WHY ERRORS SURVIVE SELECTIVE GROUNDED CORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can make mistakes even when given the sources needed to answer. A support score is the fraction of output claims fully supported by those sources. But does a higher score mean more mistakes were fixed? Across legal, document, and open-domain question answering, we study selective correction, which edits only claims a checker marks as errors. Holding the first edit fixed, we compare removing claims that still fail with editing them again. We find that **an answer can say less and score higher, or fix a mistake and keep the same score.** Editing again restores support to 37 claims that removal would discard, yet changes average support by only -0.6 to $+0.2$ percentage points. Weaker wording can also raise scores: adding "It appears that" before checking and removal raises support by 2.0–2.5 points over removal alone in our legal experiments. Human annotators find that all 52 sampled hedges narrow the assertion and that some credited repairs edit already supported claims. In 17 of 18 traced configurations, most surviving errors are not sent back for another edit, and failed repairs dominate the other. We call for reporting what each step repairs, preserves, and removes, with its token cost.

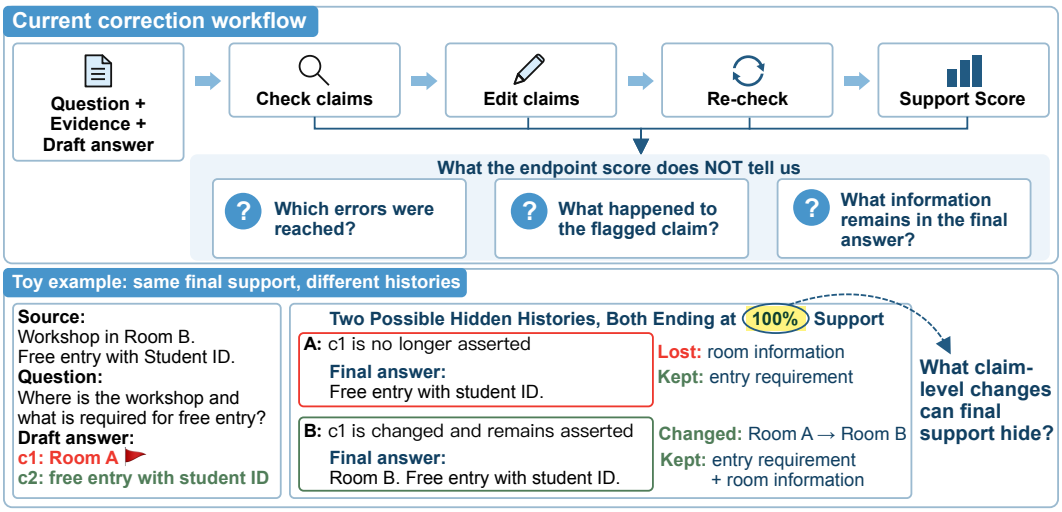

Figure 1: **The same score can hide a useful repair.** Example: A removes the incorrect room claim, while B corrects it. Both have 100% of their remaining claims supported, but only B answers both parts of the question.

1 INTRODUCTION

Large language models (LLMs) increasingly answer questions from supplied contracts, reports, and retrieved passages. Even with these sources, an answer may give the wrong date, omit an important condition, or say more than its citation supports (Gao et al., 2023b). Checking and editing improve answers in many settings (Gao et al., 2023a; Madaan et al., 2023; Gou et al., 2024). If one correction helps, trying again seems natural. But *what does another round actually improve?*

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.

A support score alone cannot tell a repaired error from a removed claim or a weaker assertion. Recent work improves correction through training and more precise error localization (Kumar et al., 2025; Samanta et al., 2026). Closest to this work, Polat et al. (2026) separate detection from repair in chemistry, vary inspection scope, and examine the evaluation criterion. Their strongest repair results supply correct reference facts. Our question differs: what does a further step add *after an identical first edit*? By matching that edit, we separate repair from removal. Controlled wording changes then test for weaker assertions.

Across ⚖ Legal, 📄 Documents, and 🌐 Open-domain QA, we study a *selective* workflow that edits only flagged claims, re-checks only the rewrites, and filters out claims still rejected (Figure 2). We also compare whole-answer self-check, where the generator checks and revises its own answer. After one self-check, adding the selective loop raises support by 2.6–10.1 percentage points for every model and setting. This gain mixes editing with filtering, so separate matched comparisons hold the first selective edit fixed.

These comparisons reveal that **support gains and successful repairs can move apart**. Filtering raises support in 16 of 18 task, generator, and editor configurations without rewriting any kept claim. Another selective edit moves average support by only -0.6 to $+0.2$ points, yet it restores support to 37 claims that stopping would delete, out of 980 unsupported after the first edit. Such repairs can stay hidden because answers with one or two supported claims can both score 100% (Figure 1).

Tracing the remaining errors shows why progress stalls. In 17 of 18 traced configurations, most errors left after the first edit lie outside the *edit list* of claims selected for the next edit. In one ⚖ Legal trace, the first check had missed 33 of the 37 errors present before filtering. Failed repairs instead account for 18 of the 30 surviving errors in the 📄 Documents trace whose first check has the highest recall. A similar stall can thus arise from **missed checks or failed repairs**. Re-inspecting overlooked claims and changing editors recover support but lose some supported content. Choosing the next step requires knowing which errors it can reach and fix.

A rising score also leaves open what changed. Adding the fixed phrase "It appears that", then checking and filtering as usual, raises legal support without using evidence to choose the edit. Two annotators independently judge that all 52 sampled hedges narrow the assertion. They also find credited repairs of already supported claims in selective correction and published-method adaptations. Judging only the final answer can confuse a wording change with a needed correction.

Correction should be judged by what the answer gains and keeps. We call for builders to report *what was fixed, what was lost, and what was overlooked* at each step, with its token cost, and for evaluators to assess both versions of a claimed repair.

2 Background and Related Work

Grounding and evidence-based editing. ALCE, FActScore, AttributionBench, and MiniCheck measure evidential support for generated claims (Gao et al., 2023b; Min et al., 2023; Li et al., 2024; Tang et al., 2024). RARR studies attribution alongside preservation, and PURR, GenAudit, and FAVA develop editors for unsupported spans (Gao et al., 2023a; Chen et al., 2023; Krishna et al., 2025; Mishra et al., 2024). VeriCite combines verification with re-synthesis (Qian et al., 2025), and AtomCite verifies supplied page citations (Qian et al., 2026a). GANDR audits legal claims and attributes its gains to drafting and a commit rule rather than rewriting (Qian et al., 2026b).

Self-correction, localization, and stopping. Self-Refine, Reflexion, CRITIC, and Self-RAG use feedback to guide generation or revision (Madaan et al., 2023; Shinn et al., 2023; Gou et al., 2024; Asai et al., 2024). Intrinsic correction can fail (Huang et al., 2024), while training, structured localization, and generation-time methods can improve it (Kumar et al., 2025; Samanta et al., 2026; Li, 2025; Gao et al., 2026). VRR-Stop models noisy verification to choose between repairing and committing (Wu et al., 2026), and Appendix F reports our restricted adaptation. Holding models fixed, we compare correction after generation, including whole-answer self-check as a baseline.

From detection bottlenecks to continuation gains. Polat et al. (2026) already study detection, repair, inspection scope, and reference criteria. Because our location-only oracle leaves the correction to the editor, its repair rates are not directly comparable to their correct-fact intervention. Our contribution is a matched attribution of an added step's *incremental progress* to repair or removal. Identical

first edits separate filtering from retry, and claim histories expose rescues hidden by the support score. Controlled wording changes probe whether a higher score reflects a better answer, and human before/after judgments test this distinction across correction families.

3 SEPARATING THE OPERATIONS OF CORRECTION

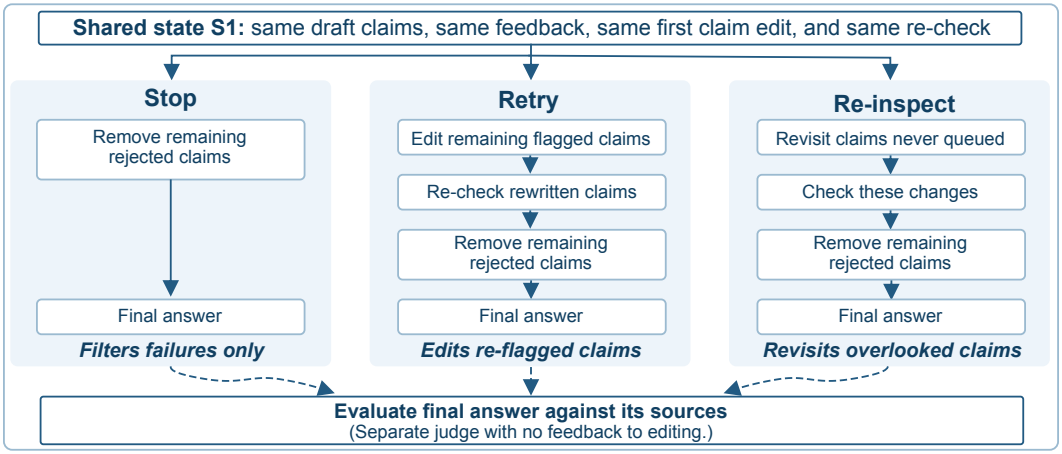

Figure 2: **Same starting point, different next steps.** All arms share S_1 before filtering. Stop removes remaining failures. Retry edits re-flagged claims, then re-checks and filters. Re-inspect revisits accepted claims never queued, then checks changes and filters. Dashed arrows lead to a separate evaluator that provides no editing feedback.

