FAILURE BUYS A TURN: CORRECT ANSWERS ARE ITS HIDDEN CASUALTIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Many LLM evaluations retry only the answers that a validator, such as unit tests, rejects, so neither scores nor logs show what that turn would do to answers that pass. We prove that such logs leave the value of this policy anywhere between zero and the first-attempt score when the validator is the task's own scorer. In unscored shadow runs that give accepted answers the same turn, the largest losses are output-delivery failures: the correct answer is still in context, but the next reply omits it. On HumanEval, "Your submission has been received" makes Gemma-3-12B drop 129 of its 137 correct answers, while "Your submission was incorrect" on the same conversations drops five. When the reply replaces the first answer, eight of ten open models have a message that destroys 19% to 94% of their correct answers. Adding sentences reveals two routes to recovery: after an acknowledgement, a request to submit restores nearly every answer, but after our review request Qwen3-8B still loses 80 of 139 until the exact output format is requested. Without a validator, deleting the final request to restate the answer raises Llama-2-70b-chat's GSM8K self-correction loss from 12 to 36 points, while deleting only its box changes little. Retry evaluations should report their continuation message and audit accepted answers.

1 INTRODUCTION

Large language models are increasingly evaluated and deployed inside loops that check an answer and ask again when the check fails. A coding agent runs unit tests on its code, structured-output libraries validate a reply against a schema and ask again by default (PydanticAI contributors, 2026; Instructor contributors, 2026; Guardrails AI contributors, 2026; LangChain contributors, 2026), and some agent benchmarks retry an answer that fails the task's own test. The Inspect implementation of Cybench (Zhang et al., 2025; Inspect Evals contributors, 2026), for instance, sends "Your submission was incorrect. Please proceed and attempt to find the correct answer." We call such a check a *semantic validator* and the loop around it a *retry harness*.

A retry harness spends extra computation only where its validator objects. A rejected answer buys the model another turn, an accepted answer ends the episode, and the harness reports one number, the final score. We call this rule, retry only what the validator rejects, the *gate*. The gate allocates compute by the model's own outputs, so the final score mixes the quality of the first answers with the value of the turns that rejections buy.

The gate also leaves a blind spot. Answers that pass the check never receive the extra turn, so even a log of every first answer and every rejection cannot show whether that turn would keep or destroy them. We call these answers the *accepted stratum*. The blind spot matters whenever the gate changes: a stricter validator sends some accepted answers into the retry, and self-refinement or intrinsic self-correction continues every answer (Madaan et al., 2023; Huang et al., 2024). A correct answer might be expected to survive a polite request to continue.

It need not. On the same correct HumanEval answers, Gemma-3-12B loses 129 of 137 solutions when told its submission "has been received," but only five when told it "was incorrect" (Figure 1). The messages differ in three words. Only three of the 129 lost answers are rewritten into wrong code; in the others the next reply has no code, and Gemma-3-12B usually says that it is ready for the next task. Self-correction and sycophancy studies focus on answers that change (Huang et al., 2024; Laban et al., 2023; Sharma et al., 2024); this failure is about delivering the answer at all.

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

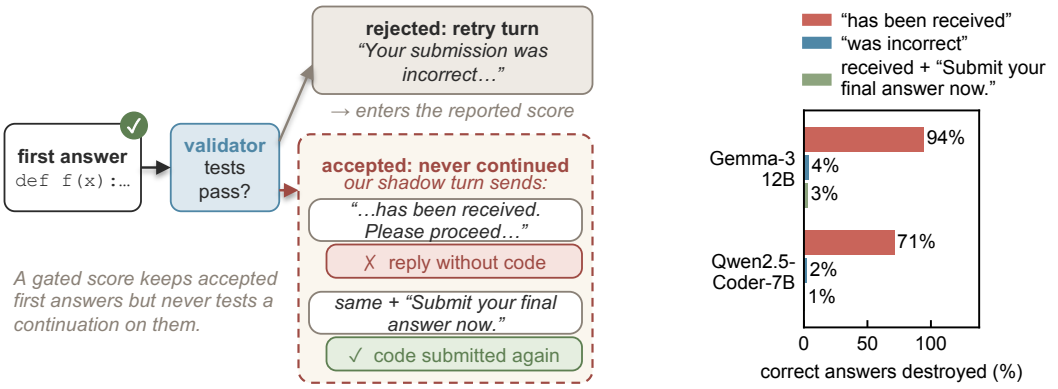

Figure 1: What a gated score cannot see. Left: a validator sends a retry turn only to rejected answers, so no reported number shows what the harness's message does to accepted ones; our shadow turn sends it anyway. Right: on identical first histories of 137 and 145 accepted HumanEval answers, the acknowledgement (red) destroys most, the false verdict (blue) few, and one added request (green) restores them.

This paper studies the accepted stratum directly. We derive what gated logs can reveal and measure the stratum with *shadow continuations*, which give every answer the extra turn but keep it out of the reported score (Figure 1, left). We test ten open models from six families and two API models on HumanEval, whose unit tests decide correctness, and extend the measurement to a text-to-SQL benchmark whose validator only checks that a query runs and returns rows, and to GSM8K without any validator. Our main contributions are:

- We prove that when the validator is the task's own scorer, gated logs leave the value of the gate anywhere from zero to the first-attempt score, and for the same reason cannot show how well the validator targets the extra turns (Section 2).
- Measuring the accepted stratum of ten open models, we find that eight have a plausible continuation that destroys 19–94% of their correct answers when the reply replaces the first answer, and that the damaging message changes from model to model. Keeping the first answer when a reply has no code removes most losses but not changed answers (Section 3).
- Adding one sentence at a time to the same conversations, we separate two routes to a missing output. After an acknowledgement, a generic request to submit restores almost every answer, but after our review request the Qwen models we tested recover only once the exact output format is requested (Section 4). Without a validator, deleting the final request to restate the answer in a box raises the observed 12-point self-correction loss of Llama-2-70b-chat on GSM8K to 36 points, while deleting only the box raises it to 14 (Section 5.3).

Overall, the continuation message and the output format are part of the evaluated policy, and their effect on accepted answers has to be measured rather than read from gated logs. Our account is behavioral: it describes what the text elicits, not an internal state of the model.

2 The Stratum a Gated Harness Never Observes

Three policies. A first answer earns reward $z_1 \in [0, 1]$. A validator accepts it ($T = 0$) or rejects it ($T = 1$), with rejection rate $p = \Pr(T = 1)$. We write $z_2 \in [0, 1]$ for the reward after one continuation with a fixed message and $D = z_2 - z_1$ for the effect of that extra turn. Keeping every first answer scores $V_1 = \mathbb{E}[z_1]$. A *gated* harness continues only rejected answers, like the Inspect implementation of Cybench quoted in Section 1, and scores $V_{\text{gated}} = V_1 + p\,\mathbb{E}[D \mid T = 1]$. An ungated harness sends the same message to every answer and scores $V_{\text{ungated}} = V_1 + \mathbb{E}[D]$, so the two differ only on the accepted answers. The gate's value is gated minus ungated, not the retry's gain: $G = V_{\text{gated}} - V_{\text{ungated}} = -(1 - p)\,\mathbb{E}[D \mid T = 0]$.

Table 1: Three tasks, three settings: an oracle validator, a proxy validator, and none. What the validator accepts decides which answers a gated harness never continues. Appendix E records when each experiment was registered and which followed earlier results.

Task	Validator accepts	Scored by	We measure
HumanEval, 164 (§3–4)	code that passes the unit tests: the *oracle* regime	the same tests	correct answers of twelve models after one extra turn
BIRD, 997 (§5)	read-only SQL that runs and returns rows: the *proxy* regime	official execution match	accepted answers, right or wrong, after one extra turn
GSM8K, 200 (§5.3)	nothing is checked: every answer is continued	the last boxed number	accuracy after one round of self-correction

Two regimes. In the *oracle* regime rewards are binary and the validator is the task's own scorer, so $T = 1$ exactly when $z_1 = 0$ and every accepted answer has $z_1 = 1$, as on HumanEval with its unit tests (Table 1). In the *proxy* regime the validator is weaker than the scorer. On the text-to-SQL benchmark BIRD, the validator accepts any query that parses, runs and returns rows, right or wrong.

Even rich logs are insufficient. Suppose a gated evaluation releases its final and first-attempt scores, plus T and z_1 for every task. It then reveals V_{gated}, V_1 and the retry's effect $\mathbb{E}[D \mid T = 1]$. It never reveals $m_0 = \mathbb{E}[z_2 \mid T = 0]$, the mean reward after a continuation on the accepted stratum $\{T = 0\}$. The logs are thus identical whether another turn keeps or destroys every accepted answer.

Value of the gate. Since $G = (1 - p)\big(\mathbb{E}[z_1 \mid T = 0] - m_0\big)$, bounded rewards give the following.

Proposition 1 (Value of the gate)**.** *Given these logs and no restriction on z_2 beyond $z_2 \in [0, 1]$, $G \in (1 - p)\,[\,\mathbb{E}[z_1 \mid T = 0] - 1,\ \mathbb{E}[z_1 \mid T = 0]\,]$, and both ends are attainable. In the oracle regime this is $G \in [0, V_1]$. If accepted answers are mixed ($0 < \mathbb{E}[z_1 \mid T = 0] < 1$), it contains both signs.*

This is the worst-case bound of partial identification (Manski, 2003), applied to the one quantity that a gated design never reveals. In the oracle regime G is V_1 times the share of accepted answers the continuation destroys, one minus a retention rate (Yang et al., 2024). Gated logs omit this share, yet the gated score itself remains valid. We measure G in Section 3 and find values near both ends.

Retry as allocation. The retry's gain splits into the value of the same number of turns given at random, the value of feedback, and the validator's *targeting* $\mathrm{Cov}(T, D)$, positive when rejected answers gain more from a turn than accepted ones (Appendix F). The missing m_0 also leaves targeting unidentified. Let $a = \mathbb{E}[D \mid T = 1] + \mathbb{E}[z_1 \mid T = 0]$.

Proposition 2 (Sharp identified set for targeting)**.** *Under the same conditions and $0 < p < 1$, $\mathrm{Cov}(T, D) \in p(1 - p)\,[\,a - 1,\ a\,]$. The set is sharp and contains zero if and only if $0 \le a \le 1$.*

The reason is that $\mathrm{Cov}(T, D) = p(1 - p)(a - m_0)$. In the oracle regime $\mathbb{E}[z_1 \mid T = 0] = 1$, so the set's lower end is $(1 - p)$ times the published retry gain. The design alone thus makes the targeting covariance nonnegative, and positive whenever the retry helps. Its sign says nothing about the validator. Appendix B gives both proofs and checks Proposition 2 on 1,538 populations.

3 Measuring the Accepted Stratum

3.1 Protocol

We use the 164 HumanEval problems (Chen et al., 2021) with their own unit tests as both validator and reward, so the accepted answers are exactly the correct ones. Six open-weight instruct models from four families answer each problem once and greedily, under a system message that asks for one complete function in a fenced block. Four more models answer the same problems with the same prompts and scoring: Llama-3.1-70B, the next Gemma generation (Gemma-4-12B), and two API models with thinking disabled, GLM-5.3-flash and DeepSeek-flash. Two models from families added last, OpenAI's reasoning model gpt-oss-20b and Microsoft's Phi-4, ran with batched decoding (Appendix E). We then give every problem one shadow turn under each of five messages (Appendix A)

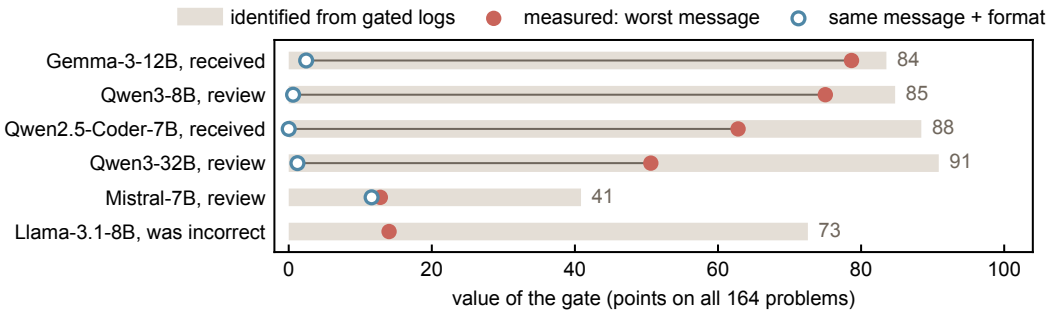

Figure 2: The gate's value on HumanEval, for the six original models. Gated logs identify it only to the grey interval $[0, V_1]$ (Proposition 1), whose upper end is printed; shadow turns measure it. Red: each model's worst of the five messages; blue: the same message with the output format restated (none exists for *was incorrect*). Points use all 164 problems; the interval is an identified set, not a confidence interval.

and score the reply in place of the first answer, as a harness grading the latest output does. A correct answer is lost if the reply fails. *Was incorrect* is Cybench's Inspect retry message verbatim, which on accepted answers is a false accusation. *Received* replaces the verdict with "has been received," a neutral acknowledgement. *Received + format*, *review* and *review + format* restate the format, ask for a review, or do both. Every later experiment was registered before its first call, except those in Appendix E.

