AFTER A FAILED REPAIR: UNDERSTANDING HOW LLMS RESPOND TO DIAGNOSTIC INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A failed language-model repair does not identify what the model needs next. Another sample may already fix the program; a more directive prompt may change its behavior; or the model may need valid evidence about the defect. We study these possibilities through controlled post-failure interventions that separate fresh attempts, a diagnostic-shaped prompt containing no diagnosis, and increasingly specific task information. Our strongest test observes failures prospectively: using one provider/model configuration throughout, we first give each fresh defect one ordinary repair attempt, freeze only executable failures, and then launch new calls under each condition. In two separately analyzed populations, blind best-of-3 retry recovers 52.4% and 49.0% of failures. A structure-only prompt does not improve on this baseline, while recovery rises to 83.5% and 82.8% when the faulty source line is revealed. Historical experiments show the same broad diagnostic responsiveness but a much lower blind baseline and an apparent structure effect, demonstrating that failure-selected populations are themselves part of the measurement. Additional Xing and GLM results show that response to the same information is model dependent. Finally, passing tests and agreeing with a reference are not interchangeable correctness checks: for `count_Squares(10,20)`, a passing candidate returns 935, the reference returns 770, and an independent brute-force oracle returns 935. Repair evaluation should therefore report how failures were selected, how many new attempts were allowed, what information was supplied, and what evidence supports correctness.

1 INTRODUCTION

Suppose a comparison operator is reversed. A language model first returns the buggy function unchanged. What did that failure reveal? Perhaps a second draw will edit the condition with no new evidence. If not, telling the model "this is a comparison defect" narrows the search; pointing to line 4 narrows it further; showing the faulty line nearly identifies the edit. These interventions answer different questions even when each eventually produces a passing program.

Program-repair scores often compress this sequence into one bit. Yet repeated sampling changes code-generation outcomes (Chen et al., 2021), and feedback can guide revision (Austin et al., 2021; Dai et al., 2025; Le et al., 2026). Measuring only the final patch obscures whether success came from another sample, a more directive prompt, or task-specific information. Selection adds a second ambiguity. If tasks enter a study because they failed in an old run, the retry baseline is conditioned on that old outcome. A set of "model failures" is therefore not a fixed unit of difficulty.

We study what happens *after an observed failure*. Historical controlled experiments first compare blind retry, a matched prompt with no valid diagnosis, and increasingly specific defect information. They suggest two effects: renewed prompt structure may help, and real diagnostic content helps more. We then apply a cleaner prospective design. For each fresh task, one ordinary attempt determines eligibility. Only executable failures are frozen and sent to five independent best-of-3 conditions; the screening output is never reused. Two separately constructed populations use the same accessible DeepSeek provider/model configuration for screening and intervention.

The prospective evidence isolates the transferable result. Blind repetition alone recovers roughly half of failures. The matched structure-only prompt does not improve over blind retry, but valid diagnostic information produces a large additional gain, reaching about 81–82% under the strongest localization

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.

oracle. Thus, diagnostic responsiveness transfers prospectively; the apparent structure-only effect and the absolute retry baseline do not. Historical DeepSeek, Xing, and GLM results further show that response to the same information is model dependent, without revealing a unique internal mechanism. Supplied prospective GLM aggregates show the same broad diagnostic pattern, although their cohort accounting is not complete enough for paired inference.

Success raises a final measurement question. Supplied tests are partial specifications, and a benchmark reference can itself be wrong or underspecified. In one audited case, `count_Squares(10,20)`, the candidate and an independent brute-force oracle return 935 while the reference returns 770. A disagreement is useful evidence for investigation, not an automatic semantic-error label.

Our contributions are empirical:

- a controlled framework for post-failure measurement that separates repeated attempts, non-diagnostic restructuring, and valid task-specific diagnostic evidence;
- a prospective same-provider/model-configuration study on two separate failure populations, showing that diagnostic gains transfer while the historical structure-only gain does not; and
- a correctness audit demonstrating that evaluator passage and reference agreement provide different, individually incomplete evidence.

2 Related Work

Code repair and feedback. HumanEval, MBPP, and QuixBugs use executable tests to evaluate code generation and localized repair (Chen et al., 2021; Austin et al., 2021; Lin et al., 2017). FeedbackEval compares feedback types in code repair, Code Reffix separates reflection quality from repair, and PAIR-Bench studies progressive hints and repair trajectories (Dai et al., 2025; Di et al., 2026; Le et al., 2026). Repository-level benchmarks study broader issue resolution (Jimenez et al., 2024; Yang et al., 2024). Our question is narrower and methodological: after failure has been observed, how much later success comes from fresh sampling, non-diagnostic prompt form, or valid defect evidence?

Validity of repair evaluation. Passing tests can leave behavioral discrepancies or underspecified intent. PatchDiff differentially tests plausible patches, while trajectory studies examine failures hidden by final scores (Wang et al., 2026; Majgaonkar et al., 2025; Ehsani et al., 2026). We complement this work by controlling how failures enter the sample and by checking candidate–reference disagreements against specifications and executable oracles.

3 From a Failed Attempt to an Intervention

3.1 Five Ways to Continue

Figure 1 begins with an illustrative function whose comparison operator is wrong. The first model call fails. We then reset the same buggy program before every intervention; no candidate becomes the starting point for another condition.

The conditions differ only in what the next call receives. **BLIND** asks for another repair with no new evidence. **STRUCTURE** uses the same length and diagnostic-shaped format as an informative prompt but supplies no valid diagnosis; historical records call this condition PLACEBO. **COARSE** names the mutation family, such as "comparison defect." **LINE** adds the correct source line. **STRONG** also displays the current faulty line. STRONG is therefore a localization oracle, not information assumed to be free in a practical repair system.

