HITCHHIKING LABELS: WHEN REPAIR VALIDATION MISSES COLLATERAL DAMAGE

Anonymous authors
Paper under double-blind review

ABSTRACT

Targeted fine-tuning is often used to repair a narrow model failure, and is typically validated using data generated from the same specification as the repair data. We show that this practice can hide regressions in other fields of a structured output. When a repair specification fixes an input factor, another output field may receive the same label on every repair example—a hitchhiking label—while matched validation never tests the alternative factor values on which that field should change. In CLEVR-Fields, a Qwen3.5-9B repair improves such a field from 83.7% to 100% on repair-matched inputs while reducing accuracy on alternate inputs from 86.0% to 0%; the collapse repeats from the public base model across three seeds. We further find that whether a repair appears safe depends on how it is evaluated. Removing the affected field from the fine-tuning loss passes a forced-choice preservation check, yet its accuracy on alternate inputs is only 3.4–7.6% when the model generates the full structured output, with most errors coming from values outside the permitted schema. We introduce DoseLint, which uses logged repair factors, reference data, and output dependencies to request preservation tests on the missing input comparisons and in the model's intended output mode. In our approval study, given the affected field, 59-row tests on the omitted factor values detect 98.6% of its regressions, compared with 30.3% for tests sampled from the original data distribution. Repair validation should therefore account jointly for the inputs omitted by the repair specification and the way affected outputs are actually produced.

1 INTRODUCTION

Fine-tuned models are often repaired as specific failures are discovered in use. A successful repair should correct the target failure without degrading behavior the previous model handled correctly (Bansal et al., 2019; Srivastava et al., 2020). In targeted fine-tuning, both repair and validation examples may be generated from the same template, rendering pipeline, or teacher-in-the-loop system (He et al., 2023; Khan et al., 2025). This creates a simple blind spot: if the specification holds some input factor fixed, matched validation may never test the alternative values needed to detect a regression elsewhere in the output.

Consider a model that returns a JSON description of a scene. Suppose we repair its answer to a question using examples in which every relevant object happens to be near. The correct `depth` field is then also `near` on every repair example, even though depth is not the repair target. We call this repeated supervision of a non-target field a **hitchhiking label**. If validation is generated from the same specification, it again contains only near examples and cannot show whether the updated model still recognizes far objects. We call the affected non-target output the *sibling field*, and validation drawn from the repair specification *repair-matched validation*.

This blind spot is not specific to a particular train–validation split. Public held-out data can preserve the same factor concentration seen in training: in our audit, all 17 checkable template cases retain the training-dominant factor value in at least 97% of official test rows (Appendix E.1). This matters especially for structured extraction, from scene descriptors to receipt quantities (Park et al., 2019). A separate test split therefore need not supply the comparison that the repair specification removed.

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.

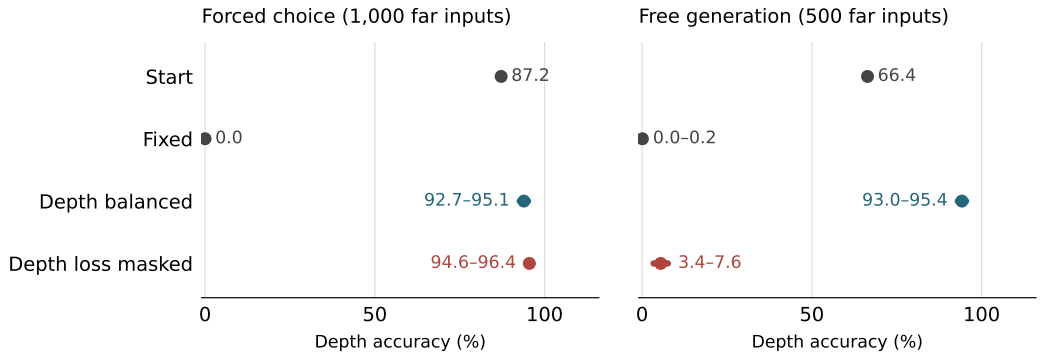


Figure 1: Depth-loss masking passes forced choice but fails when Qwen3.5-9B generates JSON. Values use the task-adapted start and three repair seeds (Tables 2, 3 and 5). Forced choice scores 1,000 far inputs; generation scores a fixed subset of 500. These are separate tests, not paired accuracies on the same rows.

At the same time, a hitchhiking label is not itself a prediction of harm: some repairs preserve the sibling field despite the same structural blind spot. The question is therefore not whether constant supervision always causes a regression, but what evidence is needed to establish preservation when repair-matched validation omits the relevant comparison.

Preservation also depends on how the affected output is evaluated. The clearest mismatch is between choosing an answer and generating one. A model may still prefer the correct value when evaluation supplies a valid prefix and restricts the answer to a small candidate set, yet fail when it must generate the complete structured response. We observe exactly this mismatch: a repair that passes a forced-choice preservation check produces mostly out-of-schema values for the affected field under free generation. Thus, checking the right inputs is not sufficient if the evaluation mode does not match how the model will actually produce its output.

We operationalize this view with DoseLint, a test-selection tool for corrective updates. Given logged repair factors, DoseLint identifies values that the repair fixes or strongly concentrates; reference data provide alternative values; and supplied output dependencies narrow the non-target fields that may need to change with those factors. When no dependency map is available, the protocol can conservatively consider all non-target fields. DoseLint turns these inputs into fieldwise preservation-test requests. The trained candidate is then evaluated on the requested inputs, including free generation when generation is the model's intended output mode. DoseLint is a command-line tool, following the idea of regression test selection in software engineering (Rothermel & Harrold, 1996; Yoo & Harman, 2012).

We evaluate this failure mode and its implications for repair approval across structured-output repairs. Our main findings are:

- **Repair-matched progress can hide severe collateral regression (Section 4.1).** On a balanced CLEVR-Fields test, a Qwen3.5-9B repair raises sibling-field accuracy from 83.7% to 100% on repair-matched inputs while reducing it from 86.0% to 0% on alternate inputs. The collapse also repeats from the public base model across three seeds.
- **A preservation verdict can change with the evaluation mode (Section 4.2).** Removing direct loss on the affected field passes the forced-choice criterion at all three seeds, yet under free generation its alternate-input accuracy is only 3.4–7.6%, largely because the model emits values outside the permitted schema.
- **Seeing alternate inputs is not the same as preserving them (Section 5).** Even repairs that contain alternate-factor examples can substantially degrade on those inputs, and increasing their count does not help consistently across seeds. Their presence in the repair data therefore cannot replace a separate preservation test.
- **Testing the omitted comparison changes repair approval (Section 6).** A target-only rule approves 22 of 23 candidates with the common target evaluation, including nine with sibling-field losses of at least ten points. Given the affected field, 59-row tests on omit-

ted factor values detect 98.6% of designated-field regressions, compared with 30.3% for samples from the parent distribution.

2 Related Work

Backward-compatible model updates. Updates that raise aggregate accuracy can introduce errors on inputs the previous model handled correctly and disrupt users' reliance on it (Bansal et al., 2019). Such incompatibilities arise even without data shift and can be overlooked by aggregate performance measures (Srivastava et al., 2020). Yan et al. (2021) measure negative flips and reduce them with focal distillation. Related work studies regression in NLP (Xie et al., 2021) and structured prediction, the setting closest to ours (Cai et al., 2022). FlipGuard addresses regression within the same task during preference alignment (Zhu et al., 2024). Schumann et al. (2024) reduce negative flips after data updates by interpolating old and new model weights. These studies assess compatibility on a given evaluation set; we ask which preservation comparisons may be missing when the same repair specification generates both training and validation data.

Shortcuts and augmentation. VQA-CP shifts answer priors between splits to expose shortcut reliance (Agrawal et al., 2018); we instead derive the missing test from an individual repair's factor assignments. Challenge sets, hidden-stratification analyses and group-robust training target what an average hides (McCoy et al., 2019; Oakden-Rayner et al., 2020; Sagawa et al., 2020). Joshi & He (2022) provide the closest empirical precedent: counterfactual augmentation can strengthen reliance on perturbed features while weakening reliance on other robust features; syntactic augmentation for NLI similarly improves one challenge subset while lowering another (Min et al., 2020). Our setting differs in that the missing comparison is induced by the repair specification itself: we derive the alternate factor values to test from the proposed repair rather than from a predefined challenge distribution.

Preservation after fine-tuning. Fine-tuning can erode earlier behaviour through catastrophic forgetting (Kirkpatrick et al., 2017). LoRA trades off learning and retention differently from full fine-tuning (Biderman et al., 2024). DFR finds that core features can remain in an ERM classifier even when spurious features dominate its predictions, and retrains the last layer on a small balanced set (Kirichenko et al., 2023). LP-FT first fits a linear head and then fine-tunes to reduce distortion of pre-trained features under distribution shift (Kumar et al., 2022). WiSE-FT and PAINT interpolate pre- and post-fine-tuning weights to retain existing performance (Wortsman et al., 2022; Ilharco et al., 2022). These methods address how to retain prior behavior once a preservation target is known. Our focus is earlier in the workflow: determining which non-target behavior must be tested before a corrective update can be approved.

Model editing and wider effects. Knowledge editing checks both the intended change and preservation of unrelated facts. RippleEdits also tests changes implied by the edited fact (Cohen et al., 2024). Narrow fine-tuning can alter unrelated behaviour, as shown by emergent misalignment after training on insecure code (Betley et al., 2026).

Behavioural testing. CheckList guides test design through a matrix of capabilities and test types that users instantiate for their task (Ribeiro et al., 2020). AdaTest supports interactive test generation (Ribeiro & Lundberg, 2022). Data validation and combinatorial coverage check dataset properties and combinations of feature values (Polyzotis et al., 2019; Lanus et al., 2021). DoseLint specifies a test from the proposed repair's factors and output dependencies. Existing testing tools can then score it. Our sampling comparison evaluates tests given the affected field; it does not compare automatic field discovery.

3 Setting and Evaluation

3.1 Repair-matched validation

A model maps an input x to a record $y = (y_1, \ldots, y_K)$. A repair targets field t and records input factors such as depth, written $F(x)$. It fixes F at f_0 when every repair row has $F(x) = f_0$. In a mixed repair, the *fixed-value share* is the fraction with that value. Reference data describe the task before repair and contain other values of F.

Table 1: A regression test needs inputs on which the previous behaviour matters. The rows describe overlapping problems, not mutually exclusive methods.

Problem	Inputs to compare before and after repair	Why the usual score may miss the loss
Shortcut reliance	Inputs where the shortcut fails	These inputs may be absent from an unchanged test split
Spurious correlation	Minority-group inputs	Aggregate accuracy dilutes their losses
Earlier-task forgetting	Inputs from the earlier task	Current-task tests omit them
Update regression	Inputs the previous model got right	Aggregate gain hides individual negative flips
Hitchhiking labels (DoseLint)	Alternate values of the repair's fixed factor	Repair-matched validation excludes those values

Definition 1. A field $k \neq t$ is a *sibling* of factor F when there exist task-valid inputs that differ in F while holding all other relevant recorded factors fixed, and for which the correct value of k changes. When the repair fixes F and gives this field the same correct label on every row, those labels are *hitchhiking labels*.

Definition 2. *Repair-matched validation* samples the repair specification. An *alternate-value test* uses valid task inputs with $F \neq f_0$ and changed correct values for the sibling field. Preservation compares the starting and updated models on these same inputs.