We compare complete correction strategies, including self-check followed by the selective loop, then isolate operations within selective correction. In *self-check*, the generator revises its whole answer from the sources, without a separate checking step or explicit edit list. Selective correction exposes that list, showing which claims are inspected, changed, or removed.

3.1 A SHARED-PREFIX EXPERIMENT

A claim is a factual statement, such as the entry requirement in Figure 1. In selective correction, the *generator* writes the answer, the *checker* (the in-loop verifier) tests its claims against their citations, and the *editor* revises the flagged claims, which form the *edit list* (or queue). A separate evaluator judges the outputs and sends no feedback to these roles.

Each item pairs a question with a fixed evidence store. The generated answer lists numbered claims, each with one citation, and a short answer line. The first check examines every claim, and later checks examine *only the rewrites*. Checks keep supported and uncertain claims and, on Documents, may move a citation to a supporting neighboring page. The editor may reword, narrow, or drop flagged claims, leaving unflagged claims unchanged. Appendix A gives full rules and prompts.

Five shared-prefix arms form a *ladder*, starting from DELETE, which removes the claims on the initial edit list. EDIT instead rewrites them once and keeps every rewrite. RE-CHECK adds verification labels to the rewrites and may update a page citation, giving the state S_1 shared by the continuations in Figure 2. STOP applies the final filter to S_1, removing claims still rejected, including initially flagged claims that the editor skipped and that were therefore never re-checked. This filter leaves the short-answer line unchanged. RETRY edits re-flagged rewrites again before the same final filtering. **From EDIT onward, all arms share the edit list, feedback, and first claim rewrites.** EDIT is reconstructed from the saved first edit, and a prefix audit confirms claim-level matching, with one short-answer parsing exception (Appendix B.4).

Why hold the first edit fixed? Independently generated rewrites would mix variation in the first correction with the value of continuing. The fixed prefix keeps that variation out of the contrasts: RE-CHECK versus EDIT isolates checking and citation updates, STOP versus RE-CHECK isolates

Error type	Definition	Example
Missed error	*An unsupported claim is never selected for editing and therefore survives unchanged.*	SOURCE: Free entry **with ID** **Free entry.** *missing ID* → not flagged → **Free entry.** *still missing ID*
Failed repair	*An unsupported claim is selected and edited, but the edited claim is still unsupported.*	SOURCE: **Room B** **Room A** *wrong* → edit A to C → **Room C** *still wrong*
New error	*A previously supported claim is changed during editing and becomes unsupported.*	SOURCE: **Ends at 5 pm** **Ends at 5 pm** *correct* → edit 5 to 6 → **Ends at 6 pm** *now wrong*

Figure 3: **The same support failure can have a different history.** Examples show a missing condition left unchecked, an unsuccessful room correction, and a correct time changed into an error. Arrows mark what happens to each claim. Deleted claims count separately as removed content.

filtering, and RETRY versus STOP measures what another generative attempt adds. A gain from removal therefore cannot be credited to a second successful rewrite.

3.2 THREE PATHS TO A SURVIVING ERROR

An *error* is a claim that fails the evaluator's criterion for full source support, whether it states a false fact, omits a qualification, or has incomplete support. Figure 3 distinguishes three histories: a *missed error* is never selected for editing, a *failed repair* is selected and edited but remains unsupported, and a *new error* turns previously supported content into an unsupported claim during editing. Removal is a fourth outcome, counted separately as deletion.

Correction must both find and fix a problem. We therefore measure *reach*, the fraction of initial errors the first check selects (*selector recall*), separately from *repair*. Selected changes include rewrites and document citation corrections. If only 4 of 10 errors reach the editor, even a perfect editor leaves 6 untouched. Missed errors stay unchanged, so their count is a lower bound on surviving errors. Comparing this bound with actual survivors separates missed errors from later failures (Appendix E.1). The experiments below therefore ask how many errors a real checker misses, which ones an editor can fix, and at what cost.

Random, checker, and oracle edit lists test error selection on each model's own drafts, and claims changed by whole-answer self-check give another comparison. Oracle lists mark error locations without corrections and can be longer than checker lists, so costs are unmatched. Editor swaps test repair on fixed Qwen3.8-27B-FP8 (Qwen Team, 2026b) drafts and edit lists. Each editor uses its own interface, and the swap includes that difference. Forced retry re-edits every initially flagged claim, whereas RE-INSPECT asks whether overlooked claims can improve. Its prompt supplies the ids of accepted claims never queued, with the answer context and sources. Only those claims may change, and changes are checked before filtering (Prompt 23). RE-INSPECT changes scope, prompt, and model role together, leaving the effect of scope alone unresolved (Appendix A.2).

3.3 MEASURE OPPORTUNITY BEFORE TAKING THE NEXT ACTION

We freeze S_1 after the first rewrite and any re-check, *before* filtering or another action. Let Q_2 be the next edit list and U_1 the errors still present under the evaluator. *Retry opportunity*, $|Q_2 \cap U_1|$, counts errors the next edit can reach, and $U_1 \setminus Q_2$ holds those it cannot. The size $|Q_2|$ alone would count supported claims that the checker wrongly re-flags. The final filter can also remove initially flagged claims that the editor skipped, which never enter Q_2. Measuring before filtering keeps these removed errors visible. The evaluator's labels diagnose opportunity and never feed back into retry.

4 EXPERIMENTAL SETUP

Table 1: **Three evidence settings used throughout our evaluation.** Each item provides a fixed evidence store containing the gold source. Every generated claim must cite the supplied source that supports it.

Setting	Benchmark	Supplied evidence	Main n	Task-level evaluation
Legal	LegalBench-RAG CUAD / ContractNLI	8 contract clauses, including the gold source	250 / model	Legal-answer rubric
Documents	MP-DocVQA / DUDE	5–20 document page images OCR also used for verification	250 / model	Answer similarity (ANLS) and gold-span coverage
Open-domain QA	ALCE-ASQA	10 retrieved passages per question	850 / model	Short-answer recall

Three evidence settings. Legal uses LegalBench-RAG over CUAD and ContractNLI, with eight clauses per item (Pipitone & Alami, 2024; Hendrycks et al., 2021; Koreeda & Manning, 2021). Documents uses MP-DocVQA and DUDE, with 5–20 page images and OCR for verification (Tito et al., 2023; Landeghem et al., 2023; Wang et al., 2024). Open-domain QA uses ALCE-ASQA with ten passages per question (Gao et al., 2023b; Stelmakh et al., 2022). These fixed evidence stores contain the gold source, keeping retrieval outside the comparison (Table 1).

Models and scale. Four main models span three families: Opus 5 (Anthropic, 2026a), Sonnet 5 (Anthropic, 2026b), Qwen3.8-27B-FP8, and Gemma4-26B-A4B-FP8 (Gemma Team et al., 2026). Each model checks its own claims in separate steps, with cross-model controls. In editor swaps, Qwen3.5-9B (Qwen Team, 2026a), Sonnet, and Opus edit fixed Qwen3.8-27B-FP8 drafts. Main blocks contain 250 items per Legal/Documents setting and 850 Open-domain QA items, while mechanism and editor-swap studies use 100. The 18 ladder configurations (Figure 4) share nine task–generator draft pools and are dependent. Their Open-domain QA cells use only Qwen3.8-27B-FP8 drafts, and Appendix D.3 compares RETRY with STOP for all four models on the full 850-item block. Eight confirmation configurations use held-back, source-disjoint items and include Gemma4-12B, which is reserved for them. Appendix A gives sampling and access.

Held-out judge and support. GPT-5.6 Sol (OpenAI, 2026), the primary held-out judge (J1), accepts only full support. It never serves as generator or checker but shares the checker's source-isolated interface, rubric, and measurement assumptions. *Support* averages the supported fraction of final numbered claims over answers, excluding the short-answer line. Empty claim sets are undefined and excluded from paired support contrasts, or scored as failures in a sensitivity analysis. *Unsupported mass* counts remaining errors per draft claim. Human annotators and J2, an open second judge reusing the Gemma4-26B-A4B-FP8 generator checkpoint, test this measurement in §5.4.

Retention and cost. One error among four claims gives 75% support, and removing it, alone or with a supported claim, gives 100%. *Kept supported content* separates these cases by judging whether the final answer still asserts initially supported claims. *Same-scope retention* rejects narrowing and uses all draft claims, while rescued initial errors, such as the room claim in answer B of Figure 1, count separately. Cost counts operational tokens, excluding evaluation, and Appendix B gives fields and denominators.

Comparisons and uncertainty. A gain from 90% to 92% support is two *percentage points*, and Table 2 covers 168 method–model–task combinations. Paired bootstrap intervals resample source clusters (contracts, documents, or questions), using 2,000 draws in the self-check and wording follow-ups. They condition on saved outputs, and independent run-level repetitions and equivalence tests were not conducted.

Table 2: **Source support (%).** Bold: highest support per column. Gray: kept supported content below 97%. Overall averages 12 configurations. Cost averages token/draft ratios over eight ⚖ Legal/🗎 Documents configurations. Self-check permits two passes, whereas +loop starts after one. Paired intervals and all 17 variants: Appendix D.