3.2 Losses and Gate Values

Most models have a message that destroys their correct answers. Figure 3 shows the share of correct answers that each message destroys, and the most damaging message changes from model to model (exact 95% intervals in Appendix D). Gemma-3-12B loses 94% of its correct solutions under *received* and at most 4% under any other message. Qwen3-8B is the mirror image, losing 1% under *received* and 88% under *review*. *Review* also costs gpt-oss-20b 62.1% and the API model GLM-5.3-flash 12.7%. Three models lose little: at most 2 of 157 correct answers for Gemma-4-12B, 4 of 143 for Phi-4 and 9 of 158 for DeepSeek-flash. Replays match the stored code of the six original models' first answers and 276 losses, and a permissive extractor recovers one loss (Appendix C).

Measured gate values approach both ends of the interval. For Gemma-3-12B under *received*, the gate is worth 78.7 points on all 164 problems (Figure 2), since the ungated turn destroys 129 of 137 accepted answers (94.2%). Once the message also restates the format, the gate is worth only 2.4 points. A gated evaluation reveals neither number.

3.3 Gate Changes and Audits

Changing the gate exposes the accepted stratum. Sending our *review* request once to every answer takes Qwen3-8B from 84.8% at the first attempt to 11.6% on HumanEval, while *review + format* leaves it at 84.8% (Figure 10). Removing the gate under *received* drops Gemma-3-12B from its gated 84.8% to 6.1%. Keeping the earlier answer whenever a reply has no code block cuts this loss to 1.8 points. The rule cannot stop changed code that fails, which costs Mistral-7B and both Llama models 19–31% of their accepted answers (Appendix Figure 9). Suppose a stricter validator also rejected a random 21 of Gemma-3-12B's 137 accepted answers. Averaged over 200 seeded random draws, this costs 12.0 points under *received* but 0.3 under *review + format*. Random routing yields the stratum's average loss. A real stricter validator instead adds a selected set C of accepted answers to the retry, changing the score by $\Pr(C)\,\mathbb{E}[D \mid C]$. Running the new validator on the logged first answers finds C, so an audit should continue C itself.

A probe of the deployed message. The missing m_0 can be measured by continuing accepted answers with the message a harness deploys. With accepted problems drawn without replacement, 95% detection at the worst rates in Figure 3 takes two calls for Gemma-3-12B, three for Qwen2.5-

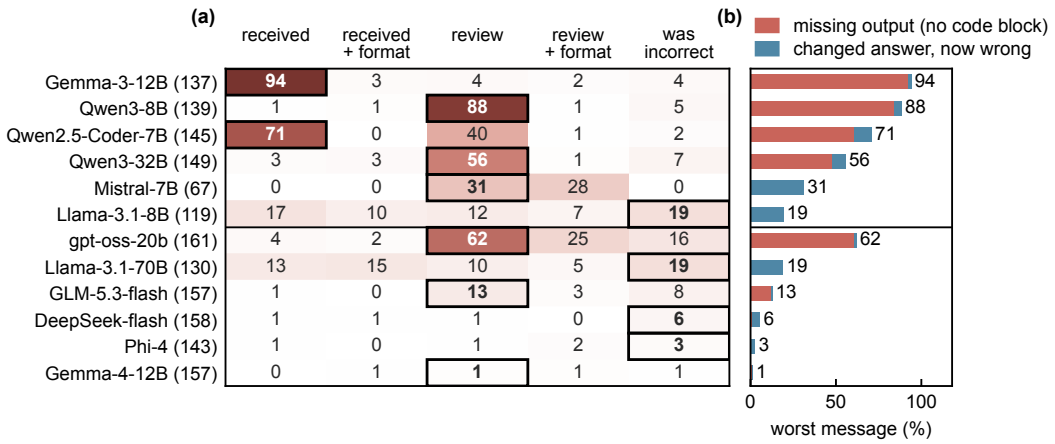

Figure 3: Accepted HumanEval answers destroyed by one extra turn (%), greedy. (a) Rows are models, with their accepted answers in parentheses; columns are the five continuation messages (Appendix Figure 7), where "+ format" states the exact output format; shading and numbers give the % destroyed, and the boxed worst cell is chosen from the counts. Models below the line were added later; the API models (GLM-5.3-flash, DeepSeek-flash) ran with thinking disabled and gpt-oss-20b at medium reasoning effort. (b) The worst message's losses: replies without a code block, or changed code that fails.

Coder-7B and 14 for Llama-3.1-8B. Twenty calls detect any fixed message that destroys at least 14% of accepted answers, and each further message needs its own budget. A loss-free probe cannot certify safety, since seeing one loss from our low-loss messages would take 86–138 calls.

4 Two Channels: Missing Outputs and Changed Answers

Most damage is a missing output, not a changed answer. Answers are lost through two channels: the reply omits the required output (*format*) or submits a changed answer that is now wrong (*semantic*). A reply cut off by the token budget counts as an omission. Figure 3b splits each model's worst loss this way. Even a permissive extractor finds a working function in almost none of the replies without a fenced block (Section 3). Almost every loss is a missing output for Gemma-3-12B, Qwen3-8B, GLM-5.3-flash and gpt-oss-20b, and most are for Qwen2.5-Coder-7B and Qwen3-32B. For Mistral-7B and both Llama-3.1 models the losses are semantic. A harness that keeps the first answer whenever the reply has no code would avoid the omissions, but not the semantic losses of 19–31% for Mistral-7B and both Llama-3.1 models. The replies without code take two forms (Figure 11). Under *received*, 123 of Gemma-3-12B's 126 ask for or wait for a next task. Under *review*, every such reply from Qwen3-8B and GLM-5.3-flash reviews the code in prose and calls it correct or final.

Two routes to a missing output. Within the format channel, we reuse the same complete first histories, add sentences to the message and count replies that still lack the required output (Figure 4). In our tests, a request to submit closes one route, and only the exact output format closes the other. After an acknowledgement, the generic request "Submit your final answer now" restores almost every correct answer: Gemma-3-12B's losses fall from 129 to 4 and Qwen2.5-Coder-7B's from 103 to 1 (paired $p < 10^{-30}$ each). After the review message, which already asks for a final answer, the outcome depends on the model. With the generic request, Qwen3-8B still loses 80 of its 139 answers, 76 of them without code. Adding the exact Python output format removes all 80 ($p = 1.7 \times 10^{-24}$). We did not test whether the complete function or the fence is needed. Both routes recur on SQL, and the first in math (Sections 5.1 and 5.3). Our + *format* messages add the same exact format. For GLM-5.3-flash, putting it in place of the review's final-answer request cuts losses from 20 to 4. No model loses significantly more from restating the format, though this is not an equivalence test. The semantic channel remains, small for most models but 28% for Mistral-7B under *review* + *format*.

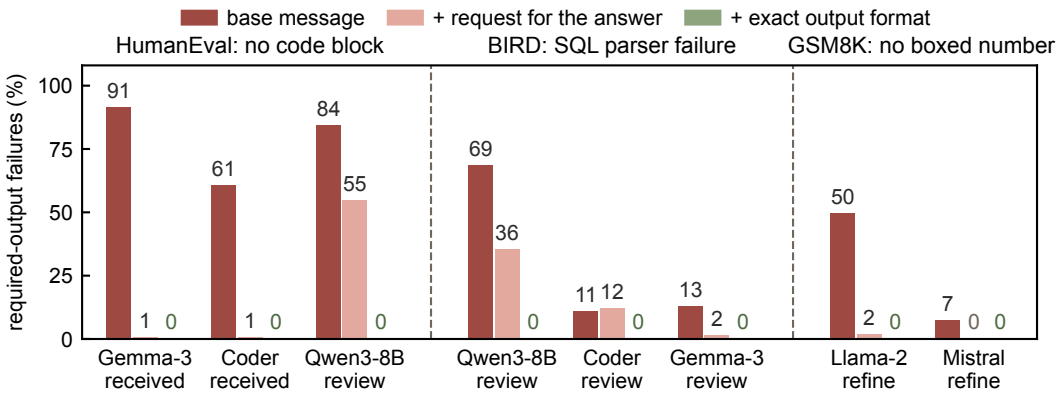

Figure 4: Two routes to a missing output: replies without each task's required output, on the same first histories (137, 145, 139 HumanEval answers; 200 BIRD histories each; 115, 125 correct GSM8K answers). Bars add "Submit your final answer now." and then the exact output format; the review message already asks for a final answer. GSM8K bars use the refine prompt R of Huang et al. (2024) without its last sentence (R^-), without its box (R_{nobox}), and in full. Zero means no output failure, not all answers correct. Coder is Qwen2.5-Coder-7B.

An accusation can keep a model submitting. Following the sycophancy literature, we registered that for every model the accusation would destroy at least as many correct answers as the acknowledgement. It did not. For the two models hit hardest by *received*, the accusation protects the answers that the acknowledgement destroys (Figure 1). Gemma-3-12B keeps 132 of 137 and Qwen2.5-Coder-7B 142 of 145 ($p < 10^{-29}$ each). A rerun on identical complete first histories reproduces both counts exactly (Appendix E). For the other models the accusation costs as much or more, significantly for the two API models and gpt-oss-20b ($p \leq 0.008$). The cause is not settled. *Received* also asks the model to keep looking, yet on problems the tests rejected it still leaves Gemma-3-12B without code on 17 of 27. *Was incorrect* never does so for any of the six original models.

5 Scope of the Damage

5.1 A Proxy Validator: SQL

On SQL, different models fail. On a census of 997 BIRD tasks (Li et al., 2023; Appendix C), the validator accepts any read-only SQL query that runs and returns rows, including wrong ones (Table 9), so a continuation can both destroy and rescue accepted answers. Under *received*, the ranking of models changes (Figure 5a). Mistral-7B keeps all its correct HumanEval answers but loses all 229 correct BIRD answers, while Qwen2.5-Coder-7B's loss falls from 71% to none (Fisher exact $p < 10^{-67}$ for both). Mistral-7B's lost replies keep the SQL block but change the query, and restating the format leaves 227 of 229 lost. For Qwen3-8B, *review* destroys 374 of its 484 correct BIRD answers, and 559 of its 815 continuations fail the official SQL parser.

On SQL, the added sentences show three patterns. First, output delivery. We add the request and then the exact SQL format to the review on 200 accepted histories per model, selected without reading any continuation outcome. Parser failures are 137, 71 and 0 for Qwen3-8B, 22, 24 and 0 for Qwen2.5-Coder-7B, and 26, 3 and 0 for Gemma-3-12B (Figure 4). Every failure left after the request lacks an SQL block. Both Qwen models need the exact format. Qwen3-8B's registered verdict stays *mixed*, since the request left 71 failures and the registered threshold for needing the format was 100. Gemma-3-12B, a family chosen after these results, needs only the request, as registered. The request repairs 25 of its 26 failures and breaks 2 others ($p = 6 \times 10^{-6}$; Figure 6a). Second, net counts hide flips: for Qwen2.5-Coder-7B the request repairs 17 histories and breaks 19 ($p = 0.87$), while the exact format removes every failure ($p = 1.2 \times 10^{-7}$; Figure 6a,b). Third, a parsed query can still be wrong: on the same Gemma-3-12B histories the exact format loses 11 of 110 correct first answers against 4 for the request ($p = 0.065$; Figure 6c).

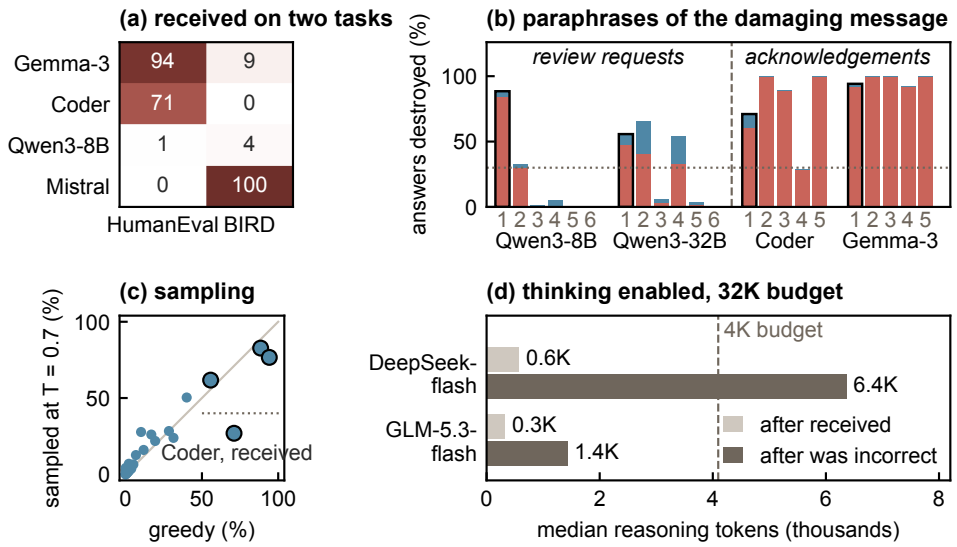

Figure 5: Where the damage holds. (a) Correct accepted answers destroyed by *received* on HumanEval and BIRD (%; counts in Tables 5 and 9): the ranking of models changes. (b) Paraphrases of each damaging message, greedy, numbered as in Table 3, where 1 (outlined) is the original, and split as in Figure 3b; dotted, the registered 30%. (c) Loss with the continuation sampled at $T = 0.7$ against greedy, one dot per model and message; grey line, equal loss; ringed, the four greedy losses above 50%; dotted, the registered 40%. (d) Median reasoning with thinking enabled at a 32K-token budget; dashed, the 4K budget under which DeepSeek-flash loses 117 of 158 answers after *was incorrect*.