Remark. If the specification fixes $F = f_0$, matched validation has no support on $F \neq f_0$, by construction, whatever its size. This is a support limitation, analogous to the overlap requirement in causal inference (Rosenbaum & Rubin, 1983). For concentrated rather than fixed factors, alternate inputs can occur but receive little weight in an aggregate score.

3.2 Tasks and training

CLEVR-Fields is our structured-output adaptation of CLEVR (Johnson et al., 2017), in which the model produces an answer together with 28 auxiliary JSON fields. Our main repair targets the answer to `largest_sphere` questions. The repair specification fixes object depth to `near`, making `answer_depth` the primary sibling field; alternate tests use valid `far` inputs for which the correct depth value changes. A separate Qwen2.5-VL-3B study uses `answer_size` as the sibling field. We also evaluate a different structured-output domain using CORD-v2 receipts, where the repair targets the printed total and the sibling field is item quantity.

Qwen3.5-9B uses one LoRA epoch over 384 rows at learning rate 5e-5, rank 16 and alpha 32. We compare the public base with a task-adapted model. The latter is checkpoint 8 of an earlier training trajectory, selected to have room for target improvement while still reading depth. The public-base experiment keeps this repair pair and budget. Qwen2.5 uses 24 steps at four learning rates, and CORD uses 300 repair rows. Appendix A gives model coverage and recipes.

A discovery sweep identified 384 repair rows as the smallest tested budget producing complete alternate-depth collapse. We then fixed the start, repair material, and 384-row budget for the three-seed evaluation reported in Section 4. Collapse also persists at 1,536 and 3,072 rows.

The fixed repair trains on complete correct responses. Balanced repair changes the input mix at the same row budget. We also compare lower learning rates, replay, loss masking and KL regularisation towards the starting model. Depth masking removes loss only on depth-value tokens. Target-only supervision trains only the intended answer. KL can accompany either full-response training or target-only training, allowing us to separate direct supervision of the other fields from a constraint on their output distribution.

3.3 Evaluation and analysis

Forced-choice scoring supplies the correct JSON prefix up to the field value and selects the candidate value with the highest conditional log-probability. Free generation instead greedily decodes the complete structured response. Forced choice therefore measures whether the model ranks the correct field value above supplied alternatives under a valid prefix, whereas free generation measures

whether it actually produces a valid and correct value as part of the full response. Missing fields, incorrect values, and schema-invalid values count as errors.

The main forced-choice evaluation uses 64 repair-matched near validation rows, 300 target questions, and 1,000 alternate far inputs. A separate balanced test contains 300 near and 300 far inputs. Free-generation evaluation uses the target test and a fixed 500-row subset of the far challenge.

We measure three outcomes separately: improvement on the repair target, preservation on repair-matched inputs, and preservation on alternate inputs. For the masking analysis, a candidate passes the forced-choice preservation criterion if target accuracy improves by at least ten percentage points and far-depth accuracy declines by at most five points. Passing this criterion does not imply preservation under free generation, which is evaluated separately. Section 6 uses a distinct ten-point decline threshold to define substantial harm in the approval analysis and compares the starting and updated models on identical rows.

Core Qwen3.5 experiments are repeated over three optimization seeds while holding the starting model, repair material, and training budget fixed. Other repetitions and test sizes are reported with the corresponding results. Paired comparisons use exact McNemar tests. Each contrast uses the baseline predictions associated with that evaluation: alternate-depth accuracy is 87.5% in the discovery and imbalance studies and 87.2% in the later task-adapted evaluation. Appendix A documents this three-row baseline discrepancy.

A *pre-recorded* rule denotes a timestamped project record written before the specified outcome was inspected; it is not a third-party preregistration. Generation error analysis and the expanded tie-selection check are post-hoc analyses.

4 Finding 1: Matched Progress Can Hide Collateral Collapse

4.1 Matched and alternate inputs

On one balanced 600-row test, fixed repair raises near accuracy from 83.7% to 100% while lowering far accuracy from 86.0% to zero. The balanced candidate scores 91% on each half. Both halves are drawn from the same test, so the contrast directly shows improvement on the repair condition alongside a loss outside it.

We observe the same collapse from both the task-adapted start and the public base: fixed repair drives far-depth accuracy to zero at all three seeds (Table 2). It also improves the target and reaches perfect near-depth validation accuracy. Balancing preserves depth accuracy relative to the task-adapted start, but the public-base run still loses 8.8 points.

Table 2: Forced-choice accuracy (%). Target: 300 questions; near depth: 64 validation rows; far depth: 1,000 rows. Three-seed ranges use 20260915–20260917; public balancing uses 20260915.

Start / update	Seeds	Target	Near depth	Far depth
Task-adapted start	—	79.0	84.4	87.2
Task-adapted / fixed	3	92.7–97.0	100.0	**0.0**
Task-adapted / balanced	3	95.3–95.7	92.2	92.7–95.1
Public base	—	80.0	67.2	97.1
Public / fixed	3	92.3–94.0	100.0	**0.0**
Public / balanced	1	94.0	95.3	88.3

Appendix B gives the per-seed and paired results.

4.2 Generation changes the mitigation decision

One way to avoid reinforcing a constant non-target label is to remove its loss during repair. Both masking variants meet the forced-choice repair criterion at all three seeds, although depth masking costs 3.3–8.0 points on near inputs (Table 5).

Under free generation, however, the same masking variants fail to preserve the affected field. Depth masking leaves only 3.4–7.6% correct far-depth answers (Table 3), and target-only supervision also

falls below the start. Balanced repair improves the target and exceeds the start's generated depth accuracy at every seed.

Table 3: Free-generation accuracy (%) from the task-adapted start. Target: 300 questions; far depth: 500 inputs. Ranges span three repair runs; single values come from one evaluation.

Update	Target	Far depth
Start	80.3	66.4
Fixed	94.0–97.0	0.0–0.2
Balanced	94.3–96.0	93.0–95.4
Depth-loss masking	92.7–96.0	3.4–7.6
Target-only supervision	82.3–90.3	35.8–53.6
Target-only + KL	91.7	58.2

Where the evaluations diverge. The depth-masked responses are valid JSON with a depth field; the failure is in the field value. Most give `back` or `middle`, outside the permitted depth values (Table 4). Target answers remain accurate.

Table 4: Depth-masked generation on the same 500 far inputs per seed. Schema errors include any value other than `near` or `far`. Agreement compares generated depth with forced-choice predictions on identical rows.

Seed	Valid JSON	Out-of-schema depth	Generation / forced-choice agreement
20260915	100.0%	95.6%	4.4%
20260916	100.0%	92.4%	7.6%
20260917	100.0%	96.6%	3.4%

Forced choice supplies a correct prefix and restricts the output to two candidate values. Free generation imposes neither constraint. The model can therefore prefer the correct value within the supplied pair while generating an invalid value when unconstrained. On the same 500 inputs, only 3.4–7.6% of generated depth values agree with the forced-choice predictions (Table 4). These results show that producing an in-schema value and choosing that value from the image are distinct behaviors.

The start already makes schema errors. Fixed repair usually emits a permitted but incorrect `near` value; balanced repair usually gets depth right. Separating the effects of the prefix and the two-value restriction would require varying each independently. Appendix B gives the full error counts.

Masking removes depth's direct loss, but shared parameters still change. The preservation check must score the generated depth value, including values outside the schema; Table 3 counts these as errors.

4.3 Standard mitigations

The lowest learning rate, replay, and full-response KL reduce far-depth damage, but none fully preserves accuracy; at 3e-5 the loss remains 87.2 points (Table 5). The lowest learning rate also misses the ten-point target-gain criterion. Target-only supervision with non-target KL preserves forced-choice depth but loses generated depth accuracy (Table 3).

Table 5: Forced-choice changes from the task-adapted start (percentage points). Target: 300 questions; near depth: 300 mixed-test rows; far depth: 1,000 rows. Ranges span three repair runs; single values come from one run.

Update	Target change	Near change	Far change
LR 1e-5	+8.3	+14.3	-58.3
LR 3e-5	+15.0	+16.3	-87.2
Replay	+15.0	+15.0	-32.3
Full-response + KL	+14.7	+7.7	-15.2
Target-only + KL	+13.7	-2.7	+4.1
Depth-loss masking	+13.3 to +17.3	-8.0 to -3.3	+7.4 to +9.2
Target-only supervision	+14.7 to +17.3	-1.7 to +3.3	-2.1 to +2.7

Adding KL to full-response supervision reduces the far-depth loss but does not remove it. Restricting supervision to the target while retaining non-target KL avoids the forced-choice loss. Its generated depth remains below the start (Table 3), so the preferred mitigation changes with the evaluation mode.

4.4 When constant supervision preserves the field

Constant supervision does not always cause collapse. Table 6 compares the main alternative repair conditions. CORD retains quantity accuracy while improving receipt totals. Qwen2.5 retains size accuracy under some full-fine-tuning rates, but collapses at the highest rate.

Table 6: Collapse and preservation under constant labels. Accuracy: %; gains: points. Qwen2.5 and CORD use seed 3407; the screen-selected pair has one run. Tests contain 300 small-object inputs, 150 quantity inputs and 300 screen-selected alternate inputs, respectively. CORD's mixed repair uses 252 quantity-one and 48 other rows.

Setting	Update	Target gain	Before → fixed	Balanced / mixed
CLEVR screen-selected pair	600-row arms	—	86.3 → **0.0**	99.0
CLEVR size	Qwen2.5 LoRA 5e-5	+12.3	81.0 → **23.0**	100.0
CLEVR size	Qwen2.5 full 5e-5	+7.3	81.0 → **0.0**	97.0
CLEVR size	Qwen2.5 full 1e-5	+16.0	81.0 → 76.0	100.0
CLEVR size	Qwen2.5 full 3e-5	+12.3	81.0 → 83.0	100.0
CORD quantity	Qwen3.5 LoRA 5e-5	+23.3	96.0 → 94.0	97.3

Two Qwen2.5 updates gain the same 12.3 target points but leave size accuracy at 23% and 83% (Table 6). The full-fine-tuning margin at 3e-5 becomes negative at step 8 and recovers by step 24. Thus even similar final target gains can accompany different update paths and different losses elsewhere.

In additional pair tests, a screen-selected CLEVR pair collapses from 86.3% to zero, while three other trained pairs show no ten-point loss. One preserved pair also reports its fixed factor directly (Appendix C). Appendix F gives the CORD comparison.

5 Finding 2: Alternate-Input Coverage Does Not Imply Preservation

Adding alternate inputs to the repair data does not by itself preserve performance on them. With a 5% far share, a 384-row repair still reduces far-depth accuracy from 87.5% to 33.0% (Figure 2(a)). At this mixture ratio, a 59-row independent sample contains at least one far example with 95.2% probability, so observing an alternate case is much easier than preserving accuracy on the alternate slice.

We next vary both the number of alternate rows and their share in the repair data (Figure 2(b)). Reducing the near share from 95% to 80% improves far-depth accuracy at all three seeds. By contrast, increasing the number of alternate rows helps consistently at 95% near share, but not at 80%: the effect reverses at one seed and disappears at another (Table 7). The target is successfully repaired in all nine cells of the original grid, with target accuracy ranging from 90.3% to 97.0%.

Because total repair size changes with both count and mixture share, this experiment does not identify a count-only effect. It does show that the presence of alternate examples cannot substitute for a separate preservation test: every condition contains such examples, yet alternate-input accuracy varies widely.