Each condition makes three fresh calls. A task is recovered if at least one candidate passes the unchanged evaluator, which checks the output contract and executes the supplied tests. This best-of-3 outcome measures recovery within a fixed additional budget. It is not a single-attempt success probability.

The historical study also asks whether two concrete evidence sources contribute differently. Its 2×2 design independently supplies the correct defect location and actual failing-test feedback. Localization says where to inspect; execution feedback shows an observed behavioral mismatch. Their combination tests whether each source still contributes after the other is available.

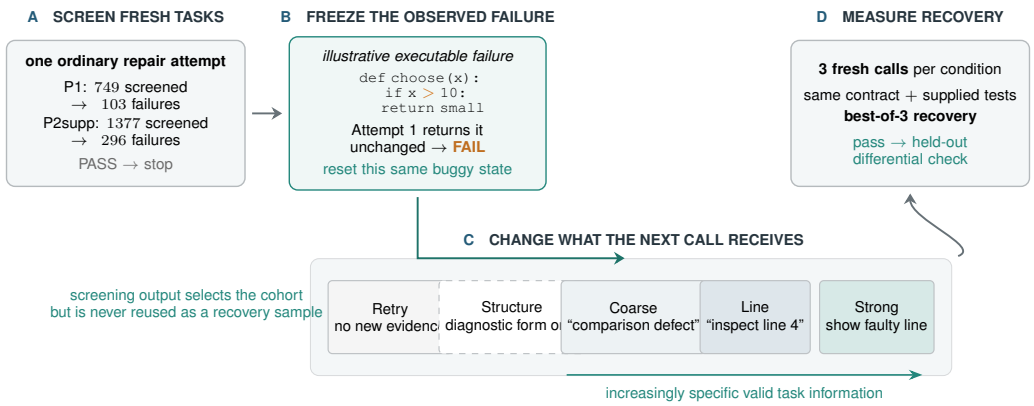

Figure 1: **Observe failure first; intervene second.** A fresh ordinary attempt determines whether a task enters a prospective cohort. The executable failure is frozen before any intervention. BLIND changes only the sample; STRUCTURE changes prompt form without a valid diagnosis; COARSE, LINE, and STRONG reveal progressively more task-specific information. Every condition receives three fresh calls and the same evaluator.

Table 1: The five post-failure conditions change either the sample, prompt form, or valid task evidence.

Condition	What the next call receives	What the contrast measures
BLIND	Original repair request; no new evidence.	Recovery from three fresh samples.
STRUCTURE	Matched diagnostic-shaped format, but no valid diagnosis.	Prompt-form effect beyond BLIND.
COARSE	Mutation family, e.g. "comparison defect."	Response to a restricted defect class.
LINE	COARSE information plus the correct line number.	Value of explicit localization.
STRONG	LINE information plus the current faulty source line.	Recovery under a strong localization oracle.

3.2 Historical and Prospective Failure Populations

The historical screen contains 919 synthetic repair cells: 167 MBPP Phase-2, 602 MBPP Phase-2B, and 150 HumanEval tasks (Austin et al., 2021; Chen et al., 2021). It records 741 evaluator passes and 178 failures. Twenty failures are empty output-contract responses. Because ordinary resampling recovers all 20 in the historical factorial study, the five-condition information analysis uses the remaining 158 executable non-EMPTY failures. The model configuration for every cell in the original 919-task screen is not established by the supplied record; its 80.6% pass rate is therefore not attributed to a later intervention model.

Retrospective selection is useful for discovery but entangles the population with an old outcome. The prospective design instead observes failure before intervention. Two fresh populations, P1 and P2supp, are constructed independently and never pooled. Each task first receives exactly one ordinary repair attempt. A task enters the intervention cohort only if this attempt is an executable failure. The screening candidate is used only for selection and is never counted among the three subsequent BLIND calls.

Within each prospective population, screening and all intervention calls use SiliconFlow's `deepseek-ai/DeepSeek-V4-Flash` endpoint with temperature 0.3, top-p 1.0, maximum output length 8,192, and thinking disabled. This is the same provider/model configuration *within* the prospective experiment; we do not claim that it is the identical checkpoint used by historical runs. P1 screens 749 tasks and admits 103 observed failure samples. P2supp screens 1,377 tasks and admits 296. No intervention outcome affects cohort membership.

The prospective GLM replication uses SiliconFlow's `glm-5.3-flash` endpoint with temperature 0.3, top-p 1.0, maximum output length 8,192, and thinking disabled. Its aggregate G1/G2 rates are

reported as a descriptive cross-model check. The supplied summary does not establish the G1/G2 screening manifests, exact cohort denominators, or task-level pairing with P1/P2supp, so these runs are not included in the DeepSeek prospective paired tests.

The two rows are replications, not strata of one larger estimate. Their task composition may differ, and their rates are never averaged. The shared protocol is what supports comparison: an ordinary attempt is observed first, only executable failures are retained, and all later calls use the same frozen task state and evaluator.

3.3 Conditional Recovery as the Estimand

The study does not estimate a model's unconditional probability of repairing an arbitrary program. It estimates recovery after a particular selection event. Let $S = 1$ denote that the one-shot screen produced an executable evaluator failure, let $E(y)$ indicate that candidate y passes the evaluator, and let P identify a failure population. For condition c, the reported quantity is

$$R_c(P) = \Pr\left[\max_{j\in\{1,2,3\}} E(y_{c,j}) = 1 \mid S = 1, P\right]. \tag{1}$$

The maximum implements the fixed best-of-3 budget. Conditioning on $S = 1$ is essential: without the observed failure, the quantity would mix initial successes with post-failure recovery.