	⚖ Legal $n = 250$ per model				🗎 Documents $n = 250$ per model				🌐 Open-domain QA $n = 850$ per model					
Method	Opus 5	Sonnet 5	3.8-27B	4-26B	Opus 5	Sonnet 5	3.8-27B	4-26B	Opus 5	Sonnet 5	3.8-27B	4-26B	Overall	Cost (×)
Reference baselines														
Draft only	73.3	78.0	83.6	90.6	90.0	80.8	89.7	81.2	68.6	68.7	79.7	91.5	81.3	1.0
Self-Refine	82.3	84.4	84.8	92.2	87.6	86.7	87.5	80.1	86.2	77.0	84.5	91.8	85.4	3.4
CoVe	78.7	80.5	85.6	92.5	90.4	86.2	90.4	79.5	78.5	69.4	82.5	93.0	83.9	5.0
RARR	77.9	83.4	85.8	90.8	90.7	88.3	89.5	81.2	73.2	74.1	83.0	91.9	84.1	2.0
VeriCite	87.4	82.4	91.0	93.5	**96.8**	92.7	94.8	**97.7**	88.0	70.1	90.2	95.2	90.0	3.7
ReAct	86.8	81.4	89.5	90.9	96.4	89.4	92.8	81.5	83.6	72.4	89.1	91.8	87.1	6.5
Our adaptations and controls														
RARR + verifier	86.6	85.8	90.7	93.7	96.3	93.1	93.5	86.2	84.0	75.8	89.9	94.1	89.1	1.9
Delete flags	85.4	84.0	90.8	93.6	94.3	91.0	93.2	93.3	81.9	76.0	90.7	94.8	89.1	1.6
Edit once	87.3	84.8	91.0	94.0	95.0	91.8	91.9	86.0	84.5	75.2	89.9	94.5	88.8	1.9
Stop	87.6	85.3	91.5	94.1	94.9	92.0	92.7	93.5	84.6	76.2	90.7	94.8	89.8	2.0
Retry	87.7	85.3	91.3	94.1	95.1	92.0	92.7	93.6	84.7	76.3	90.7	94.8	89.9	2.0
Delete/regenerate	85.3	85.2	90.4	93.4	90.4	91.1	**95.0**	90.9	81.5	77.7	90.4	94.7	88.8	2.6
Self-check	91.7	87.6	91.1	93.9	92.9	90.6	89.9	80.9	**91.9**	84.1	91.7	94.0	90.0	2.6
Self-check + loop	**92.7**	**89.7**	**93.6**	**95.4**	96.0	**93.2**	89.6	93.9	91.6	**84.1**	**94.4**	**96.2**	**92.5**	2.9

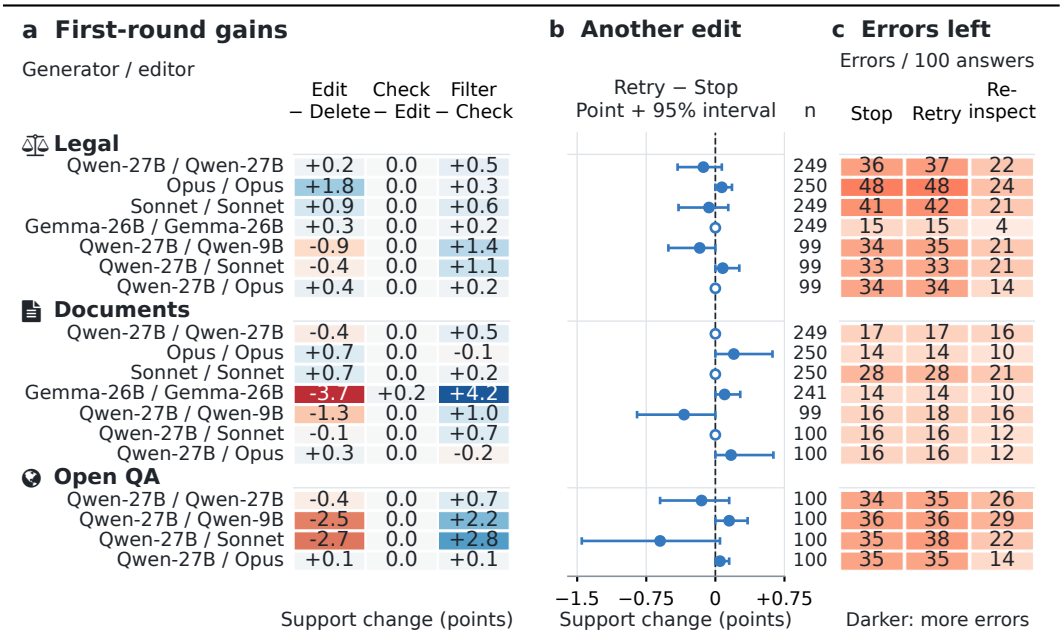

Figure 4: **First-round gains and what happens next.** All panels cover the same 18 task, generator, and editor combinations. **(a)** Paired support changes. Check–Edit is a reconstruction check. **(b)** Retry minus stop with 95% source-cluster intervals and paired n. Hollow points mark all-zero paired differences, which can still hide repairs. **(c)** Final J1 errors in each 100-answer trace, including new errors. Fewer errors can reflect deletion. Qwen-27B, Qwen-9B, and Gemma-26B denote Qwen3.8-27B-FP8, Qwen3.5-9B, and Gemma4-26B-A4B-FP8. Details: Appendices D–E.

5 RESULTS

Table 2 compares complete correction strategies. Figure 4 then takes the selective loop apart, showing what each first-round operation changes (a), what another edit adds (b), and which errors each continuation leaves (c). Together they connect an overall score to what the answer gains and keeps.

5.1 SUPPORT AND RETAINED CONTENT CAN RANK STRATEGIES DIFFERENTLY

Whole-answer self-check is a strong baseline across the three settings. Its average support over twelve configurations is 90.0%, close to selective retry's 89.9%, and it leads for Opus and Sonnet on Open-domain QA. Selective retry is stronger on Documents and retains more initially supported content on average. The ranking therefore depends on the task and on whether evaluation rewards support or retained content.

One configuration makes this concrete. With Gemma4-26B-A4B-FP8 on Documents, VeriCite reaches 97.7% support while retaining 90.9% of initially supported content. Selective retry has lower support, 93.6%, but keeps 98.5% of that content ("Loop, two rounds" in Appendix Table 21). In the same configuration, a RARR variant raises support by filtering without another rewrite (Appendix D.4). Ranking by support alone would hide both the content VeriCite loses and the removals behind the RARR gain.

Can selective checking still help after the model has revised its whole answer? Adding the selective loop to one fixed self-check output raises support in all twelve configurations by 2.6–10.1 points, with every paired 95% interval above zero (Appendix Table 23). The largest gain comes in that same Gemma4-26B-A4B-FP8 Documents configuration, again with a 1.6-point loss of initially supported content. Replacing the loop with a second self-check pass gives support point estimates close to the loop's for Opus and Sonnet on Open-domain QA, at different costs (Appendix Table 24). This increment measures the whole loop, so its gain mixes editing with filtering. The ladder below starts from the draft and holds the first edit fixed, which separates the two.

5.2 SMALL SCORE CHANGES CAN HIDE USEFUL REPAIRS

Editing can preserve information that filtering would remove. Compared with DELETE, which drops the initially flagged claims unedited, EDIT keeps more model-assessed supported content in the four same-model Legal configurations, where one model drafts, checks, and edits (Appendix B.4). Since EDIT and RE-CHECK share the same saved rewrites, adding the check can change support only by updating a citation, which happens in one Documents configuration (Check–Edit in Figure 4a). With those edits held fixed, filtering then raises support in 16 of 18 configurations, by up to 4.2 points, without changing any surviving claim. An edit can therefore save content even when the score gain comes from removal.

A second edit moves the average score far less. Across the 18 ladder configurations, RETRY minus STOP ranges from -0.6 to $+0.2$ support points at extra token cost, and every paired interval crosses or touches zero (Figure 4b). A separate comparison on all 850 Open-domain QA items per model finds similarly small changes. Opus shows a small positive effect, $+0.17$ [$+0.06$, $+0.29$], after excluding eight items whose recorded first edits or check records differ between the two arms (Appendix D.3). Such averages leave open whether any single answer gained something useful.

Claim histories answer that question. Of the 980 claims unsupported before the continuation, RETRY restores support to 37 that STOP would delete, while making three supported claims unsupported. Repairs of this kind appear even in the five configurations with exactly zero paired score differences (hollow points in Figure 4b). In one answer from each, both arms end at 100% support, yet RETRY keeps one more supported claim, as in Figure 1 (Appendix D.2). **A flat support score can hide a useful repair.** To see why progress stays limited, we turn to the errors that the next edit can reach.

5.3 MISSED CHECKS AND FAILED REPAIRS LEAVE DIFFERENT PROBLEMS

In 17 of the 18 ladder traces (100 answers each), most surviving errors lie outside the next edit list. In the Legal trace from Qwen3.8-27B-FP8, the first check missed 33 of the 37 errors present before

filtering. These untouched claims account for nearly all remaining errors, leaving little for another selective edit to reach. Yet errors can also survive repeated repair attempts.

The Gemma4-26B-A4B-FP8 Documents trace supplies that contrast. Its first check has the highest recall among the same-model traces, yet failed repairs account for 18 of the 30 surviving errors and initial misses for only 11 of the 30. More than half of the 30 enter the next edit list, so another edit has work to do. The small continuation gain here accompanies repeated repair failures, whereas the other traces mostly leave errors beyond the editor's reach.

Re-inspection reaches some overlooked errors, restoring support to 16 of the 33 missed Legal claims with Qwen3.8-27B-FP8, while also removing supported content and introducing new errors. Forcing another attempt on initially flagged claims likewise recovers some errors but loses supported claims (Appendix Table 29). An error can disappear through repair or removal, so the lower error counts in Figure 4c must be read alongside these losses.

Supplying every error location does not guarantee a fix either. With this location-only oracle, Qwen3.8-27B-FP8 repairs 4 of 26 errors on its own Documents drafts, whereas Opus repairs 20 of 25 on its different drafts. Repair success thus depends on the tested editor and drafts even when no error is missed. Changing editors also needs held-back evaluation. With fixed Qwen3.8-27B-FP8 drafts and checker on Documents, re-inspection gains $+1.67$ points over stopping on the 100 exploration items for both Sonnet and Opus. On held-back confirmation items, Sonnet's estimate falls to -0.33, while Opus's stays positive at $+1.89$ (Appendix Table 28). An exploratory gain alone is therefore insufficient for choosing an editor.