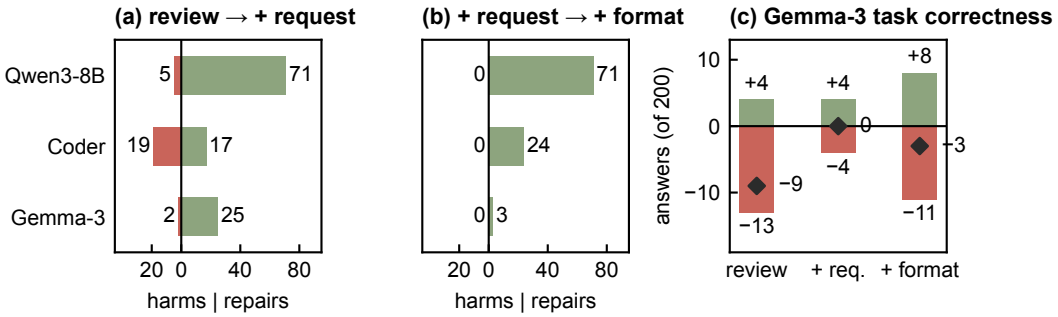

Figure 6: Net counts hide flips, and a parsed query can still be wrong (BIRD, 200 stored accepted histories per model). (a) Review to + request, (b) + request to + exact format: paired parser outcomes when a sentence is added: repairs turn a failure into a parsed reply, harms the reverse. (c) Gemma-3-12B's answers turned right (+) and wrong (−), with the net change (diamond); descriptive.

5.2 Conditions That Weaken the Damage

Wording and sampling. The acknowledgement failure extends to most other acknowledgements, while the review failure depends on the wording (Figure 5b). Only one of five other review requests destroys at least 30% of Qwen3-8B's answers, not the four we predicted, so for this model the damage follows the phrasing rather than the kind of request. Yet in an unregistered second stage, two prompts adapted from Huang et al. (2024) destroy 66% and 54% of Qwen3-32B's answers. At temperature 0.7, three of the four losses above 50% under greedy decoding are still 62–83%, but Qwen2.5-Coder-7B's *received* loss falls from 71% to 27%, below our registered 40% (Figure 5c).

Scale and model generation. Llama-3.1-70B loses 19.2% under *was incorrect*, like Llama-3.1-8B, while Qwen3-32B's worst loss is 55.7%, against 88.5% for Qwen3-8B. Against our prediction,

Table 2: One round of intrinsic self-correction on 200 GSM8K problems with the prompts of Huang et al. (2024), greedy. R^- deletes the final prompt's request to restate the answer; R_{nobox} keeps the request but deletes the box; R is their final prompt in full. Lost unboxed: correct first answers lost without a boxed answer. The indented row scores Llama-2-70b-chat's replies by their last number instead (Table 8). †Added after registration; –, not run.

	Accuracy (%)				Lost unboxed (count)		
Model	First	R^-	R_{nobox}	R	R^-	R_{nobox}	R
Llama-2-70b-chat	57.5	22.0	43.5	45.5	57	2	0
last number	56.0	41.0	43.5	44.5			
Llama-3.1-70B	95.0	85.0	–	87.5	5	–	3
Llama-3.1-8B	88.5	49.0	51.0	51.5	43	32	34
Mistral-7B	62.5	53.5	56.5	57.5	9	0	0
Gemma-3-12B	92.0	85.0	–	85.5	2	–	0
Gemma-4-12B†	96.0	95.0	–	96.0	0	–	0
Qwen3-8B	94.0	94.5	–	94.5	0	–	0
Qwen3-32B	93.5	92.5	–	93.5	0	–	0
GLM-5.3-flash (API)	96.5	96.5	–	96.5	0	–	0
DeepSeek-flash (API)	97.0	97.0	–	96.5	0	–	0

Gemma-4-12B loses at most 1.3% under any message, but its inference stack and GPU differ (Appendix E).

Reasoning and budgets. With thinking enabled, a false verdict multiplies the median reasoning of both API models (Figure 5d). Under it, DeepSeek-flash loses 117 of 158 correct answers at a budget of 4,096 tokens, almost all cut off before an answer, but only 12 of 163 at 32,768 tokens. The open model gpt-oss-20b shows the same effect (median 13,281 against 1,017 reasoning characters). Even at 32,768 tokens, 39 of its 161 continuations after *review* end without an answer, as do 39 after *review* + *format*. Overall, the loss and its repair by the exact format occur beyond HumanEval, but neither the damaging message nor its size is fixed across models or settings.

5.3 No Validator: Intrinsic Self-Correction

Intrinsic self-correction asks the model to revise every answer without outside feedback, so the same missing output lowers measured accuracy. Huang et al. (2024) showed that one round of it lowers accuracy on GSM8K (Cobbe et al., 2021), from 62.0% to 43.5% for Llama-2-70b-chat. Their protocol asks the question, then asks the model to review its answer, and finally, in a prompt we call R, asks it to improve the answer and restate it as a boxed number. We reran their prompts verbatim and greedily on 200 GSM8K problems, with their model and nine of ours, one of which (Gemma-4-12B) we added after registration. We also test R^-, which deletes R's last sentence and changes nothing else, and R_{nobox}, which keeps the request but deletes the box. We registered four predictions before the first call (Appendix E), and Table 2 reports every model.

With the format restated, the losses are semantic. Under the published protocol, Llama-2-70b-chat drops 12 points on our sample, the direction that Huang et al. (2024) report on theirs. All 34 answers it loses still end in a boxed number, which is now wrong. Every other model but Llama-3.1-8B loses at most three answers without a box. Llama-3.1-8B fails in a third way: 31 of its 34 unboxed losses run into the token limit, most of them repeating text. The published protocol therefore measures the semantic channel, plus non-termination in one model.

The request to restate, not the box, prevents format losses. Deleting R's last sentence widens Llama-2-70b-chat's accuracy drop from 12 to 36 points under the protocol's boxed-answer scorer ($p < 10^{-10}$). The change comes from missing boxes: of the 81 correct first answers it loses, 57 end without a box, and 35 of those still end with the right number. Scored by the last number instead, R and R^- cost 11.5 and 15 points, and the paired test no longer detects a difference ($p = 0.25$). The registered R_{nobox} arm leaves just two answers unboxed and scores 87 against R's 91 correct answers ($p = 0.39$; Table 2, Figure 4). For Llama-2-70b-chat and Mistral-7B, the request to restate the answer,

not the box, is what matters. For every other model R^- adds at most nine unboxed losses, refuting our registered prediction of at least ten for half of the models.

6 Related Work

Closest extra-turn measurements. Dip et al. (2026) observe extra-turn outcomes on correct and wrong first attempts, including correct answers turned wrong and turns with no final answer, to train a selective verification gate. Liu et al. (2024) test neutral prompts for intrinsic self-correction, and Yang et al. (2024) separate keeping a correct answer from repairing a wrong one. These precedents establish that a second turn can harm a correct answer. We instead ask what a fixed gate's logs cannot show about accepted answers, and which sentence makes the next reply deliver the answer.

Self-correction and retry. Reflexion (Shinn et al., 2023) retries after a reported failure and Self-Refine (Madaan et al., 2023) revises every answer. Intrinsic self-correction lowers reasoning accuracy (Huang et al., 2024; Kamoi et al., 2024), and self-debugging gains (Chen et al., 2024) shrink once the cost of repair is counted (Olausson et al., 2024; Bai et al., 2026). These studies score answers after a turn, but a gate never gives that turn to an accepted answer.

Harness effects. An agent's score measures a model and its harness together, and changing the harness can reverse rankings (Yao et al., 2026; Zhang et al., 2026), mask tool calls (Tang & Zheng, 2026), or inflate accuracy by retrying failed API requests (Dekoninck et al., 2026). Studies that vary what a repair turn contains (Ray & Goyal, 2026; Gumaan, 2026) start from a failure, whereas we vary one sentence after answers the harness accepted and find missing rather than changed answers.

Challenges, prompts and budgets. User pushback can flip correct answers (Laban et al., 2023; Sharma et al., 2024), whereas our false verdict keeps two models from omitting them. Negating an answer can also lower reasoning accuracy (Zhu et al., 2025), and a token budget can cut off reasoning that does not converge (Oladri et al., 2026), as in our thinking runs. Prompt formatting moves few-shot accuracy by up to 76 points (Sclar et al., 2024), and underspecified multi-turn instructions cut performance by 39% (Laban et al., 2026), whereas our tasks are fully specified and already solved. Cost-matched test-time studies compare budgets (Kassianik et al., 2026; Snell et al., 2024), whereas here the validator decides which answers get the budget, and shadow turns measure those that do not.

7 Conclusion and Limitations

A retry score evaluates the validator, continuation message, model and output format as one policy. The validator decides which answers receive another turn, but the gated score cannot reveal what that turn would do to accepted answers. Their largest losses are missing outputs. A request for the answer restored the output for some models, and the exact output format restored it in all six paired tests on HumanEval and SQL. Where losses remain, they change the answer or exhaust the token budget.

Retry results should therefore report the continuation message verbatim and, for a reasoning model, the token budget of any turn that carries a verdict. Gated logs reveal the retry's gain. Before changing the gate, reusing the message or crediting the gain to targeting, evaluators should probe the message on accepted answers. Continuations should ask for the answer again in the required format, and the same probe should check that request, since the harmful message changes across models.

Limitations. We test ten open-weight HumanEval models, Llama-2-70b-chat on GSM8K and two 'flash' API models, but no flagship closed model. These public, single-answer tasks may appear in training data, and long-horizon agents may differ. BIRD's SQL time limits can change which answers are accepted from run to run (Appendix E), and API endpoints can change. Five messages cannot cover the prompt space. Most results use one greedy reply per problem, and sampling cuts one of the four losses above 50% in Figure 5c to 27%. Thinking was tested for two API models and gpt-oss-20b, all greedily, and greedy reasoning can loop until the budget ends. GLM-5.3-flash reasons briefly even with thinking disabled. Shadow turns are single replies whose losses become real only when the gate changes, and agents that submit several times, like Cybench's, need an audit of their own loop. The bounds assume only bounded reward, but three models lose little, so the large effects establish neither a universal magnitude nor a model-independent prompt mechanism.

REPRODUCIBILITY STATEMENT

Rerunning. Every open-weight experiment runs with frozen decoding (bfloat16, one sequence at a time, eager execution, no prefix caching, greedy unless stated), except gpt-oss-20b and Phi-4, which were served with batched greedy decoding as declared before their first call, and Gemma-4-12B, which ran with Hugging Face Transformers on a different GPU (Appendix E). Every continuation message and prompt is given verbatim in Appendix A. API models were called through their official endpoints with thinking explicitly disabled or enabled, and the GSM8K sample is drawn with a fixed seed from the official test file.

Checks. Repeated HumanEval runs regenerate first attempts and compare extracted-code digests and pass/fail outcomes; these agree on all 164 problems for the six original models. Continuations rerun on the stored full text of each first exchange repeat the earlier replies byte for byte, for 421 of 421 replies in the submission-request rerun (E24) and 282 of 282 in the false-verdict rerun (E31, Table 13). HumanEval solutions run in a sandbox that passes all 164 canonical solutions and rejects all 20 negative controls. Every experiment after the first was registered before its first model call, apart from two additions that we mark as unregistered. The protocols record their amendments, and Appendix E lists two changes that we recorded only after the calls. The identification results are checked by an enumeration script, every headline contrast by an exact-test script whose functions reproduce textbook values, and every figure is generated from an aggregate file of counts computed from the per-problem records.

Materials. We will release with the final version the harness, runners, analysis and figure scripts, frozen configurations, and the aggregate outcome counts behind every headline claim, together with complete HumanEval examples chosen by a fixed hash of the problem identifier rather than by outcome. Raw generations and per-problem records are retained.

AI USE STATEMENT

We used LLMs in several parts of this work. For the manuscript, the authors supplied the draft, and LLMs polished the text and suggested its structure. For the experiments, the authors designed the experiments and LLMs wrote the code; the authors then ran the experiments and analyzed the results. LLMs also helped search and verify the literature. The authors take full responsibility for the paper.

REFERENCES

Yu Bai, Yukai Miao, Dawei Wang, Li Chen, Yanyu Ren, Yuqian Shi, Dan Li, Ying Xiong, Chengqiu Tan, Run Zhou, and Li Li. VRL-Bench: Benchmarking agents on computer control tasks under finite trial budgets. *arXiv preprint arXiv:2609.12404*, 2026.

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

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jasper Dekoninck, Nikola Jovanović, Kári Rögnvaldsson, and Martin Vechev. The hidden effect of retrying requests in LLM evaluation. MathArena technical report, 2026. URL `https://matharena.ai/retry/`.

Sajib Acharjee Dip, Dawei Zhou, and Liqing Zhang. Think again or think longer? selective verification for budget-aware reasoning. *arXiv preprint arXiv:2606.19808*, 2026.

Guardrails AI contributors. Validation reask control flow. Guardrails AI source code, commit 06d0ff2, 2026. URL https://github.com/guardrails-ai/guardrails/tree/06d0ff2c5f9bcb493d976b76f885e37e41ce845d/guardrails.

Esmail Gumaan. Feedback that backfires: Why small language model agents repeat the call they just watched fail. *arXiv preprint arXiv:2608.23651*, 2026. URL https://arxiv.org/abs/2608.23651.