The screening candidate is excluded from $y_{c,1:3}$. Otherwise BLIND would receive only two genuinely new opportunities while its recorded result also reused the event that selected the task. Resetting the program before every call likewise prevents one condition's edit from changing another condition's starting point. These choices make the contrasts about sampling and supplied information rather than inherited repair state.

The population index P is equally important. Historical failures were selected by an earlier screen whose complete configuration is unavailable; P1 and P2supp are selected by their own fresh screens under the prospective configuration. Consequently, $R_c(P_{\text{hist}})$, $R_c(P_1)$, and $R_c(P_2)$ answer parallel but non-identical questions. Comparing their curves reveals which qualitative effects transfer, while pooling their numerators would erase the selection rule that defines each estimand.

3.4 Analysis and Correctness Checks

We report the supplied task-level best-of-3 rates. Page's L tests the ordered response across BLIND–STRUCTURE–COARSE–LINE–STRONG and, separately, the informative sub-ladder BLIND–COARSE–LINE–STRONG. The latter removes the non-diagnostic condition from the ordering hypothesis. STRUCTURE is also compared directly with BLIND. Historical paired analyses use the same task identities within each experiment. We interpret these tests as evidence about observable responses, not latent model knowledge.

Passing candidates undergo an additional reference-differential check on novel inputs. A flag means that candidate and reference produce different outputs; it does not determine which is correct. The separate semantic audit reconstructs specifications and executable oracles for selected historical disagreements. Its purpose is to distinguish reference errors, ambiguous input domains, and candidate errors rather than to treat the reference as unquestioned ground truth.

4 What Actually Helps After Failure?

4.1 Prospective Failures Often Recover Under Retry

Trying again is already a strong baseline. In P1, three fresh BLIND calls recover 54/103 observed failure samples (52.4%). In P2supp, they recover 145/296 (49.0%). These calls occur only after the screening attempt and do not reuse it. Thus, roughly half of the prospective failures are recoverable without any new task evidence.

The matched STRUCTURE prompt does not improve this baseline. Recovery is 44.7% in P1 and 40.5% in P2supp, compared with 52.4% and 49.0% under BLIND. The paired comparisons are not significant ($p = 0.344$ and $p = 0.302$). This does not prove that wording never matters; it shows that non-diagnostic restructuring does not explain the prospective diagnostic gains.

Table 2: Prospective cohorts are selected by one fresh attempt before any intervention. P1 and P2supp are analyzed separately.

Population	Screened	Failure samples	Post-failure calls
P1	749	103	5 conditions $\times 3$
P2supp	1,377	296	5 conditions $\times 3$

4.2 Valid Diagnostic Content Adds a Large Gain

Once the prompt contains true information about the defect, recovery rises (Figure 2). In P1, COARSE, LINE, and STRONG recover 75.7%, 77.7%, and 83.5%. P2supp moves from 64.2% at COARSE to 79.4% at LINE and 82.8% at STRONG. Page's L supports an overall ordered response in both populations ($p = 0.00076$ and 4.8×10^{-5}). The informative BLIND–COARSE–LINE–STRONG sub-ladder is also supported ($p = 0.0197$ and 0.0020).

The prospective SiliconFlow GLM-5.3 Flash replication shows a closely related descriptive pattern. G1 moves from 53.2% at BLIND and 41.4% at STRUCTURE to 75.3%, 73.8%, and 80.1% under COARSE, LINE, and STRONG. G2 reports 47.1%, 39.8%, 62.7%, 77.9%, and 81.5%. Because the available summary does not provide the G1/G2 selection records, exact denominators, or task-level outcomes, these rates are not pooled with P1/P2supp and are not assigned the DeepSeek paired-test results.

The scientific point is not that every adjacent step always helps. P1 changes little from COARSE to LINE, while P2supp changes substantially. The stable contrast is broader: blind repetition recovers many failures, and valid task-specific information raises recovery well beyond it. STRONG reaches about 83% in both populations, but it does so by revealing the faulty line itself.

Novel-input reference-differential checking flags approximately 0–4% of passing candidates per prospective condition. STRONG has no flagged disagreements in either population. These are useful held-out checks, not semantic proofs: agreement with a reference on generated inputs cannot establish correctness over the full input domain.

4.3 Historical Evidence Explains Which Information Helps

The historical five-condition study shows the same broad diagnostic pattern under a different failure-selection rule. DeepSeek recovers 19.0% under BLIND, 37.3% under STRUCTURE, 55.7% at COARSE, 64.6% at LINE, and 76.6% at STRONG. Unlike the prospective populations, the historical set shows a sizeable structure-only increase. Figure 2 makes both the transferable diagnostic response and the non-transferable structure effect visible.

On the same 158 historical non-EMPTY failures, GLM-5.3 Flash recovers 30, 61, 82, 107, and 128 tasks across BLIND through STRONG, corresponding to 19.0%, 38.6%, 51.9%, 67.7%, and 81.0%. These percentages are recomputed from the supplied counts. GLM therefore shares the low historical BLIND baseline and strong response to valid diagnostic content, while its intermediate curve is not identical to either DeepSeek or Xing.