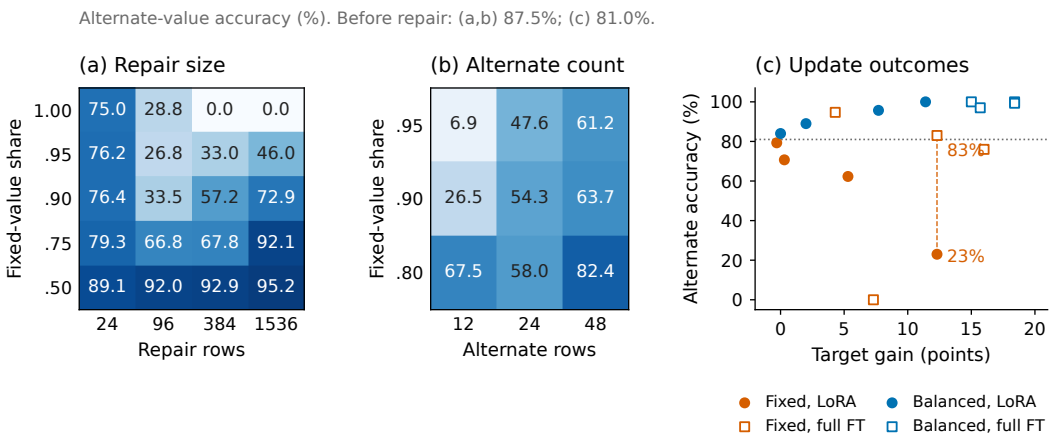


Figure 2: Alternate-input accuracy after repair. (a) Discovery-run accuracy by fixed-value share and repair size; the starting accuracy is 87.5%. (b) The first run of the count-by-share grid; Table 7 gives the repeated corner contrasts. (c) Qwen2.5-VL target gain versus alternate-size accuracy under LoRA and full fine-tuning; Table 6 gives the corresponding results.

Table 7: Far-depth differences (percentage points) on 1,000 inputs. Seed 3407 covers all nine cells; 20260912 and 20260913 repeat the four corners. Each contrast uses the same seed.

Contrast	3407	20260912	20260913
80% minus 95% near share; 12 far rows	+60.6	+72.7	+51.6
80% minus 95% near share; 48 far rows	+21.2	+13.2	+30.4
48 minus 12 far rows; 95% near share	+54.3	+43.7	+21.2
48 minus 12 far rows; 80% near share	+14.9	-15.8	0.0

6 Checking a Corrective Update

6.1 The cost of the omitted comparison

A target-only rule approves 22 of 23 candidates that have scores on the same 300-question target test. Nine approved candidates have depth losses of at least ten points. Target improvement alone therefore accepts updates that lose a previously correct output.

Would a test from the original task distribution catch these losses? We compare this parent-distribution test with a test containing only far inputs. Both score the same depth field, compare the candidate with its start on identical rows, and flag a decline of at least ten points with one-sided exact McNemar $p < .05$.

The collection contains all 31 completed candidates with the required depth scores, including 17 harmed candidates. Ten are labelled using the 1,000-row far test and seven using a 300-row subset. Both approval pools are scene-disjoint from those harm-definition rows. Each result averages 20,000 paired resamples within a candidate, then across candidates. These candidates share starts and repair material; Appendix D lists every one and the missing-score case.

Table 8: Paired depth tests at 59 rows. Detection uses 17 harmed candidates; false flags use 14 below the ten-point harm threshold. Both policies know the field and differ only in their input sampling.

Candidate group	Candidates	Parent test	Alternate test
Harm: 1,000-row evaluation	10	41.4%	98.0%
Harm: 300-row evaluation	7	14.5%	99.5%
Combined detection	17	**30.3%**	**98.6%**
Below threshold: false flags	14	5.2%	3.5%

Near-input gains can offset far-input losses in the parent average. More rows make that average more precise without necessarily revealing the loss on far inputs. Among partially damaged candidates,

parent detection remains at most 5% even at 600 rows. A false flag under the ten-point far-loss definition may still reflect a smaller far loss or a near-input regression. Table 8 measures input sampling after the field has been designated. Section 6.3 describes how the field is requested.

6.2 Depth balancing leaves another field damaged

An all-field forced-choice test finds a second constant label: whether the task's answer is tied. All fixed repair rows have the same tie label, which remains constant when depth alone is balanced. On 300 parent rows, tie accuracy falls from 100% to 31.7% after fixed repair and to 83.3% after balancing (Appendix D.1). The candidate that preserves depth still loses accuracy on tie cases.

The tie field also reveals a specific limitation of the earlier support screen. That screen relied on variation within the reference data and therefore missed cases in which the repair and reference each contained a different single value. Directly comparing repair and reference support would request the tie test in two earlier records. We treat this as an exploratory extension of the selection rule rather than as part of the original selector evaluation.

6.3 Deriving and checking the test

DoseLint requires factor assignments in a shared schema, reference support and labelled alternate inputs. Dependencies can come from the schema; without them, the candidate map includes all non-target fields. For each fixed or concentrated factor, it requests a paired test on reference-supported alternate values for every dependent non-target field. Repair filters can imply fixed values. Appendix D.3 gives the full selection and approval procedure.

The concentration threshold is fixed before scoring; $c = 1$ covers absent values and, for example, $c = .90$ also includes underrepresented ones. An alternate may relax the repair's filter while retaining task validity. If no labelled alternate can be built, approval remains unresolved. The paired rule in Section 6.1 uses a ten-point loss floor and $p < .05$; the all-field test adds Holm correction. Under this rule, PASS means no violation was detected at the available sample size. It does not guarantee preservation. Generated outputs require their own test.

A generator can log these inputs when it creates rows; Appendix D compares the resulting requests with a generic support audit. The 59-row experiment measures input selection for a known field.

For the depth repair, the request is concrete: test `answer_depth` on far inputs and compare it with the starting model. If the response will be generated, score generated depth as well. After balancing or masking, repeat the same comparison against the original start, not only against the failed fixed repair. The tie result shows why preservation cannot stop at the first affected field: preserving depth does not establish preservation of the tie field.

7 Scope and Limitations

Completed VLM repairs cover two Qwen model sizes, scene descriptions and receipt extraction (Appendix A.4). These experiments show that a missing comparison can hide large losses; they do not estimate its prevalence across models or repair tasks. Sampling compares 31 related candidates on designated depth, while the all-field test covers two candidates at one seed. Automatic selection requires complete factor records and labelled alternatives. The public-template survey, small-model diagnostics and single-output NLI case provide separate supporting evidence (Appendices E–G).

8 Conclusion

A repair specification can remove from validation the inputs needed to check another field. Neither adding alternate examples nor mitigation choices such as loss masking or balancing substitute for a preservation test. Corrective updates should be evaluated on the omitted inputs, across affected fields, and in the output mode in which the model will actually be used. When repair factors, reference support, and output dependencies are available, they can be used to derive these tests and make the missing evidence explicit before approval.

Reproducibility statement

Appendix A documents the experimental setup, training recipes, evaluation protocol, model selection, and known configuration limitations. Appendices B–G provide per-seed results, ablations, approval analyses, and supporting experiments. Public datasets and model identifiers are reported in the paper. Code and releasable experiment artifacts will be made publicly available upon publication; proprietary drawings and their renderer cannot be redistributed.

Ethics statement

The work involves no human subjects. Vendor drawings were used under agreement; no image, crop, vendor name or drawing identifier is included. Public datasets are used under their licences. Missing factors or tests can create false reassurance, so the protocol reports unavailable evidence as unresolved approval.

AI use statement

We used AI assistance for literature search, drafting and editing, experimental design, implementation and execution of experiments, analysis and pipeline code, figure preparation, interpretation of results, and feedback on the overall experimental workflow. We reviewed the text and code and verified reported values against available records. The authors take full responsibility for the final content.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't Just Assume; Look and Answer: Overcoming Priors for Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), pp. 4971–4980*, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Agrawal_Dont_Just_Assume_CVPR_2018_paper.html.

Gagan Bansal, Besmira Nushi, Ece Kamar, Daniel S. Weld, Walter S. Lasecki, and Eric Horvitz. Updates in Human-AI Teams: Understanding and Addressing the Performance/Compatibility Tradeoff. In *Proceedings of the AAAI Conference on Artificial Intelligence 33(01):2429–2437*, 2019. URL https://doi.org/10.1609/aaai.v33i01.33012429.

Jan Betley, Niels Warncke, Anna Sztyber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. Training large language models on narrow tasks can lead to broad misalignment. *Nature 649:584–589*, 2026. URL https://doi.org/10.1038/s41586-025-09937-5.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA Learns Less and Forgets Less. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2405.09673.

Deng Cai, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang. Measuring and Reducing Model Update Regression in Structured Prediction for NLP. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. URL https://papers.nips.cc/paper_files/paper/2022/hash/7af8e3dfefe6e3141144197b8fa44f79-Abstract-Conference.html.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the Ripple Effects of Knowledge Editing in Language Models. *Transactions of the Association for Computational Linguistics 12:283–298*, 2024. URL https://aclanthology.org/2024.tacl-1.16/.

Zexue He, Marco Tulio Ribeiro, and Fereshte Khani. Targeted Data Generation: Finding and Fixing Model Weaknesses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2023), pp. 8506–8520*, 2023. URL https://aclanthology.org/2023.acl-long.474/.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/bc6cddcd5d325e1c0f826066c1ad0215-Abstract.html.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), pp. 2901–2910*, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Johnson_CLEVR_A_Diagnostic_CVPR_2017_paper.html.

Nitish Joshi and He He. An Investigation of the (In)effectiveness of Counterfactually Augmented Data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2022), pp. 3668–3681*, 2022. URL https://aclanthology.org/2022.acl-long.256/. DOI 10.18653/v1/2022.acl-long.256.

Zaid Khan, Elias Stengel-Eskin, Jaemin Cho, and Mohit Bansal. DataEnvGym: Data Generation Agents in Teacher Environments with Student Feedback. In *The Thirteenth International Conference on Learning Representations (ICLR 2025). (arXiv:2410.06215)*, 2025. URL https://arxiv.org/abs/2410.06215.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training Is Sufficient for Robustness to Spurious Correlations. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023. URL https://arxiv.org/abs/2204.02937.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences 114(13):3521–3526*, 2017. URL https://doi.org/10.1073/pnas.1611835114.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations (ICLR 2022)*, 2022. URL https://arxiv.org/abs/2202.10054.

Erin Lanus, Laura Freeman, Richard Kuhn, and Raghu Kacker. Combinatorial Testing Metrics for Machine Learning. In *2021 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 12–16 April 2021, (Virtual) Porto de Galinhas, Brazil*, 2021. URL https://csrc.nist.gov/pubs/conference/2021/04/12/combinatorial-testing-metrics-for-machine-learning/final. DOI 10.1109/ICSTW52544.2021.00025.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), pp. 3428–3448*, 2019. URL https://aclanthology.org/P19-1334/. DOI 10.18653/v1/P19-1334.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. Syntactic Data Augmentation Increases Robustness to Inference Heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), pp. 2339–2352*, 2020. URL https://aclanthology.org/2020.acl-main.212/.

Luke Oakden-Rayner, Jared Dunnmon, Gustavo Carneiro, and Christopher Ré. Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging. In *Proceedings of the ACM Conference on Health, Inference, and Learning (ACM CHIL 2020), pp. 151–159*, 2020. URL https://doi.org/10.1145/3368555.3384468.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Document Intelligence Workshop at NeurIPS 2019*, 2019. URL https://openreview.net/forum?id=SJl3z659UH.