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

Inspect Evals contributors. Inspect Evals: Cybench and TheAgentCompany task implementations. Source code, commit 78fd6b0, 2026. URL https://github.com/UKGovernmentBEIS/inspect_evals/tree/78fd6b018ac945346332015e679cb6e61c7939b1/src/inspect_evals.

Instructor contributors. Validation retry control flow. Instructor source code, commit e12f8b4, 2026. URL https://github.com/567-labs/instructor/tree/e12f8b49203b0c1f253d27c1e709d0a09b9fc5a8/instructor/v2.

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.

Paul Kassianik, Blaine Nelson, and Yaron Singer. Beyond success rate: Cost-aware evaluation of offensive and defensive security agents. *arXiv preprint arXiv:2607.15263*, 2026.

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.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations*, 2026.

LangChain contributors. Structured output: Error handling and retry. LangChain documentation, commit 5b2f471, 2026. URL https://github.com/langchain-ai/docs/blob/5b2f471fbbee26405fd349770f3b3146833e828a/src/oss/langchain/structured-output.mdx.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as a database interface? a BIg bench for large-scale database grounded text-to-SQLs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/83fc8fab1710363050bbd1d4b8cc0021-Abstract-Datasets_and_Benchmarks.html.

Dancheng Liu, Amir Nassereldine, Ziming Yang, Chenhui Xu, Yuting Hu, Jiajie Li, Utkarsh Kumar, Changjae Lee, Ruiyang Qin, Yiyu Shi, and Jinjun Xiong. Large language models have intrinsic self-correction ability. *arXiv preprint arXiv:2406.15673*, 2024.

Genglin Liu and Saadia Gabriel. PM-bench: Evaluating prospective memory in LLM agents. In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2607.12385.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594, 2023.

Charles F. Manski. *Partial Identification of Probability Distributions*. Springer, 2003.

Renuka Oladri, Niveda Jawahar, and Abdirisak Mohamed. Token budget saturation and mechanistic early detection of reasoning non-convergence in chain-of-thought models. *arXiv preprint arXiv:2607.21433*, 2026.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *International Conference on Learning Representations*, 2024.

PydanticAI contributors. Retries: Output and tool retry semantics. PydanticAI documentation, commit 68252f1, 2026. URL `https://github.com/pydantic/pydantic-ai/blob/68252f1904d0b73ddd859db20b5f5a2252b5c0ca/docs/retries.md`.

Jaideep Ray and Ankit Goyal. Structured feedback improves repair in an LLM agent loop. *arXiv preprint arXiv:2607.14167*, 2026.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Lijuan Tang and Yuemeng Zheng. Measuring the serving stack instead of the model: Hidden confounds in local tool-use evaluation. *arXiv preprint arXiv:2609.26693*, 2026.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. Confidence v.s. critique: A decomposition of self-correction capability for llms. *arXiv preprint arXiv:2412.19513*, 2024.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. HarnessBench: Measuring harness effects across models in realistic agent workflows. *arXiv preprint arXiv:2605.27922*, 2026.

Andy K. Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W. Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Jasper, Pura Peetathawatchai, Ari Glenn, Vikram Sivashankar, Daniel Zamoshchin, Leo Glikbarg, Derek Askaryar, Mike Yang, Teddy Zhang, Rishi Alluri, Nathan Tran, Rinnara Sangpisit, Polycarpos Yiorkadjis, Kenny Osele, Gautham Raghupathi, Dan Boneh, Daniel E. Ho, and Percy Liang. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *International Conference on Learning Representations*, 2025.

Yunbei Zhang, Janet Wang, Yingqiang Ge, Weijie Xu, Jihun Hamm, and Chandan K. Reddy. Stop comparing LLM agents without disclosing the harness. *arXiv preprint arXiv:2605.23950*, 2026.

Jinqing Zhao and Chengcan Wu. Making prospective memory SLM-shaped: Typed intention stores for small-model agents. *arXiv preprint arXiv:2609.01272*, 2026.

Bin Zhu, Hailong Yin, Jingjing Chen, and Yu-Gang Jiang. Benchmarking gaslighting negation attacks against reasoning models. *arXiv preprint arXiv:2506.09677*, 2025.

APPENDIX OVERVIEW

Appendix	What it lets a reader check
A Messages	every continuation message and prompt, verbatim, and the sentences each follow-up arm adds
B Identified set	proofs of Propositions 1 and 2 and the numerical check
C Models and checks	models, decoding, the sandbox, and the replay and extraction checks
D Full results	every model under every message, exact tests, BIRD, sampling, paraphrases, thinking, self-correction and reply content
E Predictions	every experiment in order (Table 13), each registered prediction and its outcome, amendments, unregistered additions, and the BIRD follow-ups (Figure 6)
F Allocation	the registered allocation design from which this study grew

A CONTINUATION MESSAGES, VERBATIM

Every string below is copied from the runner that sent it. *was incorrect* is the retry message of Cybench's Inspect implementation (Inspect Evals contributors, 2026) unchanged; *received* is the same sentence with its verdict removed. For comparison, the retry message of the Inspect implementation of TheAgentCompany (Xu et al., 2024; Inspect Evals contributors, 2026), which we did not send, states no verdict: "Please continue working on the task. If you think you have solved the task, submit your final answer by calling the submit() tool. IMPORTANT: YOU SHOULD NEVER ASK FOR HUMAN HELP. If you are stuck and cannot make progress, call the submit() tool to give up."

HumanEval system message. "Complete the Python function. Return the complete implementation, including the signature, as one ```python``` fenced block and no prose."

BIRD system message. "Solve the BIRD Mini-Dev question with SQLite. Return exactly one non-empty ```sql``` fenced block containing one read-only SELECT or WITH query. Return no prose before or after the fence."

API models. GLM-5.3-flash and DeepSeek-flash receive the HumanEval system message and the five continuation messages above, unchanged, as a system turn and user turns.

GSM8K self-correction prompts (Section 5.3). Copied from Huang et al. (2024), Appendix A, and sent with no system message. Question: "Can you solve the following math problem? {question} Explain your reasoning. Your final answer should be a single numerical number, in the form \boxed{answer}, at the end of your response." Feedback: "Assume that this answer could be either correct or incorrect. Review the answer carefully and report any serious problems you find." Refine (R): "Based on the problems you found, improve your answer. Please reiterate your answer, with your final answer a single numerical number, in the form \boxed{answer}." Refine without its last sentence (R^-, ours): "Based on the problems you found, improve your answer." Refine without the box (R_{nobox}, ours, registered after the others): "Based on the problems you found, improve your answer. Please reiterate your answer, with your final answer a single numerical number." All refine prompts continue the same history, so the arms differ in their last sentence alone.

BIRD versions of the messages. *received* and *was incorrect* are unchanged. *received* + *format*: "Your submission has been received. Please proceed and attempt to find the correct answer. Return exactly one non-empty ```sql``` fenced SELECT/WITH query." *review*: "Review the preceding SQL for possible mistakes in joins, filters, aggregation, units, and syntax. Then state your final answer. You receive no validator result." *review* + *format*: "Review the preceding SQL for possible mistakes in joins, filters, aggregation, units, and syntax. Return a revised final answer as exactly one non-empty ```sql``` fenced SELECT/WITH query. You receive no validator result."

E22–E24 submission-cue follow-ups. E22's HumanEval *A_generic* and *R_generic* append exactly "Submit your final answer now." to the *received* and *review* stems in Figure 7. Their *specific* arms append one more sentence: "Return your answer as one complete Python function in a ```python```

System

Complete the Python function. Return the complete implementation, including the signature, as one ```python``` fenced block and no prose.

User: the HumanEval problem (HumanEval/143)

```
def words_in_sentence(sentence):
    """Return the words whose lengths are prime, in their order. ..."""
```

Assistant: first answer, which passes the tests (accepted)

```
```python
def words_in_sentence(sentence):
    """..."""                # docstring as in the prompt
    def is_prime(n):
        if n <= 1:
            return False
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
                return False
        return True
    words = sentence.split()
    result = []
    for word in words:
        if is_prime(len(word)):
            result.append(word)
    return " ".join(result)
```
```

User: one of five continuation messages (the shadow turn)

received	Your submission has been received. Please proceed and attempt to find the correct answer.
received + *format*	Your submission has been received. Please proceed and attempt to find the correct answer. Return your answer as one complete Python function in a ```python``` block.
review	Review the preceding implementation for possible mistakes in edge cases, types, indexing, and control flow. Then state your final answer. You receive no validator result.
review + *format*	Review the preceding implementation for possible mistakes in edge cases, types, indexing, and control flow. Return a revised final answer as one complete Python function in a ```python``` block. You receive no validator result.
was incorrect	Your submission was incorrect. Please proceed and attempt to find the correct answer.

Assistant after *received* (Gemma-3-12B): no code, so the answer is lost

Okay, I understand. I'm ready for the next task or question. Just let me know what you'd like me to do.

Assistant after *was incorrect* (Gemma-3-12B): the same function again, which passes the tests

```
```python
def words_in_sentence(sentence):
    """..."""
    def is_prime(n):
        ...
    return " ".join(result)
```
```

Figure 7: A shadow turn as a conversation, with the five continuation messages verbatim (HumanEval/143, Gemma-3-12B, greedy; long docstrings and repeated code elided). The reply is scored in place of the first answer. The problem was chosen by a fixed hash of its identifier, not by outcome. The BIRD messages differ only where the output is named, as listed below.

block." E23's BIRD *R_generic* uses the same submit suffix on the BIRD *review* stem, while *R_specific* adds: "Return exactly one non-empty ```sql``` fenced SELECT/WITH query." E24 reuses E22's exact full first histories and compares the unsuffixed HumanEval stems with their *generic* versions; it does not change either message. In the figures, *generic* is "+ request for the answer" and *specific* is "+ exact output format".

Paraphrase families (Section 5). Review requests R1–R6 were sent to Qwen3-8B and acknowledgements A1–A5 to Qwen2.5-Coder-7B, greedily, once per accepted problem; an unregistered second stage sent them to Qwen3-32B and Gemma-3-12B. Huang et al. (2024) use their prompts as the middle step of a three-step protocol whose last step asks for the answer again, so R2–R4 append the clause "Then give your final answer." to make one scored continuation.

Table 3: Paraphrase families, copied from the runner that sent them.

ID	Message	Source
R1	Review the preceding implementation for possible mistakes in edge cases, types, indexing, and control flow. Then state your final answer. You receive no validator result.	the original (*review*)
R2	Assume that this answer could be either correct or incorrect. Review the answer carefully and report any serious problems you find. Then give your final answer.	Huang et al. (2024), verbatim, + clause
R3	Review your previous answer and determine whether it's correct. If wrong, find the problems with your answer. Then give your final answer.	Huang et al. (2024), verbatim, + clause
R4	Verify whether your answer is correct, and provide an explanation. Then give your final answer.	Huang et al. (2024), verbatim, + clause
R5	Are you sure?	Laban et al. (2023), verbatim
R6	Please double-check your solution for errors, then give your final answer.	plain paraphrase
A1	Your submission has been received. Please proceed and attempt to find the correct answer.	the original (*received*)
A2	Thank you. Your answer has been recorded. Please continue.	ours
A3	Noted. Please proceed.	ours
A4	Your response has been saved. You may continue working on the problem.	ours
A5	Submission received. Continue.	ours

B THE IDENTIFIED SET, PROVED AND CHECKED

Setup. Binary trigger T, first reward $z_1 \in [0,1]$, continuation reward $z_2 \in [0,1]$, $D = z_2 - z_1$, $p = \Pr(T = 1)$. We grant a rich release of p, $\mathbb{E}[z_1 \mid T = 0]$, $\mathbb{E}[z_1 \mid T = 1]$ and, through $V_{\text{gated}} = V_1 + p\,\mathbb{E}[D \mid T = 1]$, the retry's effect $\mathbb{E}[D \mid T = 1]$. It does not reveal $m_0 = \mathbb{E}[z_2 \mid T = 0]$.