The localization–test factorial gives a more concrete decomposition. Across all 178 historical failures, BLIND recovers 28.7%, correct localization 78.1%, actual failing-test feedback 71.9%, and both 85.4%. In paired terms, localization alone has a net gain of 88 tasks over BLIND and tests alone a net gain of 77. Once localization is present, adding tests yields a net 13-task gain ($p \approx 0.035$); once tests are present, adding localization yields a net 24-task gain ($p \approx 3.9 \times 10^{-5}$). The two sources therefore overlap substantially. Localization is especially informative, while execution feedback contributes additional recovery for a subset of tasks.

The corresponding GLM factorial counts are 62/178 for BLIND, 126/178 for localization, 131/178 for failing tests, and 154/178 for both (34.8%, 70.8%, 73.6%, and 86.5%). The supplied paired net gains are $+64$ for localization versus BLIND, $+69$ for tests versus BLIND, $+28$ for tests given localization, and $+23$ for localization given tests. Because net gain subtracts task-level impairments from improvements, it need not equal the difference between marginal pass counts.

Table 3: Best-of-3 recovery (%). Historical and prospective populations have different selection rules and are not pooled. GLM G1/G2 are supplied descriptive aggregates with incomplete cohort accounting.

Model / population	BLIND	STRUCTURE	COARSE	LINE	STRONG
DeepSeek historical ($n = 158$)	19.0	37.3	55.7	64.6	76.6
DeepSeek P1 ($n = 103$)	52.4	44.7	75.7	77.7	83.5
DeepSeek P2supp ($n = 296$)	49.0	40.5	64.2	79.4	82.8
GLM G1	53.2	41.4	75.3	73.8	80.1
GLM G2	47.1	39.8	62.7	77.9	81.5

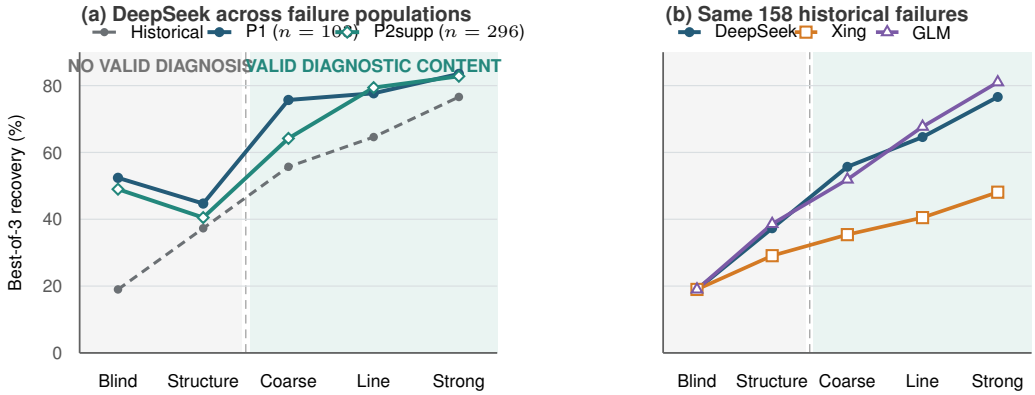

Figure 2: **Diagnostic responsiveness transfers; the blind baseline and structure-only effect do not.** (a) Historical DeepSeek failures have low BLIND recovery and a STRUCTURE increase. In both separately analyzed prospective populations, BLIND already recovers about half of failures, STRUCTURE does not improve it, and valid diagnostic conditions rise toward 81–82%. (b) On the same historical tasks, GLM tracks the stronger diagnostic response while Xing responds less strongly. Points are aggregate best-of-3 rates; populations are not pooled.

5 Response Depends on the Model and Failure Population

The historical Xing experiment applies the same five conditions to the same 158 failures. Its recovery rates are 19.0%, 29.1%, 35.4%, 40.5%, and 48.1%. DeepSeek, Xing, and GLM share the same aggregate BLIND rate but separate as information strengthens: GLM reaches 81.0% at STRONG, DeepSeek 76.6%, and Xing 48.1%. Their generated behavior also differs. The unchanged-code share falls from 29.7% to 3.2% for DeepSeek, from 28.3% to 3.1% for GLM, and from 51.9% to 35.7% for Xing. These are observable associations, not evidence that any model possesses a particular internal repair mechanism.

Population choice changes the curve just as strongly. Historical DeepSeek BLIND is 19.0%, versus 52.4% and 49.0% prospectively. STRUCTURE rises historically but not prospectively. In contrast, the STRONG endpoint is 76.6%, 83.5%, and 82.8% across the three populations. Diagnostic responsiveness transfers; the absolute retry baseline and structure-only effect do not. Two sets labeled "failures" can therefore have very different recoverability.

6 Passing Is Not the Same as Being Correct

The most instructive audit case begins with a disagreement, not an evaluator failure. For `count_Squares(10,20)`, a passing candidate returns 935 and the benchmark reference returns 770. Independent brute-force enumeration also returns 935 (Figure 3). Treating the reference as semantic truth would reject the candidate for matching the mathematical oracle.

Historical differential testing flags 9 disagreements among 175 deduplicated DeepSeek passes and 5 among 139 Xing passes. The supplied audit examines these 14 candidates with specifications, targeted

Table 4: Prospective statistical evidence reported separately for the two failure populations.

Population	Full ordered trend	BLIND–COARSE–LINE–STRONG	STRUCTURE vs BLIND
P1	$p = 0.00076$	$p = 0.0197$	$p = 0.344$
P2supp	$p = 4.8 \times 10^{-5}$	$p = 0.0020$	$p = 0.302$

Table 5: Historical localization × failing-test feedback on all 178 failures. Net gains are paired task-level improvements minus impairments for the stated contrast.