Neoklis Polyzotis, Martin Zinkevich, Sudip Roy, Eric Breck, and Steven Whang. Data Validation for Machine Learning. In *Proceedings of Machine Learning and Systems 1 (MLSys 2019)*, 2019. URL https://proceedings.mlsys.org/paper_files/paper/2019/hash/928f1160e52192e3e0017fb63ab65391-Abstract.html.

Marco Tulio Ribeiro and Scott Lundberg. Adaptive Testing and Debugging of NLP Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL 2022), pp. 3253–3267*, 2022. URL https://aclanthology.org/2022.acl-long.230/.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond Accuracy: Behavioral Testing of NLP Models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), pp. 4902–4912*, 2020. URL https://aclanthology.org/2020.acl-main.442/.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika 70(1):41–55*, 1983. URL https://doi.org/10.1093/biomet/70.1.41.

Gregg Rothermel and Mary Jean Harrold. Analyzing Regression Test Selection Techniques. *IEEE Transactions on Software Engineering 22(8):529–551*, 1996. URL https://digitalcommons.unl.edu/csearticles/13/.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally Robust Neural Networks. In *Eighth International Conference on Learning Representations (ICLR 2020)*, 2020. URL https://iclr.cc/virtual_2020/poster_ryxGuJrFvS.html. arXiv:1911.08731.

Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang. Backward Compatibility During Data Updates by Weight Interpolation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers) (EACL 2024), pp. 2846–2861*, 2024. URL https://aclanthology.org/2024.eacl-long.174/.

Megha Srivastava, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz. An Empirical Analysis of Backward Compatibility in Machine Learning Systems. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2020)*, 2020. URL https://doi.org/10.1145/3394486.3403379.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust Fine-Tuning of Zero-Shot Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022), pp. 7959–7971*, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wortsman_Robust_Fine-Tuning_of_Zero-Shot_Models_CVPR_2022_paper.html.

Yuqing Xie, Yi-An Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto. Regression Bugs Are In Your Model! Measuring, Reducing and Analyzing Regressions In NLP Model Updates. In *Proceedings of ACL-IJCNLP 2021 (Volume 1: Long Papers), pp. 6589–6602*, 2021. URL https://aclanthology.org/2021.acl-long.515/.

Sijie Yan, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi Deng, Meng Wang, Wei Xia, and Stefano Soatto. Positive-Congruent Training: Towards Regression-Free Model Updates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021), pp. 14299–14308*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yan_Positive-Congruent_Training_Towards_Regression-Free_Model_Updates_CVPR_2021_paper.html.

Shin Yoo and Mark Harman. Regression Testing Minimization, Selection and Prioritization: A Survey. *Software Testing, Verification and Reliability 22(2):67–120*, 2012. URL https://doi.org/10.1002/stvr.430.

Mingye Zhu, Yi Liu, Quan Wang, Junbo Guo, and Zhendong Mao. FlipGuard: Defending Preference Alignment against Update Regression with Constrained Optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), pp. 17333–17350*, 2024. URL https://aclanthology.org/2024.emnlp-main.960/.

APPENDIX A. EXPERIMENTAL PROTOCOL

A.1 TASKS, STARTS AND EVALUATION

CLEVR-Fields maps public CLEVR scene graphs to an answer and 28 additional JSON fields. The headline question is `largest_sphere`, and the other output evaluated is `answer_depth`. Fixed repairs contain only near inputs; balanced repairs split near and far at the same row budget. Tests and training rows are disjoint. The target test has 300 questions, near validation has 64 rows, the far challenge has 1,000 rows, and a separate mixed test has 300 rows per depth value. Generation uses the target and mixed tests and a fixed 500-row subset of the far challenge. A 300-row challenge subset labels harm for the older count/share candidates.

The public base is `Qwen/Qwen3.5-9B`, revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a`. The task-adapted start is checkpoint 8 of a base-to-corpus trajectory, screened for target headroom, readable alternate depth and non-constant predictions on a balanced test. The 384-row budget was selected as the smallest tested size with full collapse; subsequent seed repetitions hold the start, material and budget fixed. The public-base experiment repeats that pair and budget without the acquired adapter.

Forced choice supplies the correct assistant prefix up to the field value and scores candidate log-probabilities with the image. Depth candidates are `near` and `far`; the target uses eight colours. Generation is greedy over the whole JSON response. Parsing failures, truncation, missing fields and incorrect values count as errors. Values outside the field schema are incorrect even in parseable JSON.

A.2 RECIPES AND RECONSTRUCTION

The settings below come from freeze-writing code, adapter inspection and the CORD pre-run record. Complete executed configurations are still needed where the table identifies missing fields.

Table A.1. Documented settings and unresolved configuration fields.

Setting	CLEVR headline	CORD
Model	Qwen3.5-9B	Qwen3.5-9B public base
Fixed / balanced repair rows	384 / 384	300 / 300, mixed arm 252+48
Learning rate / epochs	5e-5 / 1	5e-5 / 1
Batch / accumulation	1 / 16	1 / 16
LoRA rank / alpha / dropout	16 / 32 / 0	Complete configuration not recovered
Target modules	Vision and language attention/MLP; exact regex not recovered	Exact list not recovered
Additional trainable module	Merger	Not independently verified
Maximum sequence length	4096	8192
Optimisation seed	20260915–20260917 where repeated	3407
Schedule / warmup / weight decay	Cosine / .03 / .01	Not independently verified
Optimizer / precision / attention	AdamW 8-bit / BF16 / SDPA	Not independently verified

The share package contains code, aggregate results, row IDs and server verification reports. Raw predictions and adapters are archived separately. Task-adapted adapters are private; public-base artifacts await rights clearance. Loading the acquired adapter on the server resolved its earlier absence, but did not recover the complete acquisition configuration. Proprietary drawings and their renderer are not redistributed.

A.3 ANALYSIS CHRONOLOGY AND BASELINE IDENTITY

Pre-recorded means documented in timestamped project records before inspecting the specified outcome, not third-party registration. Records identify arms, rows, criteria and failure branches. The initial count–share grid and the amendment adding two corner seeds precede their respective

outcomes. The new evaluation uses fixed success criteria and includes every completed candidate with the paired reads required for its approval analysis.

The generation error breakdown, expanded tie-selection check and small-model head refits are post-hoc analyses. The NLI all-seed decision rule and historical decision replay were also added after the original seed-1 verdict. These analyses do not retroactively change the original criteria.

The older 1,000-row depth challenge has 875 correct start predictions; the new task-adapted evaluation has 872. Every contrast uses its own measured start. The source of this three-row difference is not resolved by the available aggregates. The H100 check reused the older start predictions: it reproduces the fixed endpoint, not a fresh decoding of the 875-correct baseline. The separate 600-row mixed test has start accuracies 83.7% near and 86.0% far. These are different denominators, not alternative roundings of the challenge score.

A.4 Model eligibility

Eligibility checks target headroom, readable alternate output and non-constant predictions before repair. A failed screen gives no post-training outcome.

Table A.2. Model-loading and eligibility funnel. Failed or unavailable starts have no post-repair outcome.

candidate start	loaded under protocol	target headroom	readable off-support collateral	non-constant balanced stratum	outcome
Qwen3.5-9B, CLEVR headline pair	yes	79.7%, passes	87.5%, passes	passes	trained; Section 4
Qwen2.5-VL-3B, model-selected pair	yes	78.3%, passes	81.0%, passes	59.5%, passes	trained; Appendix C
InternVL3-8B	yes	87.3%, fails	94.1%, passes	92.7%, fails	no training
Pixtral-12B	yes	71.0%, passes	15.7%, fails	86.0%, fails	no training
SmolVLM-Instruct	yes	56.7%, passes	96.6%, passes	95.2%, fails	no training
Qwen2.5-VL-7B, drawing side token	yes	tie value 37.9%; installs	side token not fitted on its own training rows	–	target result only; no side result read
Gemma 3, Llama 3.2 Vision, Phi-3.5-vision, Molmo, Idefics2	no: access gate, remote code or download limit	–	–	–	not screened

InternVL3, Pixtral and SmolVLM already predict one depth value on most balanced-test rows. Qwen2.5-VL-7B does not fit the drawing side token on its training rows. The other listed models failed access or loading checks. These entries describe eligibility and availability, not absence of harm after repair.

Table A.3. Models, repair scale and repetitions. Screening measures a starting model's eligibility, not its response to training.

Model / task	Repair conditions	Repetitions
Qwen3.5-9B / CLEVR depth	384 rows; public and task-adapted starts	Three fixed repairs per start
Qwen3.5-9B / CLEVR mixtures	6–3,072 rows; separate nine-cell count–share study	One size sweep; four grid corners repeated at three seeds
Qwen2.5-VL-3B / CLEVR size	384 rows; LoRA and full fine-tuning; four rates	One seed per condition

Model / task	Repair conditions	Repetitions
Qwen3.5-9B / CORD quantity	300 rows; fixed and mixed repairs	One seed per condition
InternVL3-8B, Pixtral-12B, SmolVLM	Pre-repair screening only	No repair trained

APPENDIX B. CORE RESULTS AND GENERATION DIAGNOSTICS

B.1 PAIRED CORE EVALUATION

Table B.1 gives all new seed-level scores. Table B.2 reports paired depth losses and gains on the 1,000-row challenge. Public fixed repairs use three seeds; public balancing uses the first. Scores in this subsection use forced choice.

Table B.1. New core evaluation, accuracy (%). Column labels give test sizes; competence near and far use 300 rows each.

Candidate	Target (300)	Near target (64)	Near depth (64)	Near (300)	Far (300)	Far (1,000)
Task-adapted start	79.0	70.3	84.4	83.7	86.0	87.2
Public base	80.0	70.3	67.2	70.7	96.7	97.1
Task-adapted fixed 20260915	92.7	100.0	100.0	100.0	0.0	0.0
Task-adapted fixed 20260916	96.0	100.0	100.0	100.0	0.0	0.0
Task-adapted fixed 20260917	97.0	100.0	100.0	100.0	0.0	0.0
Task-adapted balanced 20260915	95.7	96.9	92.2	90.7	92.7	94.7
Task-adapted balanced 20260916	95.3	96.9	92.2	91.0	93.3	95.1
Task-adapted balanced 20260917	95.3	98.4	92.2	93.3	90.7	92.7
Public fixed 20260915	92.3	96.9	100.0	100.0	0.0	0.0
Public fixed 20260916	94.0	98.4	100.0	100.0	0.0	0.0
Public fixed 20260917	92.7	98.4	100.0	100.0	0.0	0.0
Public balanced 20260915	94.0	96.9	95.3	92.3	88.3	88.3

Table B.2. Paired alternate-depth comparisons with each start. Losses/gains count previously correct/newly correct rows. The p-value is the one-sided exact McNemar test for a decline.

Candidate	Change (points)	Rows	Losses	Gains	p (decline)
Task-adapted fixed 20260915	-87.2	1000	872	0	3.18e-263
Task-adapted fixed 20260916	-87.2	1000	872	0	3.18e-263
Task-adapted fixed 20260917	-87.2	1000	872	0	3.18e-263
Task-adapted balanced 20260915	+7.5	1000	7	82	1
Task-adapted balanced 20260916	+7.9	1000	12	91	1
Task-adapted balanced 20260917	+5.5	1000	29	84	1
Public fixed 20260915	-97.1	1000	971	0	5.01e-293
Public fixed 20260916	-97.1	1000	971	0	5.01e-293
Public fixed 20260917	-97.1	1000	971	0	5.01e-293
Public balanced 20260915	-8.8	1000	89	1	7.35e-26

The discovery run supplies the within-test example in the abstract. Its balanced candidate is distinct from the new seed-level candidates above.