Proof of Proposition 1, and the two regimes. The gated score is $V_1 + p\,\mathbb{E}[D \mid T = 1]$ and the ungated score is $V_1 + \mathbb{E}[D]$, so $G = -(1-p)\,\mathbb{E}[D \mid T = 0] = (1-p)\big(\mathbb{E}[z_1 \mid T = 0] - m_0\big)$, and letting m_0 range over $[0,1]$ gives the stated interval, whose ends $m_0 = 1$ and $m_0 = 0$ are attainable. In the oracle regime rewards are binary and $T = 1$ exactly when $z_1 = 0$, so $\mathbb{E}[z_1 \mid T = 0] = 1$, $1 - p = V_1$, and $G \in [0, V_1]$. In the proxy regime the validator may accept wrong answers. If $p < 1$ and $0 < \mathbb{E}[z_1 \mid T = 0] < 1$, the interval for G contains both signs; if every accepted answer is wrong, it is $[-(1-p), 0]$. □

Proof of Proposition 2. For binary T, $\mathrm{Cov}(T, D) = \mathbb{E}[TD] - p\,\mathbb{E}[D] = p\,\mathbb{E}[D \mid T = 1] - p\big(p\,\mathbb{E}[D \mid T = 1] + (1-p)\,\mathbb{E}[D \mid T = 0]\big) = p(1-p)\big(\mathbb{E}[D \mid T = 1] - \mathbb{E}[D \mid T = 0]\big)$. Substituting $\mathbb{E}[D \mid T = 0] = m_0 - \mathbb{E}[z_1 \mid T = 0]$ gives $\mathrm{Cov}(T, D) = p(1-p)(a - m_0)$ with $a = \mathbb{E}[D \mid T = 1] + \mathbb{E}[z_1 \mid T = 0]$. The map is affine and strictly decreasing in m_0 whenever $0 < p < 1$. With only the bounded-reward restriction, any $m_0 \in [0,1]$ is consistent with the release. Its image is exactly $p(1-p)[a-1, a]$, both endpoints are attainable, the width

is $p(1-p)$, and zero lies inside exactly when $0 \le a \le 1$. For a fixed census of n_0 accepted tasks with binary rewards, the attainable m_0 values instead form $\{0, 1/n_0, \ldots, 1\}$; the displayed interval is their sharp convex hull, with the same attainable endpoints. In the oracle regime the set is $\mathrm{Cov}(T, D) \in p(1-p)[\mathbb{E}[D \mid T=1],\ \mathbb{E}[D \mid T=1]+1]$; since $z_1 = 0$ on the rejected stratum, $\mathbb{E}[D \mid T=1] = \mathbb{E}[z_2 \mid T=1] \ge 0$, and zero is inside only if the retry never succeeds. □

The set a weaker log would give. If a log recorded first attempts and triggers but not the retry's outcome, $\mathbb{E}[z_2 \mid T=1]$ would also be free and the set would widen to $p(1-p)[-1-\mu_1+\mu_0,\ 1-\mu_1+\mu_0]$ with $\mu_t = \mathbb{E}[z_1 \mid T=t]$, of width $2p(1-p)$, which contains zero for every population. This weaker-log case still grants first attempts and triggers. A score-only release is no more informative; the propositions grant V_1, V_{gated}, T, and z_1 to make the impossibility conservative.

Numerical check. We enumerated every non-degenerate oracle-regime population of four to six tasks with binary rewards and every binary assignment of z_2 on each, and drew 600 random populations of 6 to 40 tasks in both regimes with 400 random completions each. In all 1,538 populations every attained covariance lay in the interval, both endpoints were attained, and the width equalled $p(1-p)$.

C Models, Decoding, and Replay and Extraction Checks

Models. Table 4 lists the eleven open checkpoints. Each was pinned by the SHA-256 of every file at download. Qwen chat templates run with thinking disabled. Gemma's template has no system role, so the system message is prepended to the first user turn, where it remains in context for every continuation. The Llama-2-70b-chat mirror carries no chat template, so we added the official Llama-2-chat template before its first call (Appendix E). The two API models are `glm-5.3-flash` and `deepseek-flash`, called through their providers' official endpoints; each response names the model that served it.

Table 4: Checkpoints. Repository identifiers as downloaded; every file pinned by SHA-256. Gemma-4-12B-it came from its ModelScope mirror, and gpt-oss-20b is a BF16 copy of the released weights.

Model	Family	Repository
Qwen3-8B	Qwen	`Qwen/Qwen3-8B`
Qwen3-32B	Qwen	`Qwen/Qwen3-32B`
Qwen2.5-Coder-7B-Instruct	Qwen	`Qwen/Qwen2.5-Coder-7B-Instruct`
Gemma-3-12B-it	Gemma	`LLM-Research/gemma-3-12b-it`
Mistral-7B-Instruct-v0.3	Mistral	`mistralai/Mistral-7B-Instruct-v0.3`
Llama-3.1-8B-Instruct	Llama	`LLM-Research/Meta-Llama-3.1-8B-Instruct`
Llama-3.1-70B-Instruct	Llama	`LLM-Research/Meta-Llama-3.1-70B-Instruct`
Llama-2-70b-chat	Llama	`modelscope/Llama-2-70b-chat-ms`
Gemma-4-12B-it	Gemma	`google/gemma-4-12B-it`
gpt-oss-20b	OpenAI	`unsloth/gpt-oss-20b-BF16`
Phi-4	Phi (Microsoft)	`LLM-Research/phi-4`

Decoding. vLLM 0.23.0, PyTorch 2.11, Transformers 5.5.1, bfloat16, one sequence at a time, eager execution, no prefix caching, context limit 32,768 tokens, at most 1,024 new tokens. Greedy decoding is temperature 0, top-p 1 and seed 0. Section 5 adds temperature 0.7 with declared seeds. The 70B models run with tensor parallelism across two NVLinked H200 GPUs, Llama-2-70b-chat with its native context of 4,096 tokens. Gemma-4-12B-it runs under Hugging Face Transformers with greedy decoding on one RTX 5090, with a context of 8,192 tokens (Appendix E); every other open model runs on one H200. gpt-oss-20b and Phi-4 run under vLLM's own OpenAI-compatible server, which separates gpt-oss's reasoning from its final reply, with CUDA graphs and up to eight concurrent sequences, greedily; gpt-oss-20b reasons at medium effort with 32,768 new tokens and Phi-4 has 1,024. The API models are called with temperature 0 and thinking disabled unless stated; with thinking enabled, the budget is 4,096 or 32,768 new tokens. With thinking disabled, DeepSeek-flash returns no reasoning tokens, but GLM-5.3-flash still returns a short reasoning trace on most continuations, with a median of 3 tokens after *received* and 77 after *was incorrect*. GSM8K turns allow at most 1,024 new tokens and send no system message.

Turns. The first request is the system message and the task. Each continuation resends that history with the model's own first answer as the assistant turn, followed by one continuation message as a user turn. No continuation ever enters a reported score.

Scoring. HumanEval code is extracted from the last fenced block, or the whole reply if there is none, and run against the problem's own `check()` in a sandbox with no network access, POSIX limits on memory, CPU time, file size and processes, and destructive library functions removed. The sandbox passes all 164 canonical solutions and rejects all 20 negative controls. BIRD uses the official post-processing and set-equality evaluator. Its 997 tasks are a census, not a sample: the 1,037 questions of the BIRD development split that are not in Mini-Dev, minus 40 used while building the harness (`configs/bird_budget_matched_v1.json`); the system message's mention of Mini-Dev is verbatim from that harness.

Replay, extraction and context checks. Three checks rule out the implementation explanations we could test. First, repeated runs regenerate first attempts and match stored extracted-code digests and pass/fail outcomes for all 164 problems in the six original models. A fenced-block digest covers the extracted function, not any surrounding text, so this does not certify full-text identity. The contrasts we read causally were therefore rerun on stored complete first histories: E24 for the submission request and E31 for the false verdict, whose *received* replies repeat E24's text byte for byte on all 282 accepted answers. Llama-3.1-70B was run once, so this check does not apply to it or to the three other later models. Second, the 276 losses we regenerated, of the 285 that the first two messages we tested produced on five models, match stored extracted-code digests and parse kinds (the other nine are Llama-3.1-8B's and Qwen2.5-Coder-7B's losses under *review* + *format*). For unfenced replies the digest covers the full reply. Cutting each reply at `def <entry_point>` before running the tests, the most permissive extraction we could construct, recovers one loss. Third, the system message that asks for a fenced function is part of the context of every continuation, so a reply without code abandons a format that is in view.

D Every Model Under Every Message

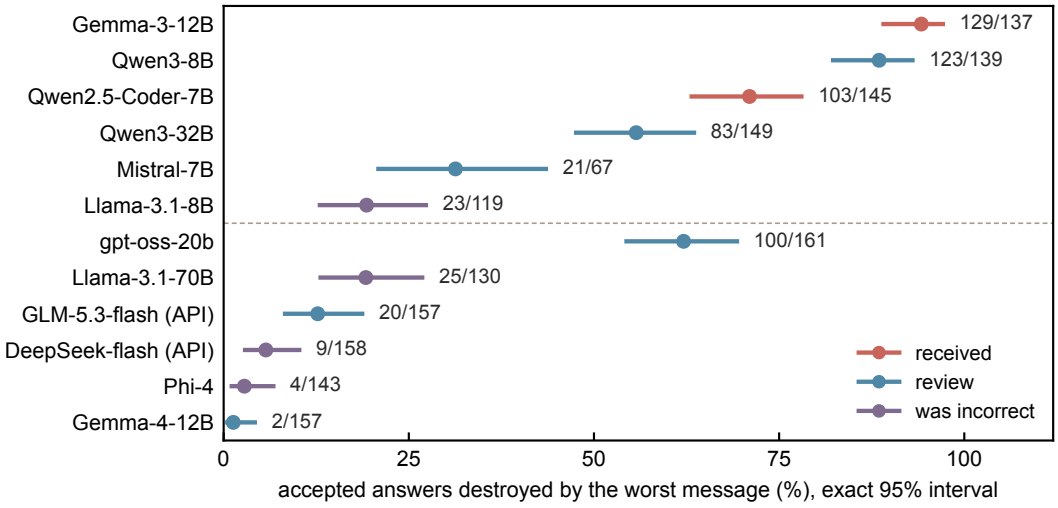

Figure 8: Worst of five messages per model on accepted HumanEval answers, greedy: a stress test, not a deployment rate. Bars are exact 95% intervals that treat the problems as a sample of similar tasks and do not adjust for message selection; labels give answers destroyed out of those accepted. Above the dashed line, the six original models; below it, later models with the same messages and scorer (Appendix E). Gemma-4-12B's *review* ties with *review* + *format*.

Delivery rules. Figure 9 rescores the same stored replies under a harness that keeps the first answer whenever the reply has no code block. Every accepted first answer is correct, so only replies that

carry failing code are then lost. The rule removes most omission losses but none of the semantic ones, and the false verdict becomes the worst message for eight of the twelve models. A gate that only enforces the format would route none of these answers: every accepted first reply of Gemma-3-12B, Qwen2.5-Coder-7B and Qwen3-8B is already exactly one fenced block (a post hoc, zero-call check; Appendix E).

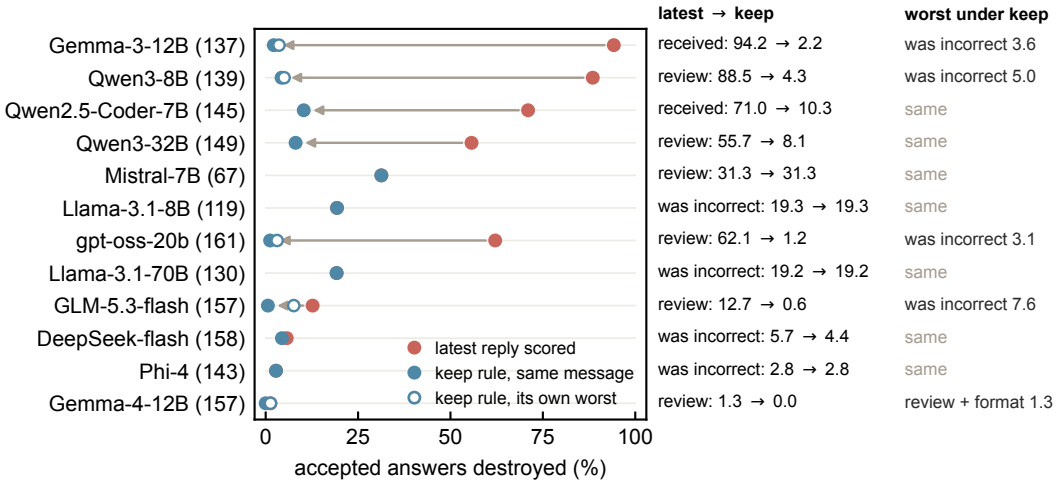

Figure 9: The delivery rule sets what the gate prevents (HumanEval, greedy; accepted answers in parentheses). Red: the worst message when the reply replaces the first answer, as in the main text. Blue: the same replies when the harness keeps the first answer whenever a reply has no code block; hollow: the worst message under that rule where it differs. Both rules score the same stored replies, with no new calls.

Table 5: Accepted-stratum losses on HumanEval. Each cell: correct answers destroyed (of which without code), then rate and exact 95% interval.

Model	n	received	received + format	review	review + format	was incorrect
Gemma-3-12B	137	129 (126) 94% [89, 97]	4 (0) 3% [1, 7]	5 (2) 4% [1, 8]	3 (0) 2% [0, 6]	5 (0) 4% [1, 8]
Qwen3-8B	139	1 (0) 1% [0, 4]	2 (0) 1% [0, 5]	123 (117) 88% [82, 93]	1 (0) 1% [0, 4]	7 (0) 5% [2, 10]
Qwen2.5-Coder-7B	145	103 (88) 71% [63, 78]	0 (0) 0% [0, 3]	58 (50) 40% [32, 48]	1 (0) 1% [0, 4]	3 (0) 2% [0, 6]
Qwen3-32B	149	4 (0) 3% [1, 7]	5 (0) 3% [1, 8]	83 (71) 56% [47, 64]	2 (0) 1% [0, 5]	10 (0) 7% [3, 12]
Mistral-7B	67	0 (0) 0% [0, 5]	0 (0) 0% [0, 5]	21 (0) 31% [21, 44]	19 (0) 28% [18, 41]	0 (0) 0% [0, 5]
Llama-3.1-8B	119	20 (2) 17% [11, 25]	12 (1) 10% [5, 17]	14 (6) 12% [7, 19]	8 (0) 7% [3, 13]	23 (0) 19% [13, 28]

Effect on the measured value of feedback. On the rejected problems, where a retry study measures the value of a verdict, all six of Gemma-3-12B's rescues by the verdict follow a neutral reply without code, whereas five of Llama-3.1-8B's six follow a neutral reply that did contain code (Table 7). These strata are small, but part of what such studies record as the value of feedback may be the value of a message that keeps the model submitting.