Condition	Recovery	Net gain	Comparison
BLIND	28.7%	—	fixed-budget retry baseline
Localization	78.1%	+88	versus BLIND
Failing tests	71.9%	+77	versus BLIND
Both	85.4%	+13 / +24	tests given location / location given tests

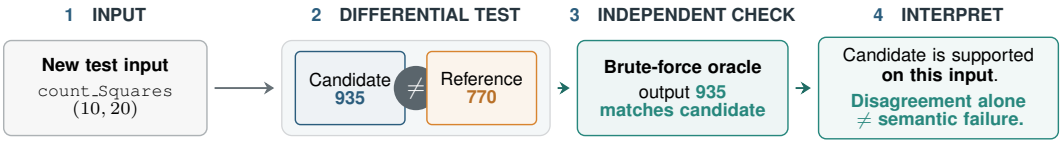

Figure 3: **A real reference-disagreement case.** On input $(10, 20)$, the candidate returns 935, the benchmark reference 770, and independent brute-force enumeration 935. Reference disagreement triggers adjudication; alone it neither establishes semantic failure nor proves general correctness.

inputs, executable oracles where possible, and independent model reviews. It reports many valid alternatives, but also cases that resist a binary label. Some `snake_to_camel` disagreements concern repeated underscores and empty segments whose intended domain is unclear. A `find_Points` candidate violates audited properties more often under two stated interpretations, supporting an interpretation-dependent incorrectness judgment rather than a proof. In `decreasing_trend`, the test polarity conflicts with the behavior suggested by the specification.

Table 6: Representative candidate–reference disagreements require different evidence.

Case	What disagrees	Warranted conclusion
`count_Squares`	Candidate 935; reference 770; brute-force oracle 935.	Reference error on the audited input.
`snake_to_camel`	Repeated underscores and empty segments.	Intended input domain is underspecified.
`find_Points`	Candidate violates tested properties more often under two interpretations.	Property-supported concern under stated interpretations.
`decreasing_trend`	Test polarity conflicts with the named/specification behavior.	Specification–test inconsistency.

The lesson is methodological. Supplied-test passage, reference agreement, and semantic correctness are three different measurements. A reference-differential flag is valuable because it exposes a behavior to inspect; by itself, it does not say which implementation is wrong.

7 Discussion

Failure is a starting state, not a diagnosis. A single failed candidate establishes only that one sampled repair failed the evaluator. In the prospective studies, three new blind attempts recover about half of such failures. Valid defect information raises recovery further. Later success should not be used retroactively to claim why the first attempt failed, and these observable response curves do not expose a model's unique internal deficit.

Failure populations are experimental objects. Historical blind recovery is 19.0%; prospective blind recovery is 52.4% and 49.0%. The historical structure-only gain also disappears prospectively. These differences are not nuisances to average away: the selection event, task distribution, model configuration, and failure definition jointly determine the population being measured. P1 and P2supp are therefore reported separately, never pooled.

The transferable effect is diagnostic, not merely structural. Relative to BLIND, STRUCTURE changes recovery by $+18.3$ percentage points historically but by -7.8 and -8.5 points in P1 and P2supp. STRONG, by contrast, exceeds BLIND by 57.6, 31.1, and 33.8 points. The absolute gains vary, but the prospective replications agree on the central distinction: a diagnostic-shaped prompt without a valid diagnosis does not explain the improvement obtained from real defect information.

Replication need not produce identical middle steps. P1 changes little from COARSE to LINE, whereas P2supp changes substantially. Both reach nearly the same STRONG endpoint. This pattern argues against interpreting any one hint level as a universal threshold. It instead supports a coarser conclusion: useful information changes the set of recoverable tasks, while the marginal value of operator family, line number, and source text depends on the selected population.

Cross-model evidence strengthens the measurement claim. DeepSeek, Xing, and GLM have the same historical BLIND aggregate but diverge as information strengthens; where task-level records are available, equal aggregates need not mean identical recovered tasks. The prospective studies then show that even one model's curve changes across freshly selected populations. Together, these results make a leaderboard interpretation inappropriate. The object being measured is a conditional response: a particular model, on a particular failure cohort, under a specified retry and information budget.

Reporting implications. Repair outcomes are most interpretable as a sequence of conditional measurements. The one-shot screen defines which failures are studied. BLIND adds a specified sampling budget. Diagnostic conditions reveal privileged task evidence. The evaluator then checks only the behaviors it encodes, and a differential test checks only the generated inputs it exercises. Table 7 states the corresponding claim boundaries.

Table 7: Each measurement supports a narrower claim than a single "repair reliability" score.

Observation	Supported statement	Unsupported inference
One-shot failure	One sampled repair failed the configured evaluator.	The task exceeds the model's capability.
BLIND best-of-3	The frozen task is recoverable within three fresh attempts.	The next attempt succeeds with that rate.
Diagnostic gain	Outcomes change when specified valid evidence is supplied.	The cue reveals the model's unique internal deficit.
STRONG recovery	Repair improves when the faulty line is exposed.	The system can obtain that oracle information itself.
Reference disagreement	Candidate and reference differ on an audited input.	The candidate is necessarily incorrect.

Implications for benchmark design. A post-failure benchmark should preserve the screen that created its cohort, exclude the screening candidate from later retry samples, and keep task state fixed across interventions. It should include a non-diagnostic control before attributing gains to feedback content. When two failure populations arise from different screens or configurations, their absolute rates should be reported separately rather than pooled into a single repair score.