Table B.3. Qwen3.5-9B discovery candidates: sibling accuracy (%) on one balanced depth test. The pooled column is the equally weighted mean of the two subgroup rates.

canonical candidate, conditional field probe	`near`, 300 rows	`far`, 300 rows	both values, 600 rows
task-adapted start	83.7%	86.0%	84.8%
fixed, 384 rows	100.0%	0.0%	50.0%
balanced, 384 rows	91.0%	91.0%	91.0%

B.2 Masking and regularisation

Depth masking zeros the loss on `answer_depth` value tokens. Target-only supervision retains loss only on the `answer` value tokens. Primary masking keeps the historical loss denominator; a separately labelled native-normalisation arm changes it. The forced-choice criterion requires target gain of at least ten points and far-depth loss of at most five points. Candidate harm in Appendix D instead uses a ten-point loss threshold. Thus the public balanced arm's 8.8-point loss is below the harm threshold but fails the preservation tolerance.

WiSE-FT was not evaluated. Its interpolation endpoints differed from the original start and fixed models by up to 0.117 and 0.248 nats in candidate log-probability, above the 0.05-nat gate, although labels agreed on all 20 check rows. This is an implementation check failure, not evidence of ineffective interpolation.

B.3 Free generation

All reported generation cells use greedy decoding without the gold prefix. The two 300-row depth columns share the mixed test; the 500-row column uses the far challenge subset. Public-base generation is not part of this comparison.

Table B.4. Free-generation accuracy (%) from the task-adapted start. All available rows parse as JSON; schema-invalid field values remain incorrect.

Candidate	Target (300)	Near (300)	Far (300)	Far (500)
Start	80.3	17.7	65.7	66.4
Fixed 20260915	94.0	100.0	0.0	0.0
Fixed 20260916	97.0	100.0	0.0	0.2
Fixed 20260917	95.7	100.0	0.0	0.0
Balanced 20260915	96.0	87.3	94.0	94.6
Balanced 20260916	95.0	85.7	95.3	95.4
Balanced 20260917	94.3	91.3	92.0	93.0
Depth-masked 20260915	92.7	0.3	1.3	4.4
Depth-masked 20260916	96.0	0.7	4.0	7.6
Depth-masked 20260917	94.7	0.3	2.0	3.4
Target-only 20260915	90.3	12.3	55.3	51.8
Target-only 20260916	82.3	19.0	35.7	35.8
Target-only 20260917	85.3	21.3	52.3	53.6
Target-only + KL 20260915	91.7	8.3	58.0	58.2

B.4 What fails after depth-loss masking

The delivered generation verification record tabulates emitted depth values and checks parsing, missing fields and truncation. Each masked seed produces 500 parseable responses on the far challenge, with neither truncation nor a missing required field. Only 22, 38 and 17 responses use an allowed depth value, respectively, and each of these values is `far`. All other values are outside the allowed set, including explicit JSON null. This gives out-of-schema rates of 95.6%, 92.4% and 96.6%.

Table B.5. Depth-value error breakdown on the same 500 far inputs. This is a post-hoc analysis of completed generation runs. Counts come from the delivered independent verification report; agreement with forced choice comes from the paired summary and its row-ID join.

Seed	Parseable	Missing / truncated	Correct `far`	`back`	`middle`	Other invalid	Out of schema
20260915	500	0 / 0	22	349	91	38	478 / 500
20260916	500	0 / 0	38	313	127	22	462 / 500
20260917	500	0 / 0	17	392	88	3	483 / 500

On these rows, generation agrees with forced-choice depth predictions on 22, 38 and 17 cases. The paired comparison therefore shows a mode disagreement on identical inputs, not merely a difference between the 500- and 1,000-row test pools. By comparison, balanced-generation agreement is 96.8%, 99.0% and 97.2%; fixed-generation agreement is 99.8%, 99.6% and 99.8%, despite its almost entirely wrong far answers.

The start already emits 145 `middle`, eight `back`, one `front` and one null among its 500 far answers; 345 answers use `near` or `far`. Fixed repair at seed 20260915 emits 499 `near` and one null; balanced repair emits 473 `far`, 26 `near` and one null. Repair can therefore improve output-schema adherence while destroying the input-dependent choice. Masking avoids that forced-choice collapse but does not preserve valid generation.

Forced choice changes two things at once: it supplies a correct prefix and restricts the available values. The records identify value-level failure, but do not identify which intervention explains the change. A causal decoding study would cross gold versus generated prefixes with restricted versus unrestricted values. No such crossed intervention is included here, and the small-model readout study is not a substitute for it.

Appendix C. Repair conditions and update paths

C.1 Size and mixture

All discovery results below use seed 3407 and the original 87.5% start on 1,000 far rows. Each fixed/balanced comparison holds repair size constant. Target accuracies differ by no more than five points between arms at every size.

Table C.1. Qwen3.5-9B discovery run (seed 3407): sibling accuracy (%) on the 1,000-row all-`far` test, by repair size. Pre-repair accuracy is 87.5%; balancing holds the row budget fixed.

repair rows	6	24	96	384	1,536	3,072
one-sided sibling accuracy	87.5	75.0	28.8	0.0	0.0	0.0
balanced sibling accuracy	87.5	89.1	92.0	92.9	95.2	95.8

Table C.2. Qwen3.5-9B discovery run: alternate-value accuracy (%) on 1,000 rows. The last column gives the number of independent mixture draws needed for 95% inclusion probability, not measured detection power.

fixed-value share	24 rows	96	384	1,536	rows for 95% counter-case visibility
1.00	75.0	28.8	0.0	0.0	no finite size
0.95	76.2	26.8	33.0	46.0	59
0.90	76.4	33.5	57.2	72.9	29
0.75	79.3	66.8	67.8	92.1	11
0.50	89.1	92.0	92.9	95.2	5

At 384 rows the discovery fixed repair gains 14.7 target points [9.9, 19.4], while balanced depth reaches 92.9%. These belong to the discovery run, not the new evaluation in Appendix B. Complete

fixed collapse persists at both larger sizes. At 95% near share, a 59-row independent mixture sample contains an alternate with probability 95.2%; this is inclusion probability, not measured test power. The 1,536-row 75%-share result (92.1% versus 95.2% under balancing) has no reported interval.

C.2 Separate count–share study

The main imbalance surface suggested that larger partially balanced repairs could do less damage. A separate prospective 3-by-3 lattice varied the number m of alternate-value rows and the fixed-value share. None of these nine cells was reused from the discovery surface. Pre-repair collateral accuracy is 87.5%; the target is repaired in every cell, 90.3–97.0%.

Table C.3. Seed-1 nine-cell study (3407): sibling accuracy (%) on 1,000 alternate-value rows. m is alternate-row count and N is total repair size.

fixed-value share	$m = 12$	$m = 24$	$m = 48$
.80	67.5% ($N = 60$)	58.0% ($N = 120$)	82.4% ($N = 240$)
.90	26.5% ($N = 120$)	54.3% ($N = 240$)	63.7% ($N = 480$)
.95	6.9% ($N = 240$)	47.6% ($N = 480$)	61.2% ($N = 960$)

At a fixed share, raising m from 12 to 48 changes collateral accuracy by +14.9, +37.2 and +54.3 points for share .80, .90 and .95. At fixed m, lowering the fixed-value share from .95 to .80 changes it by +60.6, +10.4 and +21.2 points. All six committed contrasts clear a five-point floor and Holm-corrected exact tests. The recorded seed-1 verdict was **both matter**: within this grid, neither alternate-row count nor fixed-value share alone determines the outcome.

This also separates visibility from dynamics. At a fixed-value share of .80, .90 and .95, matched validation needs 14, 29 and 59 rows respectively for 95% probability of seeing a counter-case, regardless of m. In the frozen seed-1 lattice, each fixed share has a 15–54-point behavioural range across counts; the added seeds below limit the generality of the share-.80 count contrast. Validation support and optimisation exposure are related design variables, not interchangeable explanations.

An amendment filed before training added two independent draws of the four corner cells. The table gives paired sibling-accuracy differences in points. Both skew contrasts keep the same sign at all seeds, as does the share-.95 count contrast; the share-.80 count contrast reverses at seed 2 and is flat at seed 3. The skew comparisons therefore carry the cross-seed result, while count is not a monotone general rule.

Table C.4. Four corner contrasts, in percentage points, across three seeds. Each column compares candidates at the same seed; total repair size varies with count and share.

corner contrast	seed 1 (3407)	seed 2 (20260912)	seed 3 (20260913)
count at a fixed-value share of .95: $m = 48 - 12$	+54.3	+43.7	+21.2
skew at $m = 12$: share .80 minus .95	+60.6	+72.7	+51.6
skew at $m = 48$: share .80 minus .95	+21.2	+13.2	+30.4
count at a fixed-value share of .80: $m = 48 - 12$	+14.9	**-15.8**	**0.0**

The amendment's predeclared rule assigns **not robust** when any added seed has a Holm-corrected reversal of at least five points. The -15.8-point seed-2 contrast meets that condition ($p = 6.7 \times 10^{-43}$). This does not revise the frozen nine-cell seed-1 verdict; it limits the generality of its count ordering. Because total repair rows and exposure change with m, these contrasts do not identify a count-only effect.

C.3 QWEN2.5-VL-3B

The model-specific pre-repair rule selects `largest_sphere` and `answer_size`. Start target accuracy is 78.3%; alternate-size accuracy is 81.0%; the modal prediction occupies 59.5% of a balanced test. Every sweep arm uses the same start, pair, 384 rows, seed 3407 and 24 steps. Fixed-repair cells below report target gain / alternate-size accuracy. Balanced cells report absolute target / alternate-size accuracy.

Table C.5. Fixed repairs, seed 3407: gain on the separate 300-row target test (percentage points) / sibling accuracy (%) on 300 `small` rows. Columns are learning rates; all updates use 384 repair rows and 24 steps.

recipe	1e-5	2e-5	3e-5	5e-5
LoRA	-0.3 / 79.3	+0.3 / 70.7	+5.3 / 62.3	+12.3 / 23.0
full fine-tuning	+16.0 / 76.0	+4.3 / 94.7	+12.3 / 83.0	+7.3 / 0.0

Table C.6. Balanced repairs under the same start, seed, rows and steps as Table C.5: accuracy (%) on the 300-row target test / sibling accuracy (%) on the separate 300 `small` rows.

recipe	1e-5	2e-5	3e-5	5e-5
LoRA	78.3 / 84.0	80.3 / 89.0	86.0 / 95.7	89.7 / 100.0
full fine-tuning	96.7 / 100.0	96.7 / 99.3	93.3 / 100.0	94.0 / 97.0

The fixed LoRA endpoints worsen monotonically with learning rate; full-fine-tuning endpoints do not. Both recipes exceed the recorded 30-point alternate-loss threshold only at 5e-5. The original criterion also checks a modal prediction share of at least 90% on the balanced test.

Mean gold-versus-fixed-value margins at steps 0, 8, 16 and 24 are +0.966, +0.647, +0.325 and +0.290 nats for fixed LoRA at 3e-5; +0.966, -4.029, -1.491 and +1.277 for fixed full fine-tuning at 3e-5; and +0.966, +2.829, +6.255 and +7.393 for balanced full fine-tuning. A negative intermediate mean margin can therefore recover. These means do not specify checkpoint accuracy or attribute the change to particular gradients.