5.4 Checking both versions changes what counts as a repair

So far, a repair means that a claim moves from unsupported to supported under the held-out judge. That move can follow a factual correction or a weaker assertion. To separate the two, we restrict re-inspection to qualification, to rewording or removal, or to a fixed prefix. Every variant revisits the same accepted claims, then re-checks and filters. The simplest control prepends "It appears that" without consulting evidence, although the subsequent checker still reads the source.

This mechanical change raises support by 2.0–2.5 points over stopping in the three Legal configurations (Figure 5). The effect is less consistent elsewhere: intervals cross zero on Documents, and the alternative prefix "In some cases" lowers the point estimate on the Open-domain QA cohort. The legal gains show that wording changes and the checks they trigger can produce substantial apparent progress. A narrower assertion may be justified, so judging a repair requires examining what each version asserts.

Two annotators with document-analysis experience independently assess both versions, blind to the method and model verdicts. In an audit of 326 judge-credited repairs from selective continuations, published-method adaptations, and mechanical hedges, both annotators judge all 52 hedges to narrow the assertion. With sampling weights, they find 19.8% [12.1, 28.3] of the selective cohort and 29.0% [17.4, 41.9] of the published-method cohort already fully supported before editing, with substantial task differences (Appendix Table 12). Agreement in this audit is $\kappa = 0.78$ on full support and 0.63 on paired outcomes. **Checking the final claim alone can credit a repair where none was needed.**

Human judgments also confirm recoverable gaps outside the edit list. In a separate census of 973 re-inspection outcomes from 511 distinct claims that J1 rejects but the correction procedure never selects, humans judge 22.4% [18.9, 26.0] of these outcomes to reach full support from partial or no support. Requiring repairs to keep the original scope gives a less reliable, secondary endpoint ($\kappa = 0.35$, Appendix C.4). The census thus separates improved support from same-scope repair.

Finally, the support rule determines how much remains to be corrected. Accepting partial support reduces errors from 278 to 71 across nine same-model traces, although most remaining errors still lie outside the next edit list. Much of the strict error count therefore comes from partial support, which includes missing qualifications and need not indicate a false fact. Under J2, the best feasible action among STOP, RETRY, and RE-INSPECT (highest support within 3-point loss limits on supported content, correctness, and quality) changes in 4 of the 8 confirmation configurations (Appendix C). Together, the wording controls and human checks explain why correction needs a record of what changed and an explicit standard for judging it.

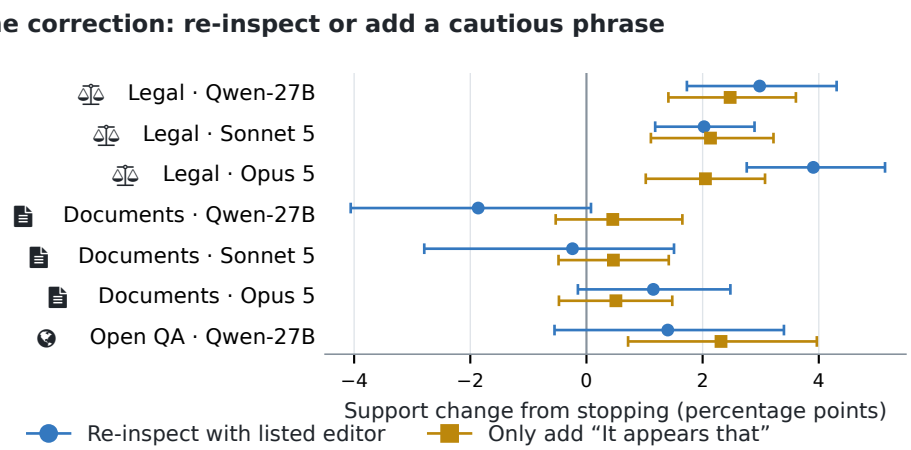

Figure 5: **A cautious phrase can raise the score.** Both continuations start from the same answers, then re-check and filter. Points show support changes from stopping with 95% source-cluster intervals. Rows identify task and editor, with Qwen3.8-27B-FP8 drafts and checker. All editing controls, sample sizes, retention, and cost: Appendix E.6.

6 Implications: What Should the Next Step Do?

Practitioners: protect useful answer content (Appendix D). Selecting by support alone can reward answers that omit requested information while every remaining claim is supported. Whole-answer self-check is a strong baseline for judging whether an added loop answers more of the question without losing supported content. *Define what the answer must retain before comparing correction scores.*

System builders: track what remains to be fixed (Appendix E). A nearly empty next edit list can coexist with recoverable errors elsewhere, and a flat score can hide claims that retry saves, so neither is a complete stopping signal. Use calibration traces to distinguish missed errors from failed repairs before choosing a continuation. Test re-inspection and alternative editors for recoveries and losses on held-back examples, because an editor's exploratory gain may not carry over to new sources. Optimal fixed-budget allocation remains open.

Evaluators: validate the claimed improvement (Appendix C). A final support label cannot show whether a claim needed correction, even when aggregate agreement with humans is high. Our audits find already supported claims among credited repairs from selective correction and published-method adaptations. Retention counts only initially supported content, so rescued claims need their own count. Hold the starting answers and evaluation rule constant, add a fixed-wording control, and judge before/after pairs, separating factual changes, justified narrowing, and removed information. *Calibrate repair judgments on changes between versions, separately from final-answer support.*

7 Conclusion

Our matched comparisons show that a support score alone cannot tell a repair from a removal or a weaker assertion. After one whole-answer self-check, the selective loop still raises support by 2.6–10.1 points in all twelve configurations, yet this gain mixes editing with filtering. With the first edit held fixed, filtering raises support in 16 of 18 configurations, and a mechanical hedge followed by re-checking and filtering raises it further on ⚖ Legal, while retry saves claims at nearly flat support. In 17 of 18 traced configurations, most surviving errors lie outside the next edit list, and in the other, most are failed repairs. We call on builders and evaluators to **judge correction by what the answer gains and keeps**, reporting what each step repairs, preserves, removes, and misses, together with its token cost.

AI USE DISCLOSURE

We used an AI assistant to check grammar and writing quality (e.g., phrasing and clarity) and to assist with debugging and code edits. The authors are responsible for the technical content, experiments, analyses, conclusions, and their verification. The LLM-as-judge (§4, Appendix I.1) is an experimental measurement instrument, evaluated against human annotations (§5.4, Appendix C), distinct from manuscript assistance.

ETHICS STATEMENT

Source support is not a guarantee of truth, completeness, or suitability for legal use. A verifier can remove valid information as well as unsupported claims; the reported retention metrics and claim histories make those losses visible. The experiments use public research benchmarks, not private client documents. Two annotators with document-analysis experience worked independently and blind to the method, model, and model verdicts (Appendix C). Code, prompts, individual annotation labels, model-audit records, and run ledgers will be released with the paper. Both annotators' judgments are retained, including disagreements.

REPRODUCIBILITY STATEMENT

Appendices A–B specify the experimental arms, splits, metrics, and inference. Appendices D–F provide per-model comparisons, paired sample sizes, and the complete confirmation analyses; accompanying supplementary data retain the complete endpoint matrices. Appendix I gives the exact prompts and verifier rules. Numerical figures and tables derive from the same cached records; model-based and human evaluations are distinguished throughout.

REFERENCES

Anthropic. Claude Opus 5 system card, July 2026a. URL https://anthropic.com/claude-opus-5-system-card.

Anthropic. Claude Sonnet 5 system card, June 2026b. URL https://anthropic.com/claude-sonnet-5-system-card.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluis Gomez, Marçal Rusiñol, C.V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Scene text visual question answering. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4290–4300, 2019. doi: 10.1109/ICCV.2019.00439. URL https://ieeexplore.ieee.org/document/9011031.

Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. Purr: Efficiently editing language model hallucinations by denoising language model corruptions, 2023. URL https://arxiv.org/abs/2305.14908.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Jiaxun Gao, Him Wai Ng, and Z. Jane Wang. Once-more: Continuous self-correction for large language models via perplexity-guided intervention. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3CKdjb5SuH.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16477–16508, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.910. URL https://aclanthology.org/2023.acl-long.910/.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6465–6488, Singapore, December 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.398. URL https://aclanthology.org/2023.emnlp-main.398/.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, Johan Ferret, Filippo Galgani, Sertan Girgin, Jean-Bastien Grill, Maarten Grootendorst, Jiaxian Guo, Cassidy Hardin, Yanzhang He, Steven M. Hernandez, Omri Homburger, Léonard Hussenot, Juyeong Ji, Armand Joulin, Aishwarya Kamath, Parnian Kassraie, Olivier Lacombe, Preethi Lahoti, Gaël Liu, Gus Martins, Luciano Martins, Tatiana Matejovicova, Ramona Merhej, Nikola Momchev, Sneha Mondal, Ryan Mullins, Sindhu Raghuram Panyam, Shreya Pathak, Sarah Perrin, André Susano Pinto, Etienne Pot, Angéline Pouget, Alexandre Ramé, Sabela Ramos, Douglas Reid, David Rim, Morgane Rivière, Karsten Roth, Louis Rouillard, Omar Sanseviero, Pier Giuseppe Sessa, Shane Settle, Danila Sinopalnikov, Sara Smoot, Piotr Stanczyk, Andreas Steiner, Lawrence Stewart, Ilya Tolstikhin, Michael Tschannen, Anton Tsitsulin, Nino Vieillard, Renjie Wu, Pingmei Xu, Haichuan Yang, Edouard Yvinec, Biao Zhang, Li Zhang, Joe Zou, Nicolas Aagnes, Abdelrahman Abdelhamed, Jakub Adamek, Shivani Agrawal, Shubham Agrawal, Ibrahim Alabdulmohsin, Jean Baptiste Alayrac, Uri Alon, Chandramouli Amarnath, Ankesh Anand, Chrysovalantis Anastasiou, Setareh Ariafar, François-Xavier Aubet, Kyriakos Axiotis, Federico Barbero, Joelle Barral, Alexei Bendebury, Urs Bergmann, Stanley Bileschi, Kat Black, Mathieu Blondel, Sebastian Borgeaud, Arthur Bražinskas, Ryan Burnell, Robert Busa-Fekete, Mu Cai, Daniele Calandriello, Glenn Cameron, Charlotte Caucheteux, Rahma Chaabouni, Garima Chadha, Jetha Chan, Blake Jianhang Chen, Jesse Chen, Lin Chen, Xu Chen, Derek Cheng, Tzu hsiang Chien, Nikolai Chinaev, Yi Chou, Zhaohui Chu, Benjamin Coleman, Pooja Consul, Sam Conway-Rahman, Scott Crowell, Dylan Cutler, Vivek Dani, Samira Daruki, Anil Das, Daniel Deutsch, Nishanth Dikkala, Li Ding, Qiuhan Ding, Shenil Dodhia, Konstantin Donhauser, Tulsee Doshi, Anca Dragan, Alex Druinsky, Sahil Dua, Zoltan Egyed, Danielle Eisenbud, Daniel Eppens, Cindy Fan, Bahare Fatemi, Yassir Fathullah, Vlad Feinberg, Milen Ferev, Sebastian Flennerhag, Takumi Fujimoto, João Gabriel Oliveira, Isaac Galatzer-Levy, João Gante, Simon Geisler, Soham Ghosal, Antonious M. Girgis, Tamara von Glehn, Alec Go, Alhaad Gokhale, Alex Grills, Yiming Gu, Mayank Gupta, Pramod Gupta, Guru Guruganesh, Raia Hadsell, Hamza Harkous, Jitendra Harlalka, Demis Hassabis, Anja Hauth, Joe Heyward, Arian Hosseini, Chih-Yang Hsia, I-Hung Hsu, Xiaopeng Huang, Yangsibo Huang, Kevin Hui, Adrian Hutter, Te I, Fotis Iliopoulos, Advait Jain, Ganesh Jawahar, Ziwei Ji, Qilin Jin, Melvin Johnson, Kandarp Joshi, Arun Kandoor, Wang-Cheng Kang, Koray Kavukcuoglu, Mehran Kazemi, Kathleen Kenealy, Amr Khalifa, Phoebe Kirk, Ivan Korotkov, Suraj Kothawade, Vitaly Kovalev, Neel Kovelamudi, Adam Kraft, Ravin Kumar, Vivek Kumar, Harish Kuppam, Justin Lannin, Chen-Yu Lee, Seungji Lee, Dmitry Lepikhin, Alon Levkovitch, Dongdong Li, Qiujia Li, Valentin Liévin, Ethan Lin, Ziqian Lin, Casper Liu, Tianlin Liu, Tianqi Liu, Xin Liu, Ivan Lobov, Mayank Lunayach, Min Ma, Gagan Madan, Andrii Maksai, Eric Malmi, Michal Matuszak, Daniel McDuff, Gaurav Menghani, Maciej Mikuła, Daniil Mirylenka, Karolis Misiunas, Vedant Misra, Andreea Mitran, Kareem Mohamed, Maksim Mukha, Eric Noland, James O'Donnell, Brendan O'Donoghue, Kate Olszewska, Bernett Orlando, Wanqiong Pan, Rina Panigrahy, Unnati Parekh, Nicolas Perez-Nieves, Chunjong Park, Eric Paskie, Liqian Peng, Bryce Petrini, Slav Petrov, Jonas Pfeiffer, Bilal Piot, Martyna Plomecka, Siim Poder, Octavio Ponce, Arijit Pramanik, David Racz, Anish Rajan, Michelle Ramanovich, Anand Rao, Marvin Ritter, Vitor Rodrigues, Evan Rosen, Mikołaj Rybiński, Noveen Sachdeva, Michaël E. Sander, Rohit Sathyanarayana, Sagar Savla, Samuel Schmidgall, Tal Schuster, George Scrivener, Benoit Seguin, Andrew Sellergren, Aliaksei Severyn, Izhak Shafran, Dhruv Shah, Bobak Shahriari, Yuan Shangguan, Ashish Shenoy, Pradeep Shenoy, Rakesh

Shivanna, Pauline Sho, Lucas Spangher, Wojciech Stokowiec, Tim Strother, Yao Su, Yinghao Sun, Mukund Sundararajan, Andrea Tacchetti, Mor Hazan Taege, Pouya Tafti, Jean Tarbouriech, Chetan Tekur, Shantanu Thakoor, Rahul Thapa, Madeleine Traverse, Lenart Treven, Tao Tu, Chien Te Tung, Çağlar Ünlü, Petar Veličković, Malini Pooni Venkat, Sagar Gubbi Venkatesh, Vidya Venkiteswaran, Francesco Visin, Alex Vitvitskyi, Kiran Vodrahalli, Weiyi Wang, Xin Wang, Tris Warkentin, Jan Wassenberg, John Wieting, Cindy Wu, Lechao Xiao, Hao Xu, Yuhui Xu, Fuzhao Xue, Arun Yadav, Jun Yan, Antoine Yang, Lin Yang, Ming-Hsuan Yang, Ziyu Ying, Jae Hyeon Yoo, Morteza Zadimoghaddam, Sajjad Zafar, Fred Zhang, Jiageng Zhang, Jianyi Zhang, Xiaofan Zhang, Chao Zhao, David Zhou, and Chen Zou. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. CUAD: An expert-annotated NLP dataset for legal contract review. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=7l1Ygs3Bamw.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.164. URL https://aclanthology.org/2021.findings-emnlp.164/.

Kundan Krishna, Sanjana Ramprasad, Prakhar Gupta, Byron C. Wallace, Zachary C. Lipton, and Jeffrey P. Bigham. GenAudit: Fixing factual errors in language model outputs with evidence, 2025. URL https://arxiv.org/abs/2402.12566.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=CjwERcAU7w.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Jordy Van Landeghem, Rafał Powalski, Rubèn Tito, Dawid Jurkiewicz, Matthew Blaschko, Łukasz Borchmann, Mickaël Coustaty, Sien Moens, Michał Pietruszka, Bertrand Ackaert, Tomasz Stanisławek, Paweł Józiak, and Ernest Valveny. Document understanding dataset and evaluation (dude). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19471–19483, 2023. doi: 10.1109/ICCV51070.2023.01789. URL https://ieeexplore.ieee.org/document/10376937.

Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. AttributionBench: How hard is automatic attribution evaluation? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14919–14935, Bangkok, Thailand, August 2024.

Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.886. URL https://aclanthology.org/2024.findings-acl.886/.

Yin Li. Decomposing llm self-correction: The accuracy-correction paradox and error depth hypothesis, 2025. URL https://arxiv.org/abs/2601.00828.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3666122.3668141.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=dJMTn3QOWO.

OpenAI. GPT-5.6 Sol model. OpenAI API documentation, 2026. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol.

Nicholas Pipitone and Ghita Houir Alami. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain, 2024. URL https://arxiv.org/abs/2408.10343.

Can Polat, Mustafa Kurban, Erchin Serpedin, and Hasan Kurban. Grounded verification of chemical and materials reasoning: detection is the bottleneck, 2026. URL https://arxiv.org/abs/2607.17417.

Chen Qian, Yimeng Wang, Yu Chen, Lingfei Wu, and Andreas Stathopoulos. AtomCite: Verification and correction of supplied page-level citations in multi-page documents, 2026a. URL https://arxiv.org/abs/2609.05802.

Chen Qian, Yimeng Wang, Yu Chen, Lingfei Wu, and Andreas Stathopoulos. GANDR: Claim auditing for verifiable legal answer generation, 2026b. URL https://arxiv.org/abs/2609.10293.

Haosheng Qian, Yixing Fan, Jiafeng Guo, Ruqing Zhang, Qi Chen, Dawei Yin, and Xueqi Cheng. VeriCite: Towards reliable citations in retrieval-augmented generation via rigorous verification. In *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2025, pp. 47–54, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400722189. doi: 10.1145/3767695.3769505. URL https://doi.org/10.1145/3767695.3769505.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Ankur Samanta, Akshayaa Magesh, Ayush Jain, Kavosh Asadi, Youliang Yu, Daniel Jiang, Boris Vidolov, Kaveh Hassani, Paul Sajda, Jalaj Bhandari, and Yonathan Efroni. Structure enables effective self-localization of errors in LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=ht7KSypUpV.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc. URL https://dl.acm.org/doi/10.5555/3666122.3666499.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: Factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 8273–8288, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.566. URL https://aclanthology.org/2022.emnlp-main.566/.

Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8818–8847, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.499. URL https://aclanthology.org/2024.emnlp-main.499/.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recogn.*, 144(C), December 2023. ISSN 0031-3203. doi: 10.1016/j.patcog.2023.109834. URL https://doi.org/10.1016/j.patcog.2023.109834.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction, 2024. URL https://arxiv.org/abs/2409.18839.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Yitao Wu, Si Shen, Rui Yang, Hong Peng, and Bin Hu. Verify, repair, repeat, or stop? robust stopping for noisy verify-repair loops in llm agents, 2026. URL https://arxiv.org/abs/2607.17641.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

GUIDE TO THE APPENDIX

Section	What to look up	Page
A	Experimental arms, baseline adaptations, model access	15
B	Metric denominators, paired inference, token/cost accounting	18
C	Human repairs, wording audits, evaluator sensitivity	20
D	Operation effects, baselines, the increment after self-check	30
E	Claim histories, editor controls, restricted editing permissions	42
F	Held-out action effects and secondary policy analysis	51
G	Worked repairs, narrowing, and deletion	53
H	Scope of inference and extended discussion	53
I	Exact prompts and verifier mapping	55

The Appendix retains the evidence needed to assess the main claims. The accompanying supplementary data preserve complete result matrices, endpoint-specific sample sizes, intervals, and claim-action counts. Compact tables here summarize those records without averaging away model differences or omitting unfavorable configurations.

A EXPERIMENTAL SPECIFICATION

This section supplies the operational detail behind §3. The main text uses descriptive operation names, while detailed results also retain condition codes. In particular, the condition called *repair* in the experiment specification performs citation correction and deletion, without a second generative repair attempt.

Table 3: **Correction operations.** The action study calls retry and re-inspect depth and breadth, respectively. The names below distinguish the operations performed.

Main-text name	Condition	Operation and comparison
Draft only	a / B0	Generate claims with one cited source each, without correction.
Flag	b	Verify without changing the claim text.
Delete	c / B1	Correct page citations where possible, then delete failed claims.
Edit	d1p / B5′	Rewrite the queue once and keep every rewrite without rechecking.
Re-check	d1k / B5	Same rewrite, with verification labels and possible recitations.
Stop	d1 / B6	Remove remaining rejected claims, including unaddressed initial flags ($R = 1$).
Retry	d / B7	Re-edit re-flagged claims before final filtering ($R = 2$).
Re-inspect	d1-sc	Re-examine accepted, never-queued claims from the shared state.
Forced retry	d-force	Re-edit every initially queued claim, regardless of re-check.
Independent check	d-ind	Use a different verifier family to select the second queue.

A.1 CYCLE AND MATCHED COMPARISONS

The draft contains an ISSUE line, 2–6 claims (1–5 for documents), and an ANSWER line. Each claim cites exactly one source. The source-isolated verifier checks only the cited evidence and returns a verdict plus a short reason. Supported and uncertain claims are kept. Document citations can be moved to a supporting neighboring page. Other claims enter the queue. The editor receives the marked answer, reasons, and allowed source ids. It may reword, narrow, or drop queued claims, but not add claims or edit unqueued claims. Rewrites are re-checked, and remaining failures are removed after the permitted rounds.

All shared-prefix comparisons reuse cached calls. B5′ is reconstructed from B5 by reverting re-check re-citations and subtracting their cost. This preserves the same sampled first edit. Numbered claims are identical on text tasks, but document re-citation can change them. The prefix audit compares draft claims, queues, feedback, and first claim rewrites. Reusing records establishes matching without requiring deterministic remote calls. One short-answer parsing discrepancy remains (Appendix B.4), so the matching claim is restricted to numbered claims.

Verification follows AtomCite's source-isolated interface (Qian et al., 2026a), specified in Appendix I.1. Legal-answer evaluation follows GANDR's rubric (Qian et al., 2026b), with thresholds in Appendix B.

A.2 Interventions and baseline adaptations

Selector. Random queues match verifier queue size with three seeds. Oracle queues sample draft claims the held-out judge finds unsupported at nominal fractions 0.25, 0.50, 0.75, and 1.00. The implementation rounds within each item and selects at least one error when any is present, so achieved recall can be substantially higher than the nominal fraction. Plots use measured recall, including verifier-triggered page re-citations. These queues diagnose reach and repairability. They are not deployable selectors. The model's whole-answer self-check supplies a further touched set.

Editor and feedback. Qwen3.5-9B, Sonnet 5, and Opus 5 replace Qwen3.8-27B-FP8 as editor while its draft and verifier queue stay fixed. Feedback variants provide no reason, an evidence quote, or a shuffled reason of matched length. Round limits are 1, 2, and 3. Full redraft rewrites the entire answer from tool feedback.

Second-round controls. Forced retry revisits every round-0 queued claim. The independent-verifier control uses Gemma4-26B-A4B-FP8 for Qwen3.8-27B-FP8 configurations and Qwen3.8-27B-FP8 for Opus configurations to determine the second queue. Its inspection scope remains restricted to rewrites.

Re-inspection. Eligible claims are accepted and never queued: no accepted rewrites or uncertain claims, but potentially citation-only corrections. Prompt 23 gives their ids, the full answer, and sources. Only those ids may change. The ordinary verifier checks changes before filtering. A random-half variant limits eligibility. Unlike whole-answer self-check (Prompt 6), this intervention restricts which claims may change. Relative to retry, it changes scope, prompt, and the editor's inspection role together. Their separate contributions are not identified.

Scope-only cached replay. A replay sanity check requests a round-1 check of every claim, with the same checker, source-isolated prompt, editor, and final policy. On 100 ⚖ Legal and 100 🗎 Documents Qwen3.8-27B-FP8 items, all 200 final answers match the narrow arm and no never-queued claim is newly flagged. Every call is a cache hit: unchanged claim–source inputs retrieve their original verdicts. Cache reuse explains the identical outcomes. A fresh full-answer check could give different results. The wider request invokes 623 versus 518 logical verifier/editor calls on ⚖ Legal and 609 versus 554 on 🗎 Documents. Estimated input/output tokens are 318,747 versus 274,193 and 875,103 versus 808,324. These totals describe the requested work, all served from cache. A fresh full-answer control with the same checker and prompt remains untested. Re-inspection combines changes to scope, prompt, and role, leaving their individual contributions unresolved.

Published-method baselines. All are adapted to the same evidence pool and output schema. Self-Refine alternates feedback and refinement. Chain-of-Verification generates questions, answers them independently, and revises (Madaan et al., 2023; Dhuliawala et al., 2024). RARR runs attribution and editing. RARR+verifier substitutes the frozen verifier's queue and reasons. VeriCite runs utility judgment, per-source extraction, verification, and synthesis. A separate delete/regenerate ablation removes failed draft claims before regeneration. ReAct receives the frozen verifier as a tool (Yao et al., 2023). Full redraft is CRITIC-style, not a claim of exact reproduction. Verifier-selected best-of-three samples three candidate drafts and chooses the one with highest verifier support. Exact local prompts are in Appendix I. We do not present trained or probability-level methods as prompt-only reproductions.

How Table 2 groups the comparisons. Reference baselines include the draft and five published methods. Self-check is our generic prompt-defined control: the generator revises its whole draft using all sources, without a separate verifier or claim queue (Prompt 6). Self-check and self-check +

loop are complete strategies. The shared-prefix ladder separately isolates operations within selective correction. Selector and editor interventions test remaining repair opportunities. The grouping records provenance. It makes no claim of algorithmic novelty or ownership of reused methods.

Baseline budgets and termination. Self-Refine permits at most two feedback–rewrite rounds. It stops early when feedback contains `NO ISSUES` in its first 200 characters or the feedback call abstains. ReAct permits twelve action proposals, including invalid or unparsed proposals. Its actions are verify, read, edit, drop, answer, and finish. Finish or an abstained proposal ends the loop. Exhausting the budget returns the current answer. A verify action may invoke the verifier, so twelve proposals do not mean twelve total model calls. A draft with no claims does not enter either baseline loop. These settings are recoverable from the implementation and consistent with the saved round logs. Unavailable commercial-backend settings remain unrecorded.

A.3 Models, data, and coverage

Opus 5 (Anthropic, 2026a) and Sonnet 5 (Anthropic, 2026b) are the commercial models used for generation, verification, and editing. Qwen3.8-27B-FP8 (Qwen Team, 2026b) and Gemma4-26B-A4B-FP8 (Gemma Team et al., 2026) are self-hosted in non-thinking mode. Qwen3.5-9B (Qwen Team, 2026a) is the small editor. Gemma4-12B is held back for the confirmation study. GPT-5.6 Sol (OpenAI, 2026) is J1 and also supplies the legal-answer and lineage judgments, but never generates the task answer. The open second judge (J2) uses the same Gemma4-26B-A4B-FP8 checkpoint as the Gemma generator and in-loop verifier. J2 is therefore not generator-independent for the Gemma cells, which are excluded from judge-independent J2 ranges.

Open models. Serving uses vLLM 0.26.0 (Kwon et al., 2023), temperature 0, a 4,096-token output limit, and non-thinking mode. Best-of-three draft sampling instead uses temperature 0.7. The server allows 65,536 context tokens. Document images share a 1,280-pixel maximum side and a twenty-page limit. Table 4 summarizes the model roles. Exact checkpoints and run settings are provided with the code and supplementary data. The Gemma4-12B server additionally uses a Transformers 5.10.4 overlay (Wolf et al., 2020).

Table 4: **Open models and their roles.** The second judge (J2) shares the main Gemma checkpoint.

Model	Role
3.8-27B	Main generator, checker, editor
4-26B	Main generator, checker, editor, J2
3.5-9B	Swapped editor
4-12B	Held-back confirmation model

Commercial models. GPT-5.6 Sol uses medium reasoning effort. Claude calls retain provider-supplied system instructions alongside our task instructions. Commercial sampling controls are not matched to the open-model settings, so temperature zero is not assumed. Backend snapshots, provider system prompts, and effective output limits are unavailable. Claude thinking settings are unrecorded. The released code documents model access and call settings.