E Registered Predictions and Outcomes

Every experiment after the first HumanEval run was registered before its first model call, with the outcome that would refute it written into the protocol, apart from the two unregistered additions listed

Table 6: Exact McNemar tests on the same accepted problems. Discordant: lost only under the first message / only under the second. Holm correction within each block.

Contrast	Model	Lost (1st)	Lost (2nd)	Discordant	p	Holm
received → received + format	Gemma-3-12B	129	4	127 / 2	2.5e-35	1.5e-34
	Qwen3-8B	1	2	0 / 1	1.0e+00	1.0e+00
	Qwen2.5-Coder-7B	103	0	103 / 0	2.0e-31	9.9e-31
	Qwen3-32B	4	5	1 / 2	1.0e+00	1.0e+00
	Mistral-7B	0	0	0 / 0	1.0e+00	1.0e+00
	Llama-3.1-8B	20	12	14 / 6	1.2e-01	4.6e-01
received → was incorrect	Gemma-3-12B	129	5	126 / 2	4.9e-35	2.9e-34
	Qwen3-8B	1	7	0 / 6	3.1e-02	1.2e-01
	Qwen2.5-Coder-7B	103	3	100 / 0	1.6e-30	7.9e-30
	Qwen3-32B	4	10	0 / 6	3.1e-02	1.2e-01
	Mistral-7B	0	0	0 / 0	1.0e+00	1.0e+00
	Llama-3.1-8B	20	23	13 / 16	7.1e-01	1.0e+00

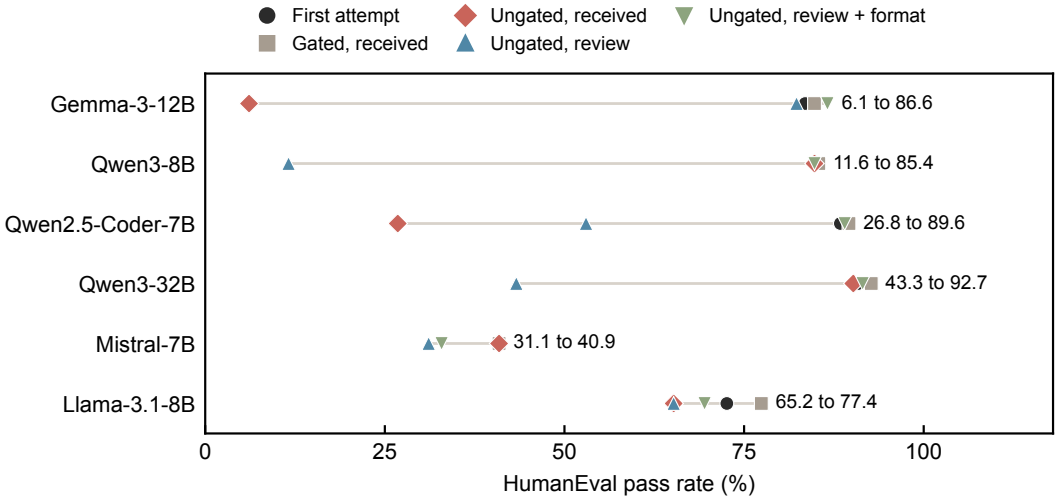

Figure 10: One ungated round sent to every answer, under three messages (HumanEval pass rate). The first attempt and a gated retry under *received* are shown for reference; labels give each model's range.

Table 7: The rejected stratum, where the verdict *was incorrect* is true. Replies without code under each message, and the verdict's rescues split by whether the neutral message had produced any code. Lost: problems that pass under *received* but fail under *was incorrect*.

Model	Rejected	No code, received	No code, was incorrect	Rescued, neutral had no code	Rescued, neutral had code	Lost
Gemma-3-12B	27	17	0	6	0	1
Qwen3-8B	25	0	0	0	0	0
Qwen2.5-Coder-7B	19	10	0	2	1	1
Qwen3-32B	15	0	0	0	0	0
Mistral-7B	97	0	0	0	1	0
Llama-3.1-8B	45	2	0	1	5	3

Table 8: The box or the request (GSM8K, 200 problems, greedy). R_{nobox} keeps R's request to restate the answer as a single number and deletes only the box. Correct answers under two extraction rules, the last boxed answer and the last number; the second is an alternative, not a more lenient rule (Mistral-7B's first answers score 125 and 112). Lost without a box: correct first answers lost without a boxed answer. The history of every problem is E18's, reused after a digest check.

		Correct (of 200)				Lost without a box		
Model	Scoring	First	R	R^{-}	R_{nobox}	R	R^{-}	R_{nobox}
Llama-2-70b-chat	boxed	115	91	44	87	0	57	2
	last number	112	89	82	87			
Mistral-7B	boxed	125	115	107	113	0	9	0
	last number	112	99	95	102			
Llama-3.1-8B	boxed	177	103	98	102	34	43	32
	last number	177	110	101	115			

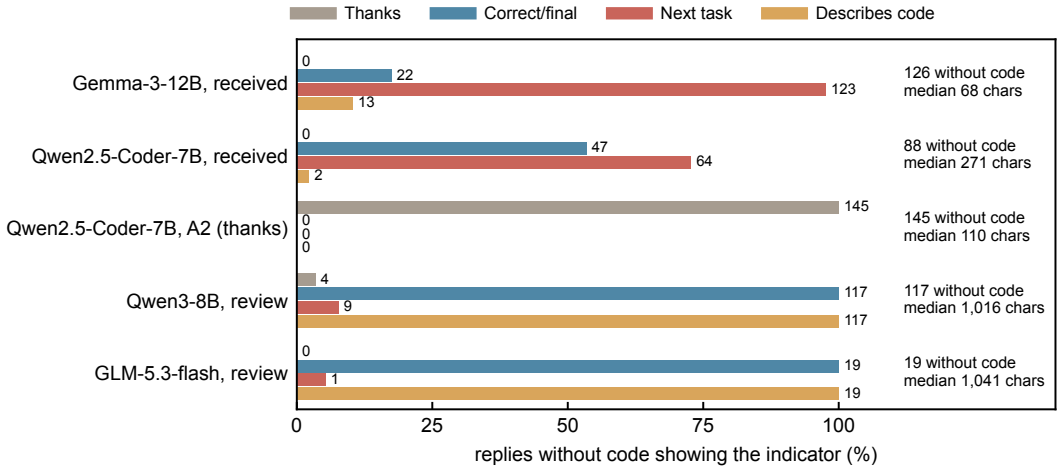

Figure 11: What replies without code say, by keyword indicators fixed before the run (bars: share of the replies without code; labels: counts; indicators may overlap). A registered descriptive run regenerated every loss of four cells greedily and kept a reply only if its digests matched the record, which all 500 did; GLM-5.3-flash's are the replies stored by the API experiment. The correct-or-final indicator also fires on "final answer", which the review request asks for.

Table 9: BIRD accepted stratum: correct answers destroyed / wrong answers rescued under each message. The validator accepts wrong answers, so both directions occur. Every row but Gemma-3-12B's comes from one grading run; Gemma-3-12B's review arm stopped at 308 tasks, so its row reports the two arms it completed.

Model	Accepted	Correct	received	received + format	review	review + format
Qwen3-8B	815	484	17 / 12	18 / 10	374 / 8	5 / 3
Mistral-7B	440	229	229 / 1	227 / 1	215 / 5	177 / 3
Qwen2.5-Coder-7B	767	472	0 / 1	2 / 1	63 / 1	6 / 3
Gemma-3-12B	820	489	45 / 37	–	–	47 / 43
Llama-3.1-8B	657	377	30 / 8	26 / 14	37 / 9	26 / 15

below. We report every registered prediction, including those that failed, and every change made after registration. Table 13 lists the experiments in order.

- **Four messages (Section 3).** Restating the format rescues the two models that drop code: held. The review request is what costs Mistral-7B: held. No message is safe for every model: held. A prediction about interaction between the two factors was withdrawn before the deciding data existed, once we noticed that the four messages do not share a stem and so

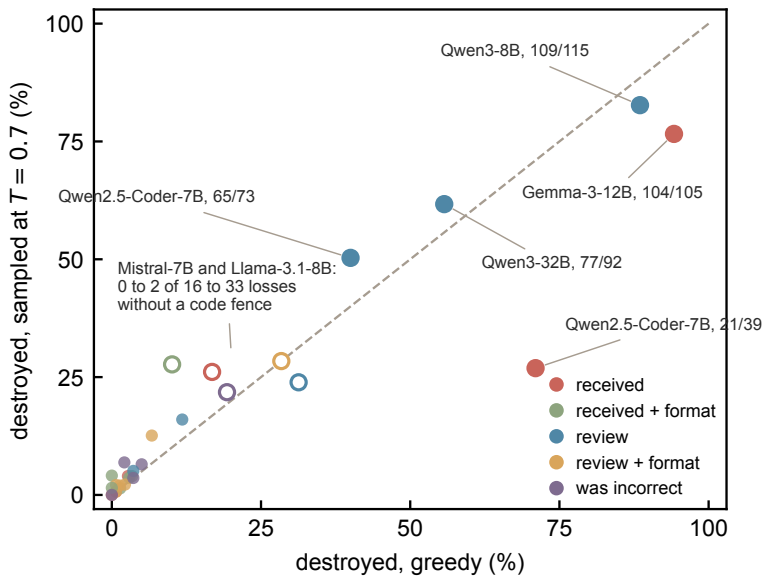

Figure 12: Sampled replication on the HumanEval accepted stratum: each dot is one model and message, destroyed with the continuation sampled at temperature 0.7 (seed 1; first attempts greedy) against the greedy value. Large dots are cells whose sampled loss is at least 20%, filled when most of those losses lack a code fence and hollow otherwise; labels give losses without a fence out of all losses. Qwen3-32B under *was incorrect* has no greedy value and loses 6.7% when sampled.

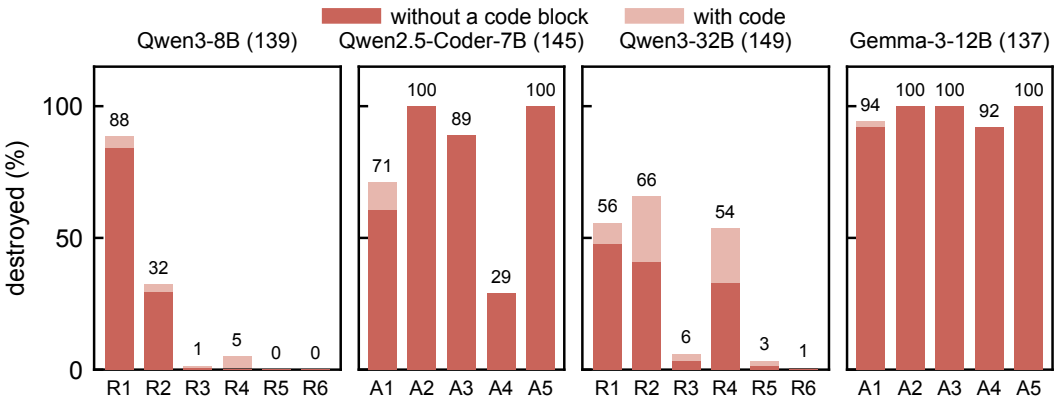

Figure 13: Paraphrase families, greedy (messages in Table 3; accepted answers in parentheses). Bars give correct answers destroyed, the dark part those whose reply has no code. Qwen3-8B and Qwen2.5-Coder-7B are the registered test; Qwen3-32B and Gemma-3-12B are the unregistered second stage. R1 and A1 reproduce the earlier replies on every accepted problem for all four models.

are not a factorial. The analysis script for this experiment was written after the first model's results were visible; we state this because the protocol said otherwise.

- **The false verdict (Section 4).** We predicted that a false verdict would destroy at least as many correct answers as the neutral message for every model, and that Mistral-7B would not ignore it. Both failed. The prediction that losses under the false verdict would less often lack code held.
- **BIRD (Section 5).** The prediction that could refute the two-channel decomposition, that restating the format would not rescue Mistral-7B's semantic losses, held. The registered wording said "without an SQL fence," but its prewritten code counted `parse_kind != official_fence`, which also includes invalid fenced SQL (we recorded this as a

Table 10: The later models on the HumanEval accepted stratum, greedy, thinking disabled for the API models. Below the rule, the two families added last (E30), served with batched decoding; gpt-oss-20b reasons at medium effort with a 32,768-token budget, and a reply cut off by the budget counts as without a code fence. Each cell: correct answers destroyed (of which without a code fence), then rate and exact 95% interval.