C.4 SCREEN-SELECTED AND RANDOMLY DRAWN PAIRS

A pre-specified screening order selected the next candidate without using its post-repair outcome. A frozen order of 342 template candidates from CLEVR, PlotQA and GeomVerse was screened on the untrained start alone: a candidate was admitted when it was weak on a paired stratum carrying both factor values and rarely wrote the repair value there. The screen read all 342 and admitted 22: 19 PlotQA, 2 GeomVerse and 1 CLEVR. An admitted candidate proceeds to fixed and balanced arms only if a 96-row balanced micro-repair raises its target by at least 20 points.

Table C.7. Pre-specified screen: candidate eligibility, feasibility and target improvement. Only a candidate meeting the +20-point target threshold proceeds to the full fixed/balanced comparison.

admitted candidate	start	installability	fixed repair, sibling accuracy	balanced repair, sibling accuracy
Screen-selected CLEVR pair	86.3% on the challenge stratum	+26.8 points, installs	**0.0%**, repair value on 100.0%	**99.0%**, repair value on 1.0%
PlotQA templates	18.7–59.3% on scored paired strata	17 scored: all gains below +20; 2 builds infeasible	not run	not run
GeomVerse `area_right`	–	micro-repair cannot be built	not run	not run

admitted candidate	start	installability	fixed repair, sibling accuracy	balanced repair, sibling accuracy
GeomVerse `degree_angle`	42.8% total	+3.3 points, does not install	not run	not run

For the screen-selected CLEVR pair, the full fixed and balanced repairs each contain 600 rows and are scored on the same 300-row alternate-value challenge. The +26.8-point qualification gain is measured separately on the paired stratum after the 96-row balanced micro-repair (54.7% to 81.5%); it is not a full-arm target gain. The table separates the qualification gain from the full-arm endpoint.

Only the screen-selected CLEVR pair met the target-improvement threshold and proceeded to the fixed and balanced arms. The 19 PlotQA candidates comprise 17 scored micro-repairs, whose gains range from +0.7 to +9.2 points, and two infeasible builds (`CD3` and `M0`). These counts follow the named entries in the run record. Neither GeomVerse candidate proceeded to full repairs. Thus 21 of 22 admitted candidates have no fixed-versus-balanced harm comparison. The screen tests a pre-specified selection procedure within these datasets; the selected pair still shares the CLEVR generator family with the size study.

The available candidate population contained seven entries, all drawn in randomized order. Six were built and screened; three were eligible and trained. A pre-repair switch rule selected the task-adapted checkpoint-8 start. The three trained candidates installed their targets and had alternate changes of -1.0, +0.3 and -4.3 points. None crossed the ten-point harm threshold.

This is 0/3 among eligible trained candidates, not a simple random sample of three independent repairs. Two entries exchange the closely related sphere-count and same-shape-count fields. The descriptive Clopper–Pearson interval for 0/3 is [0, .708]; shared construction further limits a prevalence interpretation. The results provide additional cases of preservation under constant supervision, not an estimate that collapse is rare.

Table C.8. Randomly drawn pairs trained from the task-adapted start, one seed. Differences are percentage points on each pair's 300-row tests.

Draw order	Target	Fixed factor	Scored sibling	Target gain	Alternate change
1	largest_sphere	n_spheres	n_same_shape	+16.0	-1.0
2	largest_sphere	n_same_shape	n_spheres	+13.7	+0.3
5	unique_shape	answer_size	answer_size	+22.7	-4.3

APPENDIX D. APPROVAL TESTS AND SELECTION INTERFACE

D.1 PAIRED SAMPLING POLICIES

The designated-field analysis includes 31 candidates. Seventeen have an alternate-depth decline of at least ten percentage points relative to their corresponding start; fourteen fall below that threshold. Ten harmful candidates are labelled on challenge1000 and seven on a 300-row subset of that challenge. Harm is defined by effect size on that evaluation, while the small approval test additionally requires one-sided exact McNemar $p < .05$.

For each candidate and budget, the analysis uses 20,000 paired resamples with shared pre/post row indices. Both policies score the designated depth field. Alternate rows come from the far half of competence600, which is scene-disjoint from challenge1000 and its 300-row subset; parent-pool rows are also disjoint. Rates are calculated within each candidate and then averaged. At 59 rows, alternate sampling detects 98.6% of the 17 damaged candidates; parent sampling detects 30.3%. Among 14 below the harm threshold, the corresponding false-flag rates are 3.5% and 5.2%. A false flag under that definition can still accompany a smaller depth loss or a regression on another field.

The target-only rule is a separate comparator, not a 59-row test. Among the 23 candidates with target300 scores, 22 gain at least ten target points and are approved; nine of those approved candidates have alternate-depth harm. The remaining eight candidates use a different target evaluation and do not enter that homogeneous target300 denominator.

For candidates with 10–50-point losses, parent sampling detects at most 5.0% at 59 rows and none at 600 rows in these estimates. Large near-input gains can offset the conditional loss. This calculation holds the field fixed and varies the sampling policy.

The parent all-field comparison uses the first-seed fixed and balanced candidates and 300 parent rows. It tests 28 non-target fields with Holm correction and rejects when a corrected significant decline also reaches ten points. These candidates differ from the 31-candidate designated-depth collection. On those parent rows, tie accuracy changes from 100% to 31.7% after fixed repair and 83.3% after depth balancing. At a 59-row budget, all-field rejection probabilities are 100% and 54.7%, respectively.

At equal labelled-field budgets, each designated-field test uses one field value per row; the all-field test uses 28. A budget B therefore gives the all-field policy floor(B/28) rows. A cell with fewer than one row or insufficient pool supply is marked infeasible rather than omitted. This accounting measures a different budget from equal rows.

Table D.1. Designated-depth policies across row budgets. Detection averages over 17 harmful candidates; false flags over 14 below threshold. The alternate pool contains 300 rows.

Policy	Rows	Detection (%)	False flags (%)
Parent	59	30.3	5.2
Alternate	59	98.6	3.5
Parent	150	48.2	4.0
Alternate	150	99.7	1.5
Parent	300	52.2	3.4
Alternate	300	100.0	0.0
Parent	600	55.4	2.9
Alternate	600	infeasible	infeasible

D.2 Complete candidate inventory

The inclusion rule takes every completed candidate with the required paired depth reads. Table D.2 lists the 31 candidates. The 300-row harm set is a subset of the 1,000-row challenge. One public-base target-only candidate lacked the required reads. Repeated seeds and count/share runs share starts or repair material, so the 31 rows are related candidates.

Table D.2. Complete Qwen3.5-9B candidate inventory. The candidate column gives the update family; change is alternate-depth accuracy in points relative to each candidate's own start. Harm means a decline of at least ten points. Adapted denotes the task-adapted checkpoint-8 start.

#	candidate	start	seed	harm rows	depth change	harm
1	Count/share, 12 alternate, 80% fixed	adapted	3407	300	-20.7	yes
2	Count/share, 12 alternate, 90% fixed	adapted	3407	300	-63.7	yes
3	Count/share, 12 alternate, 95% fixed	adapted	3407	300	-81.3	yes
4	Count/share, 24 alternate, 80% fixed	adapted	3407	300	-30.0	yes
5	Count/share, 24 alternate, 90% fixed	adapted	3407	300	-33.0	yes
6	Count/share, 24 alternate, 95% fixed	adapted	3407	300	-40.0	yes
7	Count/share, 48 alternate, 80% fixed	adapted	3407	300	-6.3	no
8	Count/share, 48 alternate, 90% fixed	adapted	3407	300	-23.0	yes
9	Balanced 384	adapted	20260915	1,000	7.5	no
10	Fixed 384	adapted	20260915	1,000	-87.2	yes
11	Balanced 384	adapted	20260916	1,000	7.9	no

#	candidate	start	seed	harm rows	depth change	harm
12	Fixed 384	adapted	20260916	1,000	-87.2	yes
13	Balanced 384	adapted	20260917	1,000	5.5	no
14	Fixed 384	adapted	20260917	1,000	-87.2	yes
15	Depth-loss masking	adapted	20260915	1,000	8.5	no
16	Depth-loss masking	adapted	20260916	1,000	9.2	no
17	Depth-loss masking	adapted	20260917	1,000	7.4	no
18	Target-only (native)	adapted	20260915	1,000	3.6	no
19	Target-only	adapted	20260915	1,000	2.7	no
20	Target-only	adapted	20260916	1,000	-2.1	no
21	Target-only	adapted	20260917	1,000	2.3	no
22	Balanced 384	public base	20260915	1,000	-8.8	no
23	Fixed 384	public base	20260915	1,000	-97.1	yes
24	Fixed 384	public base	20260916	1,000	-97.1	yes
25	Fixed 384	public base	20260917	1,000	-97.1	yes
26	Depth-loss masking	public base	20260915	1,000	2.6	no
27	Fixed 384, LR 1e-5	adapted	20260915	1,000	-58.3	yes
28	Fixed 384, LR 3e-5	adapted	20260915	1,000	-87.2	yes
29	Replay	adapted	20260915	1,000	-32.3	yes
30	Full-response CE + KL	adapted	20260915	1,000	-15.2	yes
31	Target CE + KL	adapted	20260915	1,000	4.1	no

D.3 Expanded tie request

Algorithm 1. Test selection and approval.

```
Inputs: repair R, reference P, factors F, target t,
        dependencies D, concentration c, budget n,
        gain/loss thresholds, test rule.
For each factor, including filter-implied ones:
  Let f0 be its dominant repair value.
  Continue only if share_R(f0) >= c.
  For each v in support_P(F) other than f0:
    For each k in D(F), k != t:
      Seek n valid rows with F=v and changed gold for k.
      Record proven invariance; block unresolved requests.
      Otherwise score k before and after repair.
Pass if target gain and all requested checks pass.
After mitigation, repeat against the original start.
```

Depth-test scope. The recorded map names `answer_depth`, `answer_occluded`, `answer_size_rank` and `nearest_relation` as depth-dependent outputs. The 31-candidate comparison scores only `answer_depth`; it cannot measure the tie-field loss.

What the older complete-field records show. Retained 24-row Discover records for `largest_sphere` and `leftmost` include a `tie` factor. In each, the repair share of `True` is 1.0, its reference share is 0.0, and the reference contains one distinct value. Both records set `flagged=false`. The delivered replay of the historical rule requires repair share at least .97 and reference minority at least .05, where minority is measured relative to the reference's own most frequent value. Its value is zero for a singleton reference, even when that singleton differs from the repair value. Missing tie is therefore not explained solely by an omitted factor or dependency edge in these records.

An exploratory arithmetic replay reproduces both recorded non-flags. Replacing reference-internal diversity with direct comparison against the repair value admits a tie-test candidate in both summaries. This check uses retained support summaries from the older 24-row cases; it is not a run of

the historical compiler on complete current fixed384 records and does not establish a built or scored tie challenge.

Filter-implied factors. Algorithm 1 includes fixed values introduced by repair filters. Test construction may relax the fixed repair condition while preserving the task's semantic validity. If the task definition makes an alternate invalid, or no gold-changing labelled alternate can be built, the request is explicitly unresolved. It is not silently discarded to obtain PASS. Historical screens that excluded filter-entailed alternatives are retained as historical results in D.4, not relabelled as outputs of this extension.