Sampling and expansion. The initial ⚖ Legal/📄 Documents split contains 50 development and 250 test items. Legal sampling is stratified by component dataset. Documents are stratified by source and page-count bin, initially with at most two questions per document. The test pool was later extended to 1,000 by retaining all 250 original test items and adding 750 eligible items. The extension permits up to five document questions across the retained development/test and added samples. Thus, the main 250-item block is the original frozen sample, not a selection based on the 1,000-item results. Saved manifests verify test inclusion and no development/test item-id overlap. Source-cluster disjointness is separately enforced for the confirmation study, not implied by item-id separation.

🌐 Open-domain QA uses 42 development and 850 test items. Intervention blocks use 100. The eighteen ladder configurations share nine task–generator draft pools, and some sources recur across pools. They are not independent replications. Tables report actual endpoint coverage. Draft-call failures are excluded consistently. Some Gemma4-26B-A4B-FP8 ReAct actions have no visible text (22 open-domain, 11 legal, 10 document runs). These are unparsed actions, not successful corrections.

B Evaluation and Statistical Specification

B.1 What each metric counts

Held-out support. For each answer, divide final claims labeled supported by the number of final claims, then average across valid answers. An empty final claim set has undefined support. The baseline, ladder, and descriptive action contrasts exclude an item when either endpoint is undefined. Confirmation policy tests instead subtract each action's marginal mean over its own defined answers, then average configurations equally within each task and tasks equally. These are different estimands when inclusion differs (Appendix F.2). Empty answers receive zero support only in the stated sensitivity analysis. Abstained judge calls are handled separately by the audit pipeline and bound to the judged content in the provenance manifest.

Unsupported mass. Count unsupported claims still present in the final answer relative to the audited draft claim count. This is not one minus final support because its denominator is different. Claims already deleted are not errors remaining in the answer. Deletion loss is a separate outcome (Appendix E.1).

Preservation. Kept supported content uses initially supported draft claims as its denominator and is undefined if there are none. Exact normalized matches in the final numbered claims survive. Otherwise, the lineage judge checks whether the complete rendered answer still asserts the content (Prompt 20). Ids are lookup keys, not evidence of survival. A missing lenient judgment falls back to token overlap (Jaccard ≥ 0.5 or intersection over the smaller token set ≥ 0.8). The separate same-scope judge (Prompt 24) rejects narrowing, qualification, broadening, and changes of subject. The reported same-scope retention uses *all draft claims*, unlike kept supported content. Incomplete strict judgments leave that endpoint undefined. Neither denominator should be silently substituted for the other. Newly rescued initially unsupported claims are counted separately from kept supported content.

Which answer fields are evaluated? Support checks each final numbered claim against its citation. It excludes the `ISSUE`, `ANSWER`, and citation-reason fields. Deterministic filtering removes claims without rewriting the `ANSWER` line, but a generative editor can update that line. Thus, full claim support does not certify every output field. The rendered answer contains the issue, numbered claims with citations, and answer line, but not citation reasons. Legal correctness and lineage consume that rendering. Document ANLS uses only the answer line. Document gold-span coverage and open-domain short-answer recall use the answer line plus claim texts. Undefined endpoints are omitted separately before each paired comparison.

Answer quality and coverage. ⚖ Legal correctness thresholds the GANDR rubric at score ≥ 4 out of 5. The continuous quality endpoint is the same score divided by five, not an independently judged completeness field. 🗎 Documents correctness uses ANLS ≥ 0.5 on the short answer (Biten et al., 2019). A lenient check accepts a verbatim gold span, and coverage records gold-span presence. 🌐 Open-domain QA coverage uses ASQA short-answer recall. Correctness thresholds recall at 0.5. The supplementary data label the task-specific continuous/coverage endpoint *completeness*, with the task definitions given here. The prespecified policy uses a different mapping: its ⚖ Legal completeness guard duplicates thresholded correctness, rather than using the continuous score, and its 🗎 Documents correctness guard uses the lenient check. These are not additional independent measurements. None is interchangeable with source support, and none alone measures all useful information rescued by a rewrite.

B.2 Pairing, clustering, and uncertainty

Contrasts preserve within-item pairing. Bootstrap units are contracts, documents, or open-domain questions. The ladder export uses 1,000 resamples, while the other standard result tables use 10,000. Pre-action ratios use 10,000 source-cluster resamples. These intervals condition on the saved model outputs. We do not have independent run-level repetitions or establish statistical equivalence. A confidence interval containing zero does not establish that the effect is practically negligible.

Confirmation tests use 2,000 synchronized source-cluster draws and Holm correction over the two prespecified comparisons. One draw per task is applied jointly across its configurations and actions, as specified in the protocol. This corrects the original implementation's independent draws by

configuration without changing the marginal-mean estimand or policy choices. Configuration-level observations are not independent evidence of transport to arbitrary future models (Appendix F.2).

The added descriptive action intervals use 10,000 NumPy bootstrap draws, synchronized across all four models and actions within each task. Each contrast uses jointly defined pairs for its endpoint. The point estimate is the mean paired difference, with 2.5th and 97.5th percentiles as interval endpoints. These intervals are not simultaneous, do not include annotation uncertainty, and do not replace the prespecified policy tests or recompute guard decisions.

B.3 Operational cost

Tokens include generation and verification roles, input and output. Held-out evaluation is excluded from method cost. Action-study incremental tokens are above the shared state and exclude the calibration examples and action trials. Monetary estimates use a common tokenization estimate and a fixed provider-price schedule. Commercial models were accessed by subscription and open models were self-hosted, so dollar columns are API-equivalent estimates, not measured marginal expenditure. Gemma4-12B uses a Gemma 3 12B proxy rate. Tokens are the primary cross-method cost measure.

B.4 Checking the measurement on existing records

Retention paths. We trace each initially supported draft claim through exact-text matching, semantic lineage, or token-overlap fallback. Table 5 decomposes the four ⚖ Legal EDIT-minus-DELETE gains. There is no token fallback in these comparisons. Most of the gain depends on semantic lineage. The broad retention audit does not directly sample DELETE-versus-EDIT pairs, so it does not validate these small differences. Complete label coverage rules out fallback as their explanation. Semantic-judge bias remains possible.

Table 5: **Which judgments contribute to the retention gain?** EDIT minus DELETE on ⚖ Legal. Entries are percentage-point contributions to the paired mean. Exact-text, semantic-lineage, and token-fallback contributions sum to the total before independent rounding. Paths identify the scoring evidence, not independent causal effects. All supported-draft decisions have complete lineage coverage, so the no-fallback restriction leaves every pair unchanged.

Model	n	Total	Exact	Semantic	Fallback
3.8-27B	246	+1.27	+0.00	+1.27	+0.00
Opus 5	249	+2.15	+0.00	+2.15	+0.00
Sonnet 5	243	+1.44	+0.00	+1.44	+0.00
4-26B	247	+1.55	+0.13	+1.41	+0.00

Across all eighteen configurations, only two 🌐 Open-domain QA first-edit comparisons contain fallback decisions. Excluding the affected answer reduces their estimated retention gains from 1.03 to 0.79 points with Qwen3.8-27B-FP8, and from 1.62 to 1.38 with the Qwen3.5-9B editor. Full point estimates and cluster intervals accompany the data. All 24 confirmation action–configuration records have complete lineage labels for supported draft claims. Excluding fallback therefore changes none of their loss-guard decisions. This does not validate the judges or the 3-point threshold.

What filtering changes. The filtering function leaves the `ANSWER` field unchanged when it removes numbered claims. In the 3,000 matched ladder observations, 2,999 stored STOP answer lines match S_1 exactly. One Opus ⚖ Legal record has identical numbered claims but a different answer line: the first-edit log rejects its replacement in S_1 and accepts it in STOP/RETRY. The 994-character replacement exceeds the current parser's 600-character guard. The S_1 answer retains 573 characters. The discrepancy arises in stored short-answer parsing, before filtering. We preserve it and report a sensitivity excluding that item: STOP-minus-RE-CHECK support remains $+0.30$ points, and RETRY-minus-STOP remains $+0.07$. Correctness differences are zero. Normalized answer-quality changes are $+0.08$ points for both contrasts. The primary records and complete endpoint-specific sensitivity remain available.

Stopping removes 188 S_1 claims from the numbered lists. None appears as a complete normalized text span in the final answer line. This lexical check does *not* detect paraphrases, entailment, or missing conditions. It therefore cannot establish that the removed information is absent from the answer. The saved audits do not supply a separate source-support label for every `ANSWER` line. Thus neither claim support nor full-rendering lineage alone certifies the complete user-visible answer. Establishing whole-answer support requires a separate semantic audit.

C Evaluator Validation

C.1 Sample and procedure

The support packet contains 300 claim–source units stratified by in-loop/J1 verdicts and draft versus rewritten claims. The retention packet has 150 draft-claim/final-answer units, and the legal packet has 50 answers. Each packet was labeled by two annotators with document-analysis experience, working independently and blind to the method, model, and model verdicts. Document units include page images. Support judgments use the displayed source alone, without outside knowledge. Full support requires every part of a claim to follow from the source. Partial support includes overstatement or a missing condition.

For retention, annotators compare each element of the original claim with the whole final answer. They choose among same scope, narrowed, widened, different point, no longer asserted, and cannot judge. A separate question asks whether information needed to answer the question was lost. Exact instructions accompany the supplementary data. We retain both independently returned labels without adjudication. Design-weighted estimates use inverse sampling probabilities and 2,000 question-cluster bootstrap draws, keeping both labels together. The 150 retention units span 134 question clusters.

Table 6: **Agreement depends on the endpoint.** Agreement rates are percentages. Weighted estimates include 95% question-cluster intervals. Strict same-scope retention agrees less with J1 than claim support does.

