	12.5	0.0→22.5	2.80 [2.01, 3.64]	46/53	9/38	0/348	-0.91 [-0.94, -0.88]
Llama3.2	433	91	21.0	2.3→56.3	2.70 [2.22, 3.18]	89/138	2/61	0/234	-0.79 [-0.83, -0.75]
SmolLM2	471	20	4.2	2.1→1.1	-0.46 [-1.96, 0.61]	19/45	1/90	0/336	-0.83 [-0.88, -0.77]
Ministral8	227	9	4.0	0.0→19.6	4.72 [3.64, 6.29]	7/7	2/8	0/212	-0.84 [-0.99, -0.64]

Notes. Retained failures are naturally occurring, candidate-mappable records. d_i Q1→Q5 reports recovery from the most failure-favored to the largest final-window candidate-margin bins. Δd_i and $\Delta\tilde{r}_i^{\text{fail}}$ are recovered-minus-unrecovered bootstrap gaps with 95% intervals. Post-query states are computed from the re-query response and are diagnostic only.

A.13 Full Cross-Model Boundary Summary

Table 19 consolidates the operating point, pre-query margin stratification, and post-query diagnostic states for all six evaluated models.

A.14 Compute Resources

All experiments were run on a Linux workstation with two NVIDIA GeForce RTX 5090 GPUs, each providing approximately 32GB of GPU memory, an AMD Ryzen Threadripper PRO 7965WX 24-core CPU with 48 logical threads, and 125GB of RAM. The software environment used NVIDIA driver 570.211.01 and CUDA 12.8 on a Linux 6.8.0-101-generic system.

The experiments used open-weight Hugging Face models through PyTorch and Transformers. Generation was deterministic throughout the pipeline, with temperature 0, no sampling, 64 maximum new tokens for ordinary task inference, and 24 maximum new tokens for the weak re-query. No model was fine-tuned. After generation, all downstream analyses were computed from cached stage-wise JSON/JSONL outputs; these post-processing stages were lightweight compared with model inference. All reported analyses were computed from cached stage-wise outputs, and the implementation details required to reproduce the analysis pipeline are described in the Appendix.

A.15 Existing Assets and Licenses

The experiments use public benchmark datasets and publicly released open-weight language models. The benchmark tasks are used only for evaluation and failure-state analysis, and no private or newly collected human-subject data is introduced. The evaluated models are used only for deterministic inference and trace extraction; no model is fine-tuned, redistributed, or modified. Dataset sources are cited where the benchmarks are introduced, and model sources and checkpoint identifiers are cited in Section 3.1. We follow the corresponding public license or usage terms.