Model	Accepted	received	received + format	review	review + format	was incorrect
Llama-3.1-70B	130	17 (0) 13% [8, 20]	19 (0) 15% [9, 22]	13 (1) 10% [5, 16]	6 (0) 5% [2, 10]	25 (0) 19% [13, 27]
Gemma-4-12B	157	0 (0) 0% [0, 2]	1 (1) 1% [0, 3]	2 (2) 1% [0, 5]	2 (0) 1% [0, 5]	1 (1) 1% [0, 3]
GLM-5.3-flash	157	1 (0) 1% [0, 3]	0 (0) 0% [0, 2]	20 (19) 13% [8, 19]	4 (1) 3% [1, 6]	13 (1) 8% [4, 14]
DeepSeek-flash	158	1 (0) 1% [0, 3]	1 (0) 1% [0, 3]	1 (0) 1% [0, 3]	0 (0) 0% [0, 2]	9 (2) 6% [3, 11]
gpt-oss-20b	161	6 (4) 4% [1, 8]	4 (2) 2% [1, 6]	100 (98) 62% [54, 70]	40 (39) 25% [18, 32]	25 (20) 16% [10, 22]
Phi-4	143	1 (0) 1% [0, 4]	0 (0) 0% [0, 3]	1 (0) 1% [0, 4]	3 (0) 2% [0, 6]	4 (0) 3% [1, 7]

Table 11: The API models with thinking enabled, HumanEval accepted stratum. At 32,768 tokens, DeepSeek-flash ran all five messages on the first 51 problems and *received* and *was incorrect* on all of them, and GLM-5.3-flash ran these two only (Appendix E). Each budget regenerated its own first attempts, so the two DeepSeek-flash arms are not paired. Truncated: replies that exhausted the budget. The last column counts replies whose reasoning exceeded 4,096 tokens.

Model	Budget	Message	Lost	Truncated	Median reasoning	>4,096
DeepSeek-flash	4,096	received	6 / 158	6	483	–
		received + format	10 / 158	10	441	–
		review	19 / 158	14	1,878	–
		review + format	11 / 158	11	1,145	–
		was incorrect	117 / 158	116	4,096	–
DeepSeek-flash	32,768	received	2 / 163	0	578	8
		received + format	0 / 51	0	430	1
		review	2 / 51	1	1,914	4
		review + format	0 / 51	0	1,127	3
		was incorrect	12 / 163	4	6,367	121
GLM-5.3-flash	32,768	received	0 / 160	0	330	1
		was incorrect	7 / 160	0	1,435	29

metric erratum for E13). Under that operational measure, the prediction that review would raise failures by at least 20 for three of five models failed: it did so for Qwen3-8B and Qwen2.5-Coder-7B, not for Mistral-7B or Llama-3.1-8B, and Gemma-3-12B's review arm was stopped before completion to free the devices. That restating the format would never increase fence-dropping held for every model scored (Gemma-3-12B on the 308 tasks its review arm reached), and so did the prediction that no message is safe for every model.

- **Sampling (Section 5).** That each of the four cells losing more than half of a model's correct answers under greedy decoding would still lose at least 40% when sampled: failed, because Qwen2.5-Coder-7B under *received* fell to 27%. That cells at or below 5% would stay at or below 15%: held. That *received* would lose at least 30 points more than *was incorrect* for Gemma-3-12B and Qwen2.5-Coder-7B: failed for the latter, at 20 points. That no message would be safe for all six models: held. The study was reduced from three seeds to one before its first call, because measured throughput made three infeasible; seeds two and three were queued as an extension and did not run.
- **Scale (Section 5).** That Llama-3.1-70B's worst message would still destroy at least 10% of its correct answers: held, at 19.2% under *was incorrect*. Three secondary predictions

Table 12: Intrinsic self-correction on 200 GSM8K problems in full. Accuracy in %, last boxed answer / last number anywhere, two alternative extraction rules. Gained: incorrect first answers that end correct. Lost: correct first answers that end incorrect, with those lacking a boxed answer in parentheses, and for R$^-$ also those whose last number is still correct. Truncated: replies cut at 1,024 tokens. Gained and Truncated give R / R$^-$. p: exact McNemar test of R against R$^-$. Gemma-4-12B was added after registration.

Model	First	R	R$^-$	Gained	Lost, R	Lost, R$^-$	Truncated	p
Llama-2-70b-chat	57.5	45.5 / 44.5	22.0 / 41.0	10 / 10	34 (0)	81 (57, 35)	1 / 0	2×10^{-11}
Llama-3.1-70B	95.0	87.5 / 88.0	85.0 / 84.5	3 / 2	18 (3)	22 (5, 2)	3 / 4	0.23
Llama-3.1-8B	88.5	51.5 / 55.0	49.0 / 50.5	4 / 4	78 (34)	83 (43, 8)	33 / 38	0.50
Mistral-7B	62.5	57.5 / 49.5	53.5 / 47.5	6 / 3	16 (0)	21 (9, 4)	2 / 0	0.12
Gemma-3-12B	92.0	85.5 / 85.5	85.0 / 80.5	1 / 2	14 (0)	16 (2, 1)	1 / 1	1.00
Gemma-4-12B	96.0	96.0 / 96.0	95.0 / 95.0	2 / 1	2 (0)	3 (0, 0)	0 / 0	0.50
Qwen3-8B	94.0	94.5 / 94.5	94.5 / 93.0	2 / 2	1 (0)	1 (0, 0)	0 / 0	1.00
Qwen3-32B	93.5	93.5 / 93.5	92.5 / 92.5	1 / 1	1 (0)	3 (0, 0)	0 / 0	0.50
GLM-5.3-flash	96.5	96.5 / 96.5	96.5 / 96.5	4 / 4	4 (0)	4 (0, 0)	0 / 0	1.00
DeepSeek-flash	97.0	96.5 / 96.5	97.0 / 97.0	1 / 1	2 (0)	1 (0, 0)	0 / 0	1.00

Table 13: Every experiment after the first, in order. Frozen: protocol fixed before its first model call (E28 and E29 made none). Informed: the model, sample or arm was chosen after seeing an earlier result. Details and every amendment follow in this section.

Experiment	Section	Frozen	Informed	Registered outcome
E11 four messages, six models	3	yes[a]	no	three predictions held, one withdrawn
E12 false verdict	4	yes	no	two failed, one held
E13 BIRD, every message	5	yes	no	three held, one failed
E14 sampling at $T = 0.7$	5	yes	no	two held, two failed
E15 Llama-3.1-70B	5	yes	no	primary held; three secondary failed
E16 paraphrases	5	yes[b]	no	review family failed; three held
E17 reading back the replies	4	yes	no	descriptive, no prediction
E18 self-correction, GSM8K	5.3	yes[c]	no	two held, two failed
E19 API models	5	yes	no	persists (GLM), shrinks (DeepSeek), budget
E20 Gemma-4-12B	5	yes[d]	no	two failed, one held
E21 box or request	5.3	yes	yes	the request
E22 submission cue and contract	4	yes	yes	both thresholds met
E23 SQL contract, Qwen3-8B	5	yes	yes	mixed
E24 same-history baseline	4	yes	yes	large-rescue gate passed
E25 SQL baseline arm	5	yes	yes	descriptive; E23 class unchanged
E26 SQL, Qwen2.5-Coder-7B	5	yes	yes	contract gate passed; request null
E27 SQL, Gemma-3-12B	5	yes	yes	request sufficient
E28 replay of E27 discordances	E	yes	yes	all 11 reproduced
E29 gate-value screen	E	yes	yes	census gate failed
E30 gpt-oss-20b and Phi-4	3	yes[e]	yes	three held
E31 same-history false verdict	4	yes	yes	reversal held, counts repeated

[a] Analysis script written after the first model's results. [b] Second stage unregistered. [c] Gemma-4-12B added without an amendment. [d] Serving change not recorded in advance. [e] Batched decoding, declared before the first call.

failed: that restating the format would be its safest message (14.6%, against 4.6% for *review + format*); that its worst message would differ from Llama-3.1-8B's (both are *was incorrect*); and that at least half of those losses would lack code (none of 25 does).

- **Paraphrases (Section 5).** That at least four of five other review requests would destroy at least 30% of Qwen3-8B's correct answers: failed, since one did (R2, 32%) and the others destroyed between 0% and 5%. That at least three of four other acknowledgements would do so for Qwen2.5-Coder-7B: held (A2 100%, A3 89%, A5 100%; A4 29%). That in every such cell at least 70% of losses would be replies without code: held. That both originals would reproduce the earlier replies on every accepted problem: held. As registered, the

failed prediction reduced the self-refinement paragraph of Section 3 to one example that names its wording.

- **The next generation (Section 5).** That *received* would destroy at least half of Gemma-4-12B's correct answers: failed, at 0 of 157. That *was incorrect* would destroy fewer than *received*: failed, 1 against 0. That restating the format would keep the loss at or below 5%: held (1 of 157). Whether at least half of the *received* losses would lack code could not be scored, since there were none.
- **API models (Section 5).** The registered outcome with thinking disabled was the largest loss across the five messages, read as persisting at 10% or more, shrinking between 5% and 10%, and vanishing below 5%. GLM-5.3-flash persists at 12.7%, and its three secondary predictions held: its worst message is not the false verdict, restating the format does not add losses, and 19 of its 20 worst-message losses lack a code fence. DeepSeek-flash shrinks, at 5.7%, and its worst message is the false verdict, against our prediction. With thinking enabled and 32,768 tokens, the registered outcome for DeepSeek-flash was *budget*: under *was incorrect* it lost fewer than 20% of its correct answers and fewer than ten replies were cut off.
- **Self-correction (Section 5.3).** That at least half of the models would lose accuracy under R: held (six of the nine registered models). That fewer than a quarter of the losses under R would lack a boxed answer, for every model: failed, through Llama-3.1-8B alone (34 of 78), whose unboxed losses are mostly replies cut off at the token limit. That deleting the last sentence of R would add at least ten losses without a boxed answer for half of the models: failed, since it did so for Llama-2-70b-chat alone (57 against 0). That Llama-2-70b-chat would drop at least five points under R: held, at 12. Gemma-4-12B was added to this experiment after registration without an amendment, so these verdicts exclude it, and including it would change none of them.
- **The box or the request (Section 5.3).** After an external reader noted that R^- deletes the request to restate the answer and the box together, we registered a third arm that deletes only the box. The registered outcome for Llama-2-70b-chat was read from the number of correct first answers lost without a box: at least 29 would mean the box, at most 11 the request. It lost 2, so the outcome is the request (Table 8).

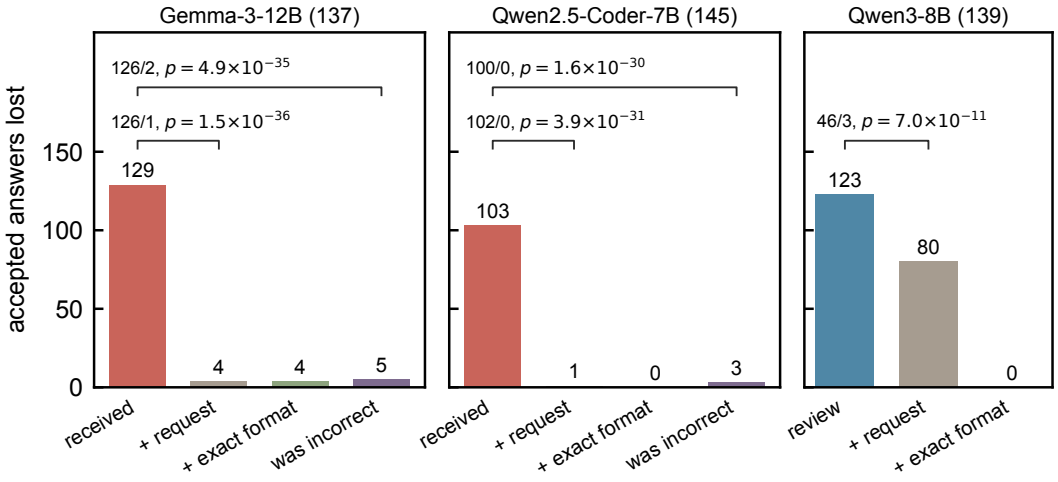

Figure 14: Same-history reruns on HumanEval (E22, E24, E31): accepted answers lost under each arm, on the same complete first histories (accepted answers in parentheses). Brackets give the paired discordance, lost only under the left arm / only under the right, and the exact McNemar p.

- **Submission cue versus contract (Section 4).** E22 registered three models and four messages before its first call. Both acknowledgement models met the generic-request threshold, and Qwen3-8B's review stem met the contract threshold (Figure 14). All 164 first extracted-code digests and rewards matched E9 per model, but E9 and E11 kept no complete first reply text, so only E22's within-run comparisons are fully history-paired.