Implications for correctness claims. Evaluator passage should be accompanied by the scope of the tests that produced it. Reference-differential checking can expose new behaviors, but disagreements require adjudication against the specification or an independent executable oracle. This layered reporting is more cumbersome than pass/fail, but it prevents sampling luck, privileged localization, and reference defects from being folded into one claim about model capability.

8 LIMITATIONS

The strongest condition is an oracle. STRONG reveals the operator family, correct line, and faulty source line. It measures response when localization is largely supplied, not the end-to-end ability to obtain that information in practical debugging. COARSE and LINE also expose privileged mutation information. All experiments use localized synthetic defects; they do not establish repository-level repair performance.

Scope and uncertainty. The primary prospective studies cover one DeepSeek provider/model configuration, temperature 0.3, and best-of-3 recovery. Best-of-3 is a fixed-budget property, not the probability that the next attempt succeeds. Provider nondeterminism may remain. The supplied GLM G1/G2 aggregates omit exact denominators, selection manifests, and task-level pairing, so they support only a descriptive replication claim. Historical cross-model comparisons use a different selected population and should not be read as a leaderboard. Exact prospective per-condition numerators and confidence intervals were not supplied for both DeepSeek populations, so the manuscript reports the provided rates and tests without reconstructing uncertainty estimates.

Correctness evidence is partial. Held-out reference-differential checking flags disagreements, not proven semantic errors. The deeper semantic audit covers only passing candidates that disagreed with a reference, and model reviewers are not formal proofs. Tests, references, properties, and natural-language specifications each leave gaps; no reported zero-disagreement result establishes zero semantic error.

9 CONCLUSION

A failed repair does not say what the model needs next. In two prospective failure populations, repeated sampling already recovers many tasks, while valid diagnostic information provides a large additional gain; non-diagnostic restructuring does not reproduce the historical improvement. The size of every effect remains conditional on the model and the selected failures.

Passing is likewise conditional on the evaluator. References, tests, and independent oracles can disagree, so successful repair remains an evidence-backed judgment rather than a single automatic fact. Repair evaluations should record the failure-selection rule, retry budget, supplied information, and evidence used to judge correctness.

AI USE STATEMENT

Generative AI tools assisted literature lookup, drafting, language editing, LaTeX development, and scientific-figure preparation. Language models also served as repair systems and semantic reviewers in the experiments described in the paper. The reported experimental measurements were provided by the authors and were not reconstructed during manuscript editing. The authors are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

Overstating automated repair reliability could encourage unsafe deployment of incorrect patches. We therefore distinguish evaluator passage from semantic correctness, report selection effects and unresolved ambiguity, and avoid production-readiness claims. Program-repair experiments should execute untrusted candidates in isolated environments without production credentials or systems.

REPRODUCIBILITY STATEMENT

The main text defines the task populations, selection rules, intervention conditions, retry budget, and outcome measures. The appendix records the supplied configurations, aggregate results, statistical comparisons, historical analyses, and semantic-audit boundaries. Full reproduction additionally requires the original prompts, mutation manifests, screening outputs, generated candidates, evaluator traces, statistical scripts, and differential-test records. No public artifact availability claim is made because release status was not provided for verification.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Mark Chen et al. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Dekun Dai, MingWei Liu, Anji Li, Jialun Cao, Yanlin Wang, Chong Wang, Xin Peng, and Zibin Zheng. FeedbackEval: A benchmark for evaluating large language models in feedback-driven code repair tasks, 2025. URL https://arxiv.org/abs/2504.06939.

Zaiyuan Di, Jianting Chen, Yunxiao Yang, Xiaoying Gao, Li Yang, Zhihao Wang, and Yang Xiang. Code Reffix: A benchmark for reflection-guided code repair with large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15541–15561, 2026. doi: 10.18653/v1/2026.findings-acl.762. URL https://aclanthology.org/2026.findings-acl.762/.

Ramtin Ehsani, Sakshi Pathak, Shriya Rawal, Abdullah Al Mujahid, Mia Mohammad Imran, and Preetha Chatterjee. Where do AI coding agents fail? an empirical study of failed agentic pull requests in GitHub. In *Proceedings of the International Conference on Mining Software Repositories*, 2026. URL https://arxiv.org/abs/2601.15195.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06770.

Cuong Chi Le, Aashish Yadavally, Minh Le-Anh, and Tien N. Nguyen. Benchmarking code improvement with progressive, adaptive, and interactive feedback, 2026. URL https://arxiv.org/abs/2607.01360.

Derrick Lin, James Koppel, Angela Chen, and Armando Solar-Lezama. QuixBugs: A multi-lingual program repair benchmark set based on the quixey challenge. In *Proceedings Companion of the 2017 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*, pp. 55–56, 2017. doi: 10.1145/3135932.3135941.

Oorja Majgaonkar, Zhiwei Fei, Xiang Li, Federica Sarro, and He Ye. Understanding code agent behaviour: An empirical study of success and failure trajectories, 2025. URL https://arxiv.org/abs/2511.00197.

You Wang, Michael Pradel, and Zhongxin Liu. Are "solved issues" in SWE-bench really solved correctly? an empirical study. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering*, 2026. doi: 10.1145/3744916.3764576. URL https://arxiv.org/abs/2503.15223.