Concentrated rather than singleton support. Algorithm 1 accepts a concentration parameter fixed before outcome inspection, such as c = .90. The old .97 screen would not request a factor with .95 share solely from that share. The extension covers such concentrations, but no new end-to-end detection rate is claimed for it. The support observation in Section 3.1 concerns absent support, not a guarantee for c < 1.

Dependencies. The protocol accepts schema-declared dependencies and uses an all-to-all candidate map when none is supplied. A schema proof that a field is invariant can discharge a candidate without a test; unresolved relevance cannot. Re-rendering one factor at a time and recomputing deterministic gold labels is a possible way to construct dependencies. It has not been verified as an automatic dependency extractor for the reported system.

Implementation status. The local check runs the legacy and expanded support predicates on saved summaries. The package lacks the complete current factor rows and executable compiler needed for a full rerun. Algorithm 1 describes the expanded protocol; main Tables 8–9 report the designated-field and all-field evaluations, respectively.

Table D.3. Exploratory support-summary check, not a compiler rerun. Both old repair sets contain 24 rows. The extended predicate uses repair share at least .90 and any reference mass outside the repair value.

Historical case	Repair tie=True share	Reference tie=True share	Reference distinct values	Legacy flag	Expanded candidate
largest_sphere	1.0	0.0	1	no	yes
leftmost	1.0	0.0	1	no	yes

D.4 Interface sensitivity and requested tests

The guarantee is relative to the factor schema and relevance relation supplied to Discover. On the same CORD-v2 repair rows, changing only that interface produces the following output.

Table D.4. CORD-v2 interface ablation with repair rows held fixed. Challenge size counts requested rows, not observed harmful updates.

run	factor in schema	relevance	audit action	challenge demanded
full interface	yes	output	**challenge**	150
factor removed	no	absent	none	0
relevance edge removed	yes	nuisance	pass	0
interface restored	yes	output	**challenge**	150

The two omissions fail differently under a sparse supplied relation. Without the factor, Discover cannot see the asymmetry. Without its relevance edge, it reports singleton support but treats the factor as a nuisance and issues no challenge. This is a predictable false negative; because the authors supplied these edges, the ablation does not independently validate semantic relevance. Replacing the sparse relation with all-to-all relevance removes that logical dependence while increasing test burden. All-to-all expands candidate pairs rather than asserting that every pair is semantically related: only a labelled alternate that changes the named output discharges an obligation, and an unresolved candidate cannot support **PASS**. Across five configurations the fallback adds two strata in one and none in four, so relevance acts mainly as test triage in these small-output cases.

The following comparison uses only the fixed and balanced repair/reference supports and declared schema; no downstream harm outcome enters the counts. A challenge is unbuildable when the requested alternate stratum violates the repair filter or has no labelled feasible construction.

Table D.5. Static audit comparison: requested test groups on the fixed and balanced candidates. No downstream model-outcome comparison enters these counts.

audit	asked, fixed	asked, balanced	unbuildable	includes later-collapsed coordinate	approval verdict
no audit	0	0	0	no	none
generic support/diversity	8	7	1	yes	none
challenge-all, screens off	6	3	1	yes	none
DoseLint	**5**	**2**	**0**	**yes**	**BLOCK / PASS**

This is illustrative cost accounting and an actionability check, not a general efficiency benchmark or harm-prediction result. Generic support checking also contains the relevant coordinate, but asks for nuisance or infeasible strata; DoseLint uses the declared relevance relation to narrow the tests and convert them into a merge decision. If relevance is replaced by the all-to-all relation, the guarantee remains but the candidate set approaches generic support checking.

The same compiler was run on 33 further interventions composed from public chart-question material. For each intervention, the exposed factor supports, flagged factor–output pairs, required challenge strata and structural verdict were committed before any downstream model-outcome comparison. The implementation and output schema were unchanged across all 33, with 22 PASS exits and 11 BLOCK exits.

The equal-budget generic support audit and DoseLint agree on the structural verdict in 30 of 33 cases but name disjoint factor sets in 11. Eight of DoseLint's eleven BLOCKs are supply artifacts: the minority rows needed for a challenge or balance cannot be built. These are structural results, not evidence that one audit detects more unsafe fine-tuning outcomes than the other. No behavioural endpoint comparison is reported here.

The interface has been instantiated in drawing VLM, CLEVR and NLI settings. The separate registry contains 1,062 templates from 23 dataset configurations. The registry measures data concentration; model behaviour is evaluated only in the repair experiments.

These static outputs are not the 31-candidate outcome experiment. The drawing comparison uses the old filter-compatibility rule; the expanded protocol can request filter-implied alternatives. The older decision replay is also distinct from the trained replay baseline: it applies post-hoc approval policies to existing candidates, assumes some matched approvals and changes the final candidate across policies. We do not use it as a detection-rate estimate. Its BERT comparator lesson is reported in Appendix F. Under a depth-only request, a balanced candidate can pass while its tie field still regresses.

APPENDIX E. PUBLIC TEMPLATES AND INTEGRATION

E.1 TEMPLATE AND ARTIFACT SCREENS

Template registry. The machine-readable registry has 1,062 eligible templates with at least 20 training rows across 23 public dataset configurations. We merge GQA's answer-polarity-coded template identifiers before counting, so answer polarity is not supplied by the grouping key. A factor qualifies when one value occupies at least 97% of a template's training rows and its dataset-wide training minority share is at least 5%. Factors include exact answer values and numerical-answer bins. This yields 25 fixed or near-fixed answer-factor templates across ten configurations. Of these, 17 template cases have checkable official held-out data and all 17 retain at least 97% concentration on the training-dominant factor value; eight cases remain unknown under the registry's checkability rule.

Table E.1. Public template registry. A template needs at least 20 training rows; concentration is at least 97%. The held-out denominator contains only checkable template cases.

configuration with detected factor	eligible templates	fixed/near-fixed templates	retained in checkable held-out cases	unknown
CLEVR v1.0	90	1	1/1	0
Super-CLEVR	90	1	—	1
CLEVR-Math, general / multihop	19	3	2/2	1
CLEVR-CoGenT	90	1	1/1	0
GQA balanced, polarity IDs merged	71	4	2/2	2
PlotQA / GeomVerse	93	2	2/2	0
TDIUC, generated rows	8	1	1/1	0
DVQA	406	12	8/8	4
other 13 configurations	195	0	—	0
total	**1,062**	**25**	**17/17**	**8**

Seventeen counts template cases, not independent official splits. A nearly fixed numerical bin need not be a nearly fixed exact answer; near-fixed templates do not satisfy the exact-support condition in Section 3.1. Eight unknown cases are excluded from the 17-case denominator, not treated as preservation successes. The registry summarises public annotation files; its 23 configurations include related variants rather than independent generator families. It establishes inherited concentration in inspected official held-out material, not that these datasets emit corrective repair batches or that their pipelines approve updates on matched-only scores.

Pipeline artifact screen. A separate prospectively committed convenience screen inspected 48 public corrective pipeline artifacts. None released the full precision combination of row-level factor assignments, reference support, factor–output relevance and constructible labelled alternate strata. The screen was single-coded and is not a representative prevalence estimate. It measures recoverability of the audit interface, whereas the template registry measures answer-factor concentration and held-out persistence. The two denominators must not be pooled.

E.2 Recoverable inputs

The "full precision interface" makes four inputs recoverable without model-based latent-factor inference: row-level factor values or a deterministic derivation, a reference population or support table, a mapping from factors to affected output coordinates, and a constructible labelled alternate stratum. Conservative execution needs the first, second and fourth; it may replace the mapping with all-to-all relevance at greater test cost. Documentation alone does not count unless it determines row assignments. An ambiguous case is recorded as unavailable rather than absent. Because there was no independent second coder, the screen supports an engineering-gap observation, not a literature-wide frequency claim.

One external pipeline was taken through the feasibility protocol before the larger artifact screen. Its released GQA row contains an identifier, question, answer and image; the teacher's description and inferred weak skill survive only in a side log that is not keyed to the emitted row. The public challenge is constructible and the released workflow can conditionally run with its hosted teacher, but row-level factor provenance cannot be joined to repair and reference rows. The protocol therefore stopped before training rather than reconstructing factors after seeing outcomes.

Across the later 48-artifact convenience screen, the most common failure was not that generators lacked internal concepts. It was that the emitted repair rows and the reference corpus lacked a shared factor field, or that the only preserved factor was the target itself and no collateral output coordinate existed. The intended integration point is intervention construction, before those assignments are discarded; the screen measures retrofit availability of the full precision interface, not feasibility of native deployment. Native systems should persist assignments at row creation, define reference support in the same schema, and optionally declare affected outputs to reduce test burden. The screen does not estimate how often deployed systems suffer behavioural harm.

APPENDIX F. SUPPORTING TASKS AND CONTROLS

F.1 CORD RECEIPTS

CORD-v2 is an independently authored structured-output dataset of receipt photographs. The corrective target is the printed grand total; the collateral coordinate is the printed quantity of a named line. The fixed repair holds quantity at one. Among 707 eligible parent receipts, the factor takes 25 values and 206 rows, 29.1%, are off the repair value. Before any model output was scored, Discover emitted a 150-row non-one challenge; the balanced specification resolved the flag.

Table F.1. CORD-v2, Qwen3.5-9B, seed 3407: quantity/total accuracy (%) on matched and alternate rows, and percentage emitting quantity one. The mixed arm contains 252 quantity-one and 48 non-one rows, not a 50:50 split.

arm	matched quantity, $n = 120$	challenge quantity, $n = 150$	matched total	challenge total	writes repair value on challenge
pre-repair	100.0 [96.9, 100]	96.0 [91.5, 98.2]	76.7	52.7	2.0
fixed, 300 rows	100.0 [96.9, 100]	94.0 [89.0, 96.8]	100.0	98.0	3.3
balanced, 252+48	100.0 [96.9, 100]	97.3 [93.3, 99.0]	98.3	96.0	2.7

The repair installs: matched total rises by 23.3 points [15.8, 30.9], exact $p = 7.5 \times 10^{-9}$. The matched sibling score is 100% but every matched row has quantity one, so reader and constant remain observationally equivalent. Behavioural harm does not materialise: challenge quantity moves from 96.0% to 94.0%, a paired -2.0 points [-4.9, +.9], exact $p = .375$, below the five-point reporting floor. A separate 29-row validation challenge agrees: 96.6% before and 100.0% after the fixed repair.

This is an informative negative with all three preconditions intact: the update is effective, the collateral coordinate is readable, and the approval support is maximally blind. It confirms that structural blindness is not a harm predictor. The balanced candidate has similar target accuracy; this case has no collapse from which to measure recovery.

F.2 NLI: RESCUE AND PRESERVATION

NLI has one output, so this case falls outside the sibling-field definition. Min et al. (2020) already reported the HANS trade-off and interpreted it as reduced lexical-overlap reliance. Our comparison asks whether rescue relative to an augmented model also preserves performance relative to an MNLI-only comparator.

The three candidates are trained independently from `bert-base-uncased` using MNLI, three epochs and learning rate 2e-5. The inversion repair contains 1,215 rows, including 788 strict token-permutation inversions labelled non-entailment. The equal-budget mitigation has 608 inversions and 607 passivised entailments. Passivised rows are not strict permutations: this changes label composition rather than balancing the exact flagged pairing. MNLI validation has ten entailed strict-permutation cases among 9,815 rows. Those ten requested cases were not scored before and after repair; HANS measures a broader shortcut and cannot substitute for them.