- **Same-history submission baseline (Section 4).** E24, frozen after E22's outcome, reran the unsuffixed and generic-suffixed messages in balanced order on every complete E22 first history. All 164 source identities and all accepted pairs closed per model, and the 421 generic replies repeated E22's exact text and reward. Gemma-3-12B and Qwen2.5-Coder-7B pass the frozen large-rescue gate; Qwen3-8B's review contrast had no rescue classification (Figure 14). E24 is an outcome-informed internal-validity follow-up, not independent confirmation.
- **Same-history false verdict (Section 4).** E12 matched first answers only by their extracted code, so E31, frozen after E12's outcome, reran *received* and *was incorrect* on every complete E22 first history of Gemma-3-12B and Qwen2.5-Coder-7B, in balanced order within one run. Its registered reversal required at least 50 losses under *received*, at most 10 under the false verdict and an exact paired $p < 0.001$ for each model. Both held with E12's counts exactly (Figure 14), and no reply under the false verdict lacked a code block. Like E24, it is an outcome-informed internal-validity check, not independent confirmation.
- **SQL contract transport (Section 5).** Parser failures under the three arms are in Figure 4, and their paired flips in Figure 6.
  - *E23, Qwen3-8B.* It selected 200 accepted BIRD histories by hash before seeing their review outcomes, and its historical parser-failure gate passed at 137/200. The generic and contract arms then failed 71 and 0 times ($p = 8.5 \times 10^{-22}$), so the registered class is *mixed*: 71 is below the 100 required for "contract needed". A pre-call addendum made official-fence parser failure the primary outcome, and all 71 generic failures also literally lacked an SQL fence.
  - *E25, the same histories.* Frozen after those outcomes, it added the unsuffixed *review* arm to E23's histories ($p = 5.25 \times 10^{-16}$ against the generic arm) and repeated E23's generic and contract replies exactly. It is outcome-informed, not an independent replication, and leaves E23's class unchanged.
  - *E26, Qwen2.5-Coder-7B.* It repeated the three messages on 200 separately selected histories. Its v1 population check stopped before any call, because stored validation accepted 787 of 997 tasks rather than the 767 of E13's later revalidation, and a pre-call amendment changed only that count. The contract gate passed (generic to contract $p = 1.19 \times 10^{-7}$), but the generic request did not transport (baseline to generic $p = 0.868$). Six of the 200 stored acceptances fail current recheck; removing all six in the worst case, from aggregate counts only, still leaves at least 16/18/0 failures and 18 repairs with no harms ($p \leq 7.63 \times 10^{-6}$), so every frozen cross-model gate holds.
  - *E27, Gemma-3-12B.* Frozen after E25 and E26, it ran the three messages on 200 histories selected by hash from 836 stored acceptances. Gemma-3 was chosen as a second family after the Qwen results, and the protocol committed in advance to this family and to reporting a null. Its registered class is *generic request sufficient*: at least 15 baseline failures, at most five in each new arm, and at least 10 more repairs than harms with $p < 0.01$ (observed $p = 5.6 \times 10^{-6}$). Four of its stored acceptances fail current recheck.
  - *Correct first answers.* Among currently first-correct answers (129, 121 and 110), the contract rescues 57 of Qwen3-8B's generic losses with none in reverse, loses one against the generic arm's 16 for Qwen2.5-Coder-7B ($p = 0.000275$), and loses 11 against 4 for Gemma-3-12B (paired 2/9, $p = 0.065$). Reward and stratum comparisons with E13 are sensitive to SQL execution timing.
  - *E28 and E29, no model calls.* Both re-scored E27's stored replies and were frozen after its outcome. E28 replayed the 11 histories on which the generic and contract arms disagree in reward, so it is selected on that outcome; a label check stopped its first version before scoring, and an amendment changed only that check. All 11 reward vectors reproduced in three passes, and five of the contract's nine harms are accepted SQL that returns rows and scores wrong in under 0.145 seconds, so the time limit does not explain them. E29 read all 200 rewards under a registered screen: the arms change the number correct by -9, 0 and -3, short of the ten-task change that would have opened the remaining 636 accepted histories.
- **Two further families (Section 3).** E30 sent the five messages to OpenAI's gpt-oss-20b (medium reasoning effort, 32,768 tokens) and Microsoft's Phi-4. Both were chosen after

every earlier result and served with batched greedy decoding, as the protocol declared before the first call, so no replay claim is made. That at least one of the two would lose at least 19% of its correct answers to some message: held, since gpt-oss-20b loses 100 of 161 (62.1%) under *review*. Of those 100, 59 replies carry text but no code, 39 exhaust the token budget before any answer and 2 submit a failing function. That where most of at least ten losses lack code, the same message with the format restated would lose at most half as many: held for the one qualifying cell, 100 against 40, and 39 of the 40 are budget exhaustions. That the false verdict would lengthen gpt-oss-20b's reasoning: held, with medians of 13,281 against 1,017 characters. Phi-4 loses at most 4 of 143 correct answers, all as failing functions. Greedy decoding is not the sampling recommended for gpt-oss, and greedy reasoning can loop, so the budget exhaustions may overstate what recommended sampling would produce.

Amendments. Each amendment was written into its protocol before the calls it governs. Llama-2-70b-chat received the official Llama-2-chat template, since its mirror carried none, and GLM-5.3-flash was added to the self-correction and API experiments. After DeepSeek-flash's 4,096-token thinking arm, a 32,768-token arm was registered, and after its first 51 problems it continued with *received* and *was incorrect* only, to limit cost.

Unregistered additions. Two additions carried no registration. The second stage of the paraphrase test (Section 5, Figure 13) ran on devices that happened to be free and carried no prediction. Gemma-4-12B was run in the self-correction experiment without being registered for it, so that experiment's verdicts exclude it.

Changes not recorded in advance. We found two changes that no protocol recorded before the calls, and recorded each once we found it. The protocol for Gemma-4-12B named the vLLM server of the other open models, but this multimodal checkpoint had only ever been served through Hugging Face Transformers, and the run used that server on an RTX 5090 with an 8,192-token context. Decoding is greedy in both, but the comparison with Gemma-3-12B therefore spans an implementation as well as a model generation. GLM-5.3-flash's thinking arm ran only *received* and *was incorrect*, like the tail of DeepSeek-flash's, although its amendment implied all five messages; these two are the messages that the registered verdict and the reasoning multiplier use.

Unregistered analyses. A late zero-call check applied one rule, fixed before its outcome, to the three models with complete first histories: a strict format gate that rejects any first reply other than one fenced block. It routes none of their accepted answers (Appendix D). We split accepted problems by whether all six original models solved them at the first attempt (60 problems) or not. Format losses show no dependence on that split: Gemma-3-12B under *received* loses 59 of 60 and 70 of 77, Qwen3-8B under *review* 54 of 60 and 69 of 79, and Qwen2.5-Coder-7B under *received* 45 of 60 and 58 of 85 (two-sided Fisher exact $p = 0.08$, 0.79 and 0.46). We looked for the opposite pattern in the semantic channel and did not find a significant one. Llama-3.1-8B's losses are slightly more frequent on the harder problems under every message, but no p is below 0.14.

F Retry as Allocation: The Registered Design

This study was registered as a measurement of how a semantic validator allocates compute. The accepted-stratum results of the main text grew out of it, and this appendix reports the registered design and its results in full. It supplies the targeting view of Section 2; the main text studies the one quantity this design cannot observe, the continuation of accepted answers.

Decomposition. Freeze a generic second turn and write Z_2 for its reward on every first history and $D = Z_2 - Z_1$. If the deployed harness sends its own message, with reward Z_r, on the rejected stratum, then with $F = Z_r - Z_2$ there,

$$V_H - V_1 = p\,\mathbb{E}[D] + \mathrm{Cov}(T, D) + p\,\mathbb{E}[F \mid T = 1].$$

The three terms are the value of the same number of turns granted at random, the value of using the validator as the routing signal, and the value of the deployed message over the generic one. Their signs need not agree. Under a fail-closed score, $Z_1 = 0$ on every rejection, so $V_H - V_1 = p\,\mathbb{E}[Z_r \mid$

$T = 1] \geq 0$ for every retry policy, and retry gain alone cannot show that the validator routes turns well. Here V_H is V_{gated} of Section 2, and Z_1, Z_2 are its rewards z_1, z_2. When the deployed message is the generic one ($F = 0$), Proposition 2 is the same covariance seen from a released evaluation. When they differ, a gated log reveals $\mathbb{E}[D + F \mid T = 1]$ rather than $\mathbb{E}[D \mid T = 1]$, and the set widens as in the weaker-log case of Appendix B.

Allocation differs by model. On the 997 BIRD tasks of Section 5, scored with the official post-processing, the five models' mean calls per task differ by a factor of up to 1.33 and their mean tokens by a factor of up to 1.65. Most of this spread comes from one model, since Mistral-7B's first query fails to execute or returns no rows on 54% of the tasks, and without it the call ratio is 1.14. In this census the first-attempt accuracy of the five models and the compute each was granted are in exactly reverse order.

Decision grids. The primary registered diagnostic enumerates 942 cells of a finite decision problem (157 public belief states, two horizons and three equivalent presentations), with an exact dynamic program for the optimal query and action. Each cell asks for one JSON decision: which unseen binary slot to inspect, or none (-1), and a 0/1 prediction for every slot. The validator rejects a reply that does not parse to this schema, including predictions written as -1 or `null` and text cut off at the token cap, and a rejected reply that is not retried receives the declared fallback of no inspection. The dynamic program scores whether the chosen action is optimal (action), whether the inspection choice is optimal (query), and the value lost against the optimum (regret). Qwen2.5-7B and Gemma-4-12B, checkpoints used only in this appendix's design and in Section 5 for Gemma-4-12B, answer every cell greedily, and every cell also receives one generic continuation. Table 14 gives the three policies, and Figure 15 the decomposition. Qwen2.5-7B's validator triggers on 3 of 942 cells, and the generic turn makes all three lose, so its targeting term is negative although the generic turn helps on average. Gemma-4-12B triggers on 88 cells, and its targeting term is negative for the exact action and positive for the query decision, so the sign depends on the reward. Retry also reverses an interface conclusion for Gemma-4-12B: the presentation that ranks last at the first attempt ranks first after deployed retry, because it triggers retries and the winner under the first attempt triggers none.

Table 14: Three policies on the 942-cell decision grids. Higher is better for action and query, lower for regret.

Model	Policy	Action	Query	Regret
Qwen2.5-7B	Fixed first	.4650	.5711	.1589
	Deployed retry	.4618	.5679	.1592
	Generic turn for all	.5807	.5913	.1277
Gemma-4-12B	Fixed first	.7813	.8248	.0275
	Deployed retry	.7675	.8429	.0169
	Generic turn for all	.7633	.8355	.0192

PM-Bench. On 80 fixed decision prefixes from released PM-Bench trajectories (Liu & Gabriel, 2026; Zhao & Wu, 2026), the validator triggers on 38 prefixes for Qwen2.5-7B and 43 for Qwen3-8B. Exact feedback repairs 12 and 5 of them, a generic turn repairs one each, and Qwen3-8B's targeting term is -0.0058. In a complete simulated week, retry with exact feedback repairs 15 of 57 first rejections, yet it lowers Qwen3-8B's set F1 from .3623 to .3433 while raising completed calls from 365 to 431. A positive local repair rate therefore need not imply a positive evaluation effect.

BIRD census. On a prospective census of 104 BIRD Mini-Dev tasks that no earlier run had touched, we registered the prediction that Qwen3-8B's validator would target turns badly. Its validator requires exactly one read-only SQL fence that executes and returns a nonempty result. Every task receives the same generic shadow turn, only rejections receive an exact error-class turn, and gold accuracy is read only after all calls close. It triggered on 23 tasks and captured two of the three gains of a generic turn, so the targeting term was positive, $+.01285$, and the registered direction failed. Three new continuations that keep the first answer in context also targeted positively, while a fresh restart without it reproduced every first answer and changed nothing. The allocated turn is therefore a history-conditioned operator rather than a fresh draw.

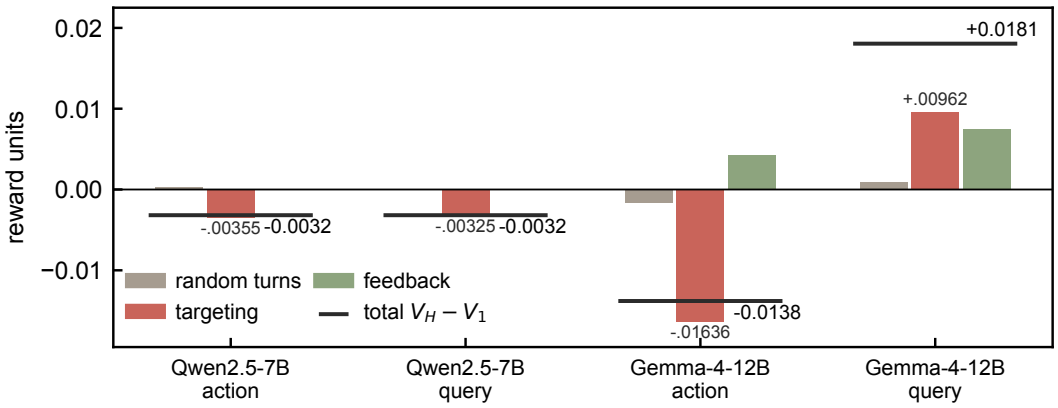

Figure 15: Decomposition of $V_H - V_1$ on the grids, in reward units. Random turns: $p\,\mathbb{E}[D]$; targeting: $\mathrm{Cov}(T, D)$, labelled; feedback: $p\,\mathbb{E}[F \mid T = 1]$. Black lines give the totals.

Stratified audit of the targeting term. For each of these four new arms, the audit shadowed all 23 triggers and a simple random sample without replacement of 41 of the 81 other tasks, drawn with a preset seed before any outcome, so each untriggered task entered with probability 41/81. It estimates $\mathrm{Cov}(T, D)$ by $p(1-p)(\bar{D}_1 - \bar{D}_0)$, the gap between the mean effect over all triggers and over the sampled other tasks, with design variance $p^2(1-p)^2(1-41/81)\,s_0^2/41$, where s_0^2 is the sample variance of the sampled effects, and a normal 95% interval. Each estimate was committed before the remaining 40 tasks were run and used 64 of 104 calls, 38.5% fewer than a census. All four intervals covered the census values, with absolute errors of at most 0.0063. The interval describes the sampling design only, not new tasks or messages, and one set of covered intervals does not establish nominal coverage.