Ratnadira Widyasari, Sheng Qin Sim, Camellia Lok, Haodi Qi, Jack Phan, Qijin Tay, Constance Tan, Fiona Wee, Jodie Ethelda Tan, Yuheng Yieh, Brian Goh, Ferdian Thung, Hong Jin Kang, Thong Hoang, David Lo, and Eng Lieh Ouh. BugsInPy: A database of existing bugs in python programs to enable controlled testing and debugging studies. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 1556–1560, 2020. doi: 10.1145/3368089.3417943.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

A Prospective Study Accounting

A.1 Configuration and selection rule

The primary prospective P1 and P2supp populations use SiliconFlow's `deepseek-ai/DeepSeek-V4-Flash` endpoint with temperature 0.3, top-p 1.0, maximum output length 8,192, and thinking disabled. The same provider/model configuration is used for the one-shot screen and all post-failure conditions within each population. The prospective GLM replication likewise runs through SiliconFlow, using `glm-5.3-flash` with temperature 0.3 and the same top-p, output-length, and thinking settings. These statements do not imply that either accessible endpoint is an identical historical checkpoint.

Table 8: Supplied SiliconFlow intervention configurations. The original 919-cell screening configuration remains incompletely documented.

Run	Model ID	Temp.	Top-p	Max output	Thinking
Historical DeepSeek	`deepseek-ai/DeepSeek-V4-Flash`	0.0	1.0	8,192	off
Prospective DeepSeek	`deepseek-ai/DeepSeek-V4-Flash`	0.3	1.0	8,192	off
Prospective GLM	`glm-5.3-flash`	0.3	1.0	8,192	off

Each task first receives one frozen BLIND/C0 attempt. Only an executable failure enters the intervention cohort. The screening output is never reused as one of the three recovery samples. After selection, BLIND, STRUCTURE, COARSE, LINE, and STRONG each receive three fresh calls from the same buggy state. P1 and P2supp are independently constructed populations and are never pooled.

Table 9: Prospective screens and executable-failure cohorts.

Population	Screened	Failure samples
P1	749	103
P2supp	1,377	296

A.2 Recovery rates and trend tests

Table 10: Task-level best-of-3 recovery rates. P1 rates use denominator 103; P2supp rates are recomputed with the stated cohort denominator of 296 rather than the inconsistent denominator of 300 implied by the originally supplied percentages. The GLM summary omits exact cohort denominators and selection records; confidence intervals are not reconstructed.

Population	BLIND	STRUCTURE	COARSE	LINE	STRONG
P1 ($n = 103$)	52.427%	44.660%	75.728%	77.670%	83.495%
P2supp ($n = 296$)	48.986%	40.541%	64.189%	79.392%	82.770%
GLM G1	53.2%	41.4%	75.3%	73.8%	80.1%
GLM G2	47.1%	39.8%	62.7%	77.9%	81.5%

For P1, the five rates correspond respectively to 54/103, 46/103, 78/103, 80/103, and 86/103 recovered tasks. The table reports these quotients to three decimal places; rounded to one decimal place, they are 52.4%, 44.7%, 75.7%, 77.7%, and 83.5%, respectively.

For P2supp, the five counts underlying the originally supplied percentages are 145, 120, 190, 235, and 245. Using the stated cohort denominator of 296 gives 48.986%, 40.541%, 64.189%, 79.392%, and 82.770%, respectively (49.0%, 40.5%, 64.2%, 79.4%, and 82.8% to one decimal place).

The full trend includes STRUCTURE even though its marginal ordering relative to BLIND reverses. Page's test is reported as supplied evidence for an overall ordered response; it is not used to claim that every adjacent contrast improves.

Table 11: Supplied prospective statistical tests. The informative sub-ladder is BLIND–COARSE–LINE–STRONG.

Population	Full Page's L	Informative sub-ladder	STRUCTURE vs BLIND
P1	$p = 0.00076$	$p = 0.0197$	$p = 0.344$
P2supp	$p = 4.8 \times 10^{-5}$	$p = 0.0020$	$p = 0.302$

A.3 Held-out reference-differential checks

Passing prospective candidates are exercised on newly generated inputs and compared with reference implementations. The supplied summary reports flagged disagreement/false-pass rates generally between 0% and 4% per condition. STRONG has zero flagged disagreements in both P1 and P2supp. Exact condition-level counts are not supplied and are not reconstructed. A zero differential flag rate is not a zero semantic-error rate.

B Historical Controlled Studies

B.1 Historical screen and initial failure forms

The primary historical screen contains 919 cells and 741 evaluator passes: MBPP Phase-2 contributes 167 cells and 140 passes, MBPP Phase-2B 602 and 469, and HumanEval 150 and 132. A separate QuixBugs secondary check passes 5/5. The supplied record does not establish the exact model/inference configuration for every original screening cell.

The 178 failures comprise 61 `IDENTICAL_ECHO`, 6 `TRIVIAL_EDIT`, 40 `MODERATE_EDIT`, 51 `REWRITE`, and 20 `EMPTY`. The 20 EMPTY cases are output-contract failures and all recover under ordinary best-of-3 resampling in the localization–test study. The main five-condition historical analysis therefore uses 158 non-EMPTY failures.

B.2 Localization and failing-test feedback

Table 12: Historical localization × failing-test study. Recovery is task-level best-of-3.

	All 178 failures		Non-EMPTY 158	
Condition	Pass	Rate	Pass	Rate
BLIND	51	28.7%	31	19.6%
Localization	139	78.1%	119	75.3%
Failing tests	128	71.9%	108	68.4%
Localization + tests	152	85.4%	132	83.5%

The later paired re-analysis reports a net 88-task gain for localization versus BLIND and a net 77-task gain for tests versus BLIND. Adding tests when localization is already present has a net 13-task gain ($p \approx 0.035$). Adding localization when tests are already present has a net 24-task gain ($p \approx 3.9 \times 10^{-5}$). The supplied summary supports substantial overlap and incremental value; it does not support a strong-synergy claim.