Table F.2. BERT seed 1: accuracy (%) and non-entailment emission rate (%). The first row is the independently trained MNLI-only comparator, not a checkpoint continued by the other candidates.

arm, verdict seed	target inversion subcase	MNLI validation	HANS `le_ conjunction`	non-entailment on entailed subcases
MNLI-only comparator	0.4	84.0	78.6	3.0
released repair	100.0	84.0	39.4	35.5

arm, verdict seed	target inversion subcase	MNLI validation	HANS `le_ conjunction`	non-entailment on entailed subcases
label-composition mitigation, equal budget	100.0	83.8	61.5	18.0
authors' same-size shuffle control	0.0	84.1	88.9	1.0

Table F.3. BERT repetition: target accuracy (%), MNLI/HANS accuracy differences (percentage points), and non-entailment emission rates (%). Brackets reproduce the intervals in the retained results; `base` is the matched MNLI-only comparator.

seed	target, base → repair	MNLI validation, repair - base	HANS harm, repair - base	restoration, restored - repair	non-entailment on entailed subcases: base / repair / restored
1	.4 → 100.0	+.05 [-.43, +.53]	**-39.2** [-42.3, -36.1]	**+22.1** [+19.2, +25.0]	3.0 / 35.5 / 18.0
2	.5 → 100.0	-.04 [-.51, +.43]	**-28.5** [-31.6, -25.4]	**+14.9** [+12.0, +17.8]	3.6 / 25.8 / 18.1
3	0 → 100.0	+.16 [-.31, +.64]	**-21.9** [-24.6, -19.2]	**-7.4** [-9.5, -5.3]	1.4 / 21.7 / 23.8

The seed-1 rescue criterion requires at least ten HANS points relative to repair, corrected $p < .05$, and target retention within five points. It passes, but the candidate remains 17.1 points below the independently trained MNLI-only comparator. Mitigation improves the HANS subcase at seed 2 and worsens it at seed 3. The post-hoc all-seed policy therefore blocks. Rescue and preservation answer different questions; the already-published HANS degradation is not a new discovery here.

F.3 Drawing-task controls

The motivating proprietary task predicts a dimension path, derivation, calculation and answer for a drawing field. Twenty-seven contracts define units and valid references. The base corpus has 2,215 rows. A 24-row repair fixes direction to left-to-right; reference direction counts are 66 left-to-right, 25 right-to-left and seven other. Evaluation includes 188 right-to-left sheets. No private image or identifier is reproduced.

Table F.4. Drawing-task controls. Zero side-token loss was tested without installing the target repair.

Tested possibility	Intervention	Observation
Added-row volume	Row-matched control	Target and side match the start
Pixel reading	White images in place of sheets	Repair writes its side on 516/516; start writes its opposite default on 542/547
Original side-label error	Geometry-correct labels in both repair sets	Fixed constant and balanced contrast recur
Delayed side fitting	Corpus only, four epochs; higher-rate checks	Minority probability .32, .78, .41, .28; higher-rate endpoints .44 and .52
Shared adapter movement	Zero loss on every direction token for two epochs	Minority .24 and .25 versus start .27
Reading at parity	25:25 and held-out 33:12 arms	Seen-majority .80 versus unseen .52; minority seen .329 versus unseen .315
Decodable side representation	Linear and 64-unit probes across layers and checkpoints	Best AUC .563; no cell reaches .60
Target-row supply	30-row sum control against 60-row arm	72.6% versus 74.2%, $p = .68$

A zero-loss side-token control leaves minority-output probability at .24/.25 versus .27 at the start. It shows little movement under that control; the target-installing masked repair was not run. A linear

probe reaches best AUC .563, so this drawing run does not show the decodable factor found in the small model.

The first balanced draw was exploratory. On a second independent draw, target accuracy is 89.1% versus 86.8% for fixed repair (12 gains, eight losses, exact p=.503); repair-value emission on opposite-side sheets falls from 99.5% to 12.8%. Modal shares remain 81.6%, 71.0% and 87.2% across the three strata, missing the pre-recorded 80% every-stratum bar. The candidate stays blocked. The two balanced draws lean in opposite directions, so breaking the original constant does not establish restored reading.

Teacher-forced gradient checks also reject two proposed explanations. Opposite-side corpus rows have median residual .651, not at most .20 as proposed, and gradient coherence .14 versus the repair rows' .08, not the proposed twofold repair advantage. Residual mass or within-group coherence alone does not explain the update. These are separate drawing diagnostics, not attribution of the Qwen2.5 checkpoint margins.

Drawing accuracy uses a 0.51 mm tolerance, exact paired McNemar tests and Wilson intervals. Parsing failures and missing numbers are wrong. One LoRA epoch over the corpus plus repair takes 3.5–4.5 hours on an 80 GB A100. The support scan takes about ten CPU seconds; construction and decoding dominate the audit cost.

APPENDIX G. SMALL-MODEL DIAGNOSTIC STUDY

Status. Exploratory. The design, arms and readouts (a project design record) were written before the first run; the starting-state sweep and both head refits below were added after the main grid was read and are labelled post hoc. The study runs in under a minute on a laptop CPU.

Model and data. Inputs have 24 dimensions: 6 determine a four-class target by a fixed random projection, 1 carries the binary factor F with Gaussian noise ($\sigma = 0.8$, Bayes accuracy about 0.89), and 17 are nuisance. Every row labels both the target and the sibling ($= F$). A two-layer ReLU encoder of width 64 feeds a target head and a sibling head. Pretraining uses Adam at 10^{-3} for five epochs on 20,000 rows with F balanced and target class 3 at 2%. Repairs continue training for two epochs (48 steps, batch 16) on 384 rows of class 3: `fixed` ($F = 0$ on every row), `balanced` (192/192) and `masked` (`fixed` with the sibling loss zeroed). `full` updates all parameters; `lowrank` trains rank-2 adapters on both encoder layers with the heads frozen. Learning rates are 3×10^{-4}, 10^{-3}, 3×10^{-3} and 10^{-2}; five seeds. Backpropagation was checked against finite differences.

Readouts. Alternate-value accuracy uses 1,000 fresh class-3 rows with $F = 1$. For both factor values, the signed sibling margin is $z_1 - z_0$. Let μ_f be its mean on class-3 rows with $F = f$. The global shift is the pre-to-post change in $(\mu_0 + \mu_1)/2$; the reading contrast is $\mu_1 - \mu_0$. The alternate-value margin change therefore equals the global shift plus half the contrast change. The first-order sum $\sum_t \langle \nabla M, \Delta\theta_t \rangle$ uses the mean $F = 1$ margin in the trained parameterisation. The probe fits a fresh logistic regression of F on the 2,000 evaluation features (1,000 per factor value), randomly split into separate probe-training and probe-test halves. Features are standardised using the training half; fitting uses 400 gradient steps at rate 0.1 with L2 penalty 10^{-3}.

The sibling KL is $D_{\mathrm{KL}}(p_{\mathrm{post}} \| p_{\mathrm{pre}})$, averaged over the repair rows. The both-heads statistic sums the target and sibling KL values. These definitions follow the retained analysis implementation.

Table G.1. Small-model grid: means over five seeds. Accuracies are fractions; margin summaries are in nats. Rows spanning learning rates report ranges across those rates, not uncertainty intervals.

arm, update	lr	counter-case sibling	target	global shift	contrast	first-order / actual
fixed, full	3e-4	0.89 → 0.52	0.56 → 0.97	−2.7	5.8 → 5.5	−2.9 / −2.9
fixed, full	1e-3	0.89 → 0.06	0.56 → 1.00	−6.0	5.8 → 5.2	−6.3 / −6.3
fixed, full	3e-3	0.89 → 0.00	0.56 → 1.00	−10.4	5.8 → 4.7	−10.9 / −11.0
fixed, full	1e-2	0.89 → 0.00	0.56 → 1.00	−26.3	5.8 → 4.0	−26.9 / −27.2

arm, update	lr	counter-case sibling	target	global shift	contrast	first-order / actual
fixed, rank-2	1e-3	$0.89 \to 0.85$	$0.56 \to 0.74$	-0.4	$5.8 \to 5.7$	-0.5 / -0.5
fixed, rank-2	3e-3	$0.89 \to 0.37$	$0.56 \to 1.00$	-3.4	$5.8 \to 5.2$	-3.6 / -3.7
fixed, rank-2	1e-2	$0.89 \to 0.00$	$0.56 \to 1.00$	-10.5	$5.8 \to 3.8$	-11.2 / -11.5
balanced, full	1e-3 to 1e-2	0.87–0.91	1.00	-0.2 to $+0.8$	5.9–8.9	near 0
masked, full	1e-3 to 1e-2	0.83–0.88	1.00	-0.1 to $+0.1$	4.2–5.5	near 0

Values are means over five seeds; margins are in nats. No rank-2 cell collapses at 3×10^{-4}. As an accounting check across all 120 repairs, the first-order sum and actual change correlate at 1.000, with a median absolute remainder of 2% of the change. Across the 40 fixed repairs, the global shift is a median 92% of the counter-case margin change, and the Spearman correlation of the sibling drop with on-support KL is 0.93 for the sibling head and 0.82 for both heads.

Post hoc: bias refit. After each fixed full repair, we fitted only the sibling-head bias on 64 newly sampled labelled class-3 rows (32 per value), separate from repair and evaluation rows, using 500 gradient steps at rate 0.5. Tables G.2–G.3 report recovery on alternate rows and the trade-off on fixed-value rows. That trade-off accompanies a shrinking reading contrast; it does not by itself show loss of factor information.

Post hoc: full head refit. On the same 64 rows per seed, we froze the encoder and trained a fresh binary linear sibling head from zero on standardised features (1,000 full-batch gradient steps, rate 0.1, L2 penalty 10^{-3}; fixed before examining these outcomes). No repair or evaluation row entered the fit. At 10^{-2}, alternate-value accuracy is 81.6%, 79.4%, 87.0%, 81.4% and 75.3% across seeds, versus 67.2%, 63.3%, 67.0%, 70.9% and 65.4% after bias-only refitting. Head refitting improves over bias-only recovery at this rate for all five seeds, while remaining below pre-repair accuracy. Tables G.2–G.3 give all rate-level means. This is a fresh diagnostic classifier, not the original head repaired in place.

Table G.2. Small-model post-hoc refits: mean accuracy (%) over five seeds. Each refit uses the same 64 calibration rows (32 per factor value), separate from repair and evaluation rows.

learning rate	alternate before repair	after fixed	after bias refit	after head refit
3×10^{-4}	88.9%	51.8%	88.6%	84.5%
10^{-3}	88.9%	5.8%	86.7%	84.3%
3×10^{-3}	88.9%	0.0%	81.9%	83.2%
10^{-2}	88.9%	0.0%	66.8%	80.9%

Table G.3. The same candidates on fixed-value evaluation rows. Means over five seeds; accuracy (%). This reports the other side of the refit trade-off.

learning rate	fixed-value before repair	after fixed	after bias refit	after head refit
3×10^{-4}	88.3%	98.9%	87.1%	81.7%
10^{-3}	88.3%	100.0%	84.2%	81.4%
3×10^{-3}	88.3%	100.0%	78.3%	81.2%
10^{-2}	88.3%	100.0%	62.4%	80.3%

Post hoc: starting state. Pretraining for 1, 2, 5 or 15 epochs gave starting counter-case margins of 2.5–3.2 nats and fixed-repair accuracies of 0.05–0.09 at 10^{-3}. The manipulation did not produce distinct starting states, so the study is silent on starting-state dependence.