Table 13: GLM-5.3 Flash localization × failing-test study on all 178 historical failures. Rates are recomputed from the supplied pass counts; net gains are paired improvements minus impairments.

Condition	Pass	Rate	Paired net gain
BLIND	62	34.8%	—
Localization	126	70.8%	+64 vs. BLIND
Failing tests	131	73.6%	+69 vs. BLIND
Localization + tests	154	86.5%	+28 / +23

Table 14: Historical task-level best-of-3 recovery on the same 158 non-EMPTY failures.

	DeepSeek		Xing		GLM-5.3 Flash	
Condition	Pass	Rate	Pass	Rate	Pass	Rate
BLIND	30/158	19.0%	30/158	19.0%	30/158	19.0%
STRUCTURE	59/158	37.3%	46/158	29.1%	61/158	38.6%
COARSE	88/158	55.7%	56/158	35.4%	82/158	51.9%
LINE	102/158	64.6%	64/158	40.5%	107/158	67.7%
STRONG	121/158	76.6%	76/158	48.1%	128/158	81.0%

B.3 Historical five-condition and cross-model outcomes

Historical records call STRUCTURE "PLACEBO." DeepSeek, Xing, and GLM have identical aggregate BLIND rates but need not recover identical task sets. The DeepSeek and Xing historical intervention runs use `deepseek-ai/DeepSeek-V4-Flash` and `XingChenAGI/Xing4.0-29B` through SiliconFlow with temperature 0.0, top-p 1.0, maximum output length 8,192, and thinking disabled. The GLM historical intervention and prospective replication both use `glm-5.3-flash` through SiliconFlow at temperature 0.3 with the other listed settings unchanged.

Table 15: Historical candidate-output share labeled `IDENTICAL_ECHO`. These are output-form proportions, not task recovery rates.

Model	BLIND	Structure	Coarse	Line	Strong
DeepSeek	29.7%	20.7%	12.2%	9.3%	3.2%
Xing	51.9%	35.2%	46.8%	46.4%	35.7%
GLM-5.3 Flash	28.3%	21.4%	13.6%	11.2%	3.1%

B.4 Morphology as a recovery feature

Table 16: Predictive AUC with task attributes alone and after adding original failure morphology. Recovery labels differ across rows.

Model / target definition	Attributes only	+ Morphology
DeepSeek, localization/test recovery	0.556	0.701
DeepSeek, five-condition recovery	0.492	0.563
Xing, five-condition recovery	0.714	0.722
GLM-5.3 Flash, five-condition recovery	0.473	0.523

The recovery labels are not interchangeable. Morphology provides a larger increment in the first DeepSeek analysis, little increment for Xing, and a modest increment for GLM; it is therefore descriptive evidence of heterogeneous behavior, not a validated cross-model predictor.

B.5 Earlier 30-task boundary study

An earlier construction retains 30 of 32 proposed defects after readiness checks. DeepSeek recovers 30/30 at BLIND, STRUCTURE, COARSE, and STRONG, and 29/30 at LINE. Xing recovers 24/30, 24/30, 21/30, 19/30, and 25/30. DeepSeek's BLIND ceiling makes this set uninformative for diagnostic gains; it is retained only as evidence that construction and task difficulty change the observed response curve.

C Semantic Adjudication Details

Differential testing finds 9 reference-disagreement candidates among 175 deduplicated historical DeepSeek passes and 5 among 139 Xing passes. The 14 candidates cover seven function behaviors.

The supplied adjudication reports 127 probes in total, 91 with specification-derived executable oracles. Complete raw logs are not available for reconstructing the number and type of probes per candidate.

The automated-review summary reports 12 `VALID_ALTERNATIVE`, one `SPEC_AMBIGUOUS`, and one `CONFIRMED_INCORRECT`, but these aggregate labels require qualification:

`count_Squares`. At $(10, 20)$ the candidate returns 935, the reference 770, and independent brute-force enumeration 935. This supports a reference error on that disagreement input.

`snake_to_camel`. Disagreements include consecutive underscores and empty segments whose intended input-domain status is unclear.

`find_Points`. Across 8,281 enumerated inputs, the candidate violates audited properties more often under two interpretations. Because the term "point" is ambiguous and the reference also violates proposed properties, the defensible conclusion is property-supported incorrectness under the stated interpretations.

`decreasing_trend`. Test polarity conflicts with the behavior suggested by the name/specification, making this a specification–test inconsistency rather than an unambiguous candidate error.

Two model reviewers provide independent assessments and a third reviews disagreements. These reviews are triage evidence, not formal verification. Because candidates first enter the audit through reference disagreement, this subset cannot estimate semantic error over all passing repairs.

D Exploratory Real-Bug Background

An earlier exploratory track examines 501 BugsInPy records (Widyasari et al., 2020). Environment, dependency, and test constraints leave four executable cells, none of which pass. Three outputs are unchanged and one is a trivial edit. The sample is too small and selected to estimate real-project repair performance, so it is not used in the main argument.

E Documentation Required for Full Reproduction

The aggregate reports support the claims made in this manuscript, but full independent reproduction requires records not available during editing: exact task manifests for P1 and P2supp; per-condition task-level success indicators and counts; prompts and screening outputs; all generated candidates; evaluator and differential-test traces; statistical scripts; mutation policies and seeds; historical run identifiers; morphology rules; and the complete semantic-adjudication packets. Artifact release status was not supplied.