Quantity	Estimate
Inter-annotator support agreement, raw	84.3%
Inter-annotator support κ, four-way / binary	0.67 / 0.70
J1–human support agreement, weighted	93.2 [90.2, 95.5]
J2–human support agreement, weighted	89.1 [83.8, 93.4]
Lenient retention agreement with J1	92.6 [88.0, 95.9]
Strict same-scope agreement with J1	75.8 [63.4, 87.5]
Inter-annotator retention κ, content / scope	0.97 / 0.69
Inter-annotator retention agreement, raw, content / scope	98.7 / 86.7
Scope κ among 106 units both judge retained	0.64
Relation κ / needed-information loss κ	0.71 / 1.00
Legal correctness agreement with J1	86.0 [78.0, 93.9]

Retention disagreement varies by task. Among claims J1 labels as lost supported content, humans find the original point still asserted in 20/30 labels on both ⚖ Legal and 🌐 Open-domain QA, versus 9/30 on 📄 Documents. Among claims J1 labels as narrowed, humans judge 18/30 labels same scope on 📄 Documents, versus 5/30 on ⚖ Legal. Each count pools two labels per sampled unit. These stratum-specific disagreements locate weaknesses in the judge's retention decisions without estimating a corpus-wide loss rate. Complete counts accompany the data.

Table 7: **Human support labels by sampling stratum.** Entries are yes / partly / no labels, pooled over two annotators. V denotes the in-loop verifier. Counts are labels, not independent claims. The five strata contain 300 claims and 600 labels in total.

Sampling stratum	Open-domain QA	Legal	Documents
V accepts, J1 accepts	59/1/0	59/1/0	57/3/0
V rejects, J1 rejects	10/30/0	14/17/9	4/3/33
V accepts, J1 rejects	25/25/0	28/20/2	19/24/7
V rejects, J1 accepts	16/3/1	16/4/0	19/1/0
Rewritten by retry/re-inspect	28/2/0	28/2/0	24/3/3

C.2 Consequential disagreements

Of 320 labels on J1-accepted claims, 304 are yes, 15 partly, and one no. The disputed unit is an open-domain claim that the other annotator labels supported. Of 280 labels on J1-rejected claims, 102 are yes, 124 partly, and 54 no. For the particular stratum accepted in-loop and rejected by J1, labels are 28/20/2 on legal, 19/24/7 on documents, and 25/25/0 on open-domain, with 25 claims and 50 labels per task.

These distributions limit any interpretation of all J1-rejected claims as clear hallucinations. They also do not prove that judge bias cancels between methods: different editing styles can move different numbers of claims across a threshold. The trace establishes what the loop revisits under the chosen judgments. The human audit checks how often those judgments align with readers. Establishing human-based method rankings would require a separate, complete re-evaluation.

Which human labels cover the action state? Joining by configuration, item, claim text, and citation matches 31 of the 300 broad support units to the traced S_1 claims: one inside the next queue and thirty outside. Of the rest, 218 lie outside the eighteen 100-answer trace cohorts. The other 51 belong to those cohorts but are absent or changed at S_1. The original weights refer to selected draft/final occurrences, not a complete S_1 sampling frame, and cannot yield a human-defined outside-queue population estimate. The missed-error census (Appendix C.4) matches every never-selected error claim to S_1, but selects only those claims. It supports local improvement rates and, together with the complete queue counts, the conservative bounds below.

C.3 A finite-cohort bound from existing human labels

Consider a fixed S_1 cohort and the criterion that both annotators label a claim partly or not supported. Let K be the number of observed failures outside Q_2, and $M = |Q_2|$ count *all* queued claims. If O and I are the outside and inside failure counts under this same criterion, then $O \geq K$ and $I \leq M$, so

$$\frac{O}{O+I} \geq \frac{K}{K+M}. \tag{1}$$

Unlabelled outside failures can only raise the fraction. Treating every queued claim as a failure makes the bound conservative, without requiring representative sampling of the outside claims.

All 511 annotated before versions match the traced claim text and citation, and none belongs to Q_2. In the Qwen3.8-27B-FP8 and Opus 100-answer Legal cohorts, both annotators identify eighteen and eight outside failures against complete queues of four and three claims, giving bounds of 81.8% and 72.7%. Table 8 gives all eighteen mechanism traces with the any-annotator counts. The bounds condition on fixed cohorts and returned labels. They do not correct for measurement error or estimate prevalence in other tasks, and are distinct from confidence intervals. Partial support remains part of the criterion. The bound establishes overlooked support gaps, while the before/after audit separately tests whether interventions improve them.

Table 8: **A conservative human bound on problems outside the next list, per mechanism configuration.** Each row is a fixed 100-answer trace at S_1. Missed: never-selected J1 errors. K: those that both annotators label partly or not supported (Any: at least one annotator). M: every claim in Q_2, labelled or not. The last column is $K/(K+M)$ under the both-annotator criterion: even if all queued claims fail, at least this fraction of failures lies outside. Finite-cohort bounds conditional on the returned labels, not population estimates or counts of unambiguously false facts.

Configuration	Missed	K (both)	Any	M	Lower bound
Legal					
Opus 5	44	8	20	3	72.7%
4-26B	15	8	10	1	88.9%
3.8-27B	33	18	23	4	81.8%
3.8-27B→Opus 5	33	18	23	4	81.8%
3.8-27B→Qwen3.5-9B	33	18	23	10	64.3%
3.8-27B→Sonnet 5	33	18	23	9	66.7%
Sonnet 5	40	23	31	6	79.3%
Documents					
Opus 5	8	2	5	1	66.7%
4-26B	11	4	7	22	15.4%
3.8-27B	15	3	8	3	50.0%
3.8-27B→Opus 5	15	3	8	1	75.0%
3.8-27B→Qwen3.5-9B	15	3	8	5	37.5%
3.8-27B→Sonnet 5	15	3	8	2	60.0%
Sonnet 5	18	5	14	2	71.4%
Open-domain QA					
3.8-27B	32	15	21	6	71.4%
3.8-27B→Opus 5	32	15	21	2	88.2%
3.8-27B→Qwen3.5-9B	32	15	21	18	45.5%
3.8-27B→Sonnet 5	32	15	21	16	48.4%

C.4 Human audit of claims outside the edit list

The broad sample may underrepresent the exact claims behind a mechanism finding. We therefore audit all claims rejected by J1 but never selected for editing in the 28 configurations with a re-inspection arm: the eighteen 100-answer mechanism traces, the eight source-disjoint confirmation cells (273 Legal and 143 Documents answers), and the remaining Qwen3.8-27B-FP8 self-editor test answers (220 Legal, 140 Documents). This census contains 511 distinct claims and 973 configuration-level outcomes, because configurations that share a Qwen3.8-27B-FP8 draft share their before versions. Coverage is restricted to this J1-defined set. Human-defined errors elsewhere remain outside the census. Each annotator independently labels every before version and every distinct after version, and compares each distinct change or removal in randomized A/B order. For Documents, the cited page image accompanies its OCR text. Accepted claims never queued and unchanged by every editor of the draft pool serve as controls, drawn in two stages by source cluster. Sampling targets one control per distinct before version, subject to the eligible pool, and yields 473 controls. The support guide decides fully versus partly element by element: a claim is fully supported only if each of its elements is stated in or follows directly from the source. Paraphrase or omission that leaves the assertion unchanged never lowers the label. The same two annotators complete the broad retention and claim-transition evaluations, working independently and blind, with no adjudication or discussion. Per annotator, the packet holds 1,408 support units, 474 pairs, and 971 retention units.

Table 9 counts an improvement only when a claim is judged partly or not supported before editing and fully supported after editing. Rates average the two annotators' indicators for each outcome. The improvement rate is 22.4% [18.9, 26.0] over all 973 outcomes, 27.3% on Legal, 24.6% on Open-domain QA, and 8.8% on Documents. Retaining the point while allowing narrower scope gives 14.9% [12.1, 17.8] of all outcomes. Restricting further to the original scope gives 3.8% [2.6, 5.2], with the same denominator. Thus, some overlooked problems can be repaired, but fewer repairs keep their original scope. In 0.7% of all outcomes, a claim judged fully supported before editing is retained but loses full support. In 1.4% [0.7, 2.2], a claim judged fully supported before editing is removed. Needed answer information is lost in 6.5% [4.8, 8.2] of all outcomes.

Table 9: **Re-inspection can improve support without preserving scope.** Two annotators independently judge all claims rejected by J1 but never selected for editing in the 28 re-inspection configurations (973 outcomes from 511 distinct claims). There are 473 accepted controls, with design-weighted rates. Other percentages average the annotators' indicators over all outcomes in the column, except the explicitly marked subset of J1-credited repairs. The 95% intervals use 10,000 source-cluster draws over contracts, documents, or questions. Improvement requires a partly or not supported before label and a fully supported after label. Removed claims are never repairs. The same-scope and narrowing rows are nested subsets with the same denominator. Shared before versions are not independent samples.

Outcome, % [95% CI]	All	⚖ Legal	Documents	Open-domain QA
n (cell-claim rows; clusters)	973 (250)	602 (158)	243 (65)	128 (27)
Fully supported before	39.1 [34.0, 44.3]	40.2 [33.8, 46.9]	33.7 [24.5, 44.0]	43.8 [28.6, 60.7]
Accepted controls, fully before	93.2 [90.1, 95.8]	90.5 [85.1, 94.7]	95.9 [92.9, 98.3]	98.0 [93.6, 100.0]
Fully supported after	59.4 [55.0, 63.6]	65.0 [60.1, 69.8]	41.8 [33.3, 51.0]	66.4 [56.4, 76.7]
Not fully supported before, fully after	22.4 [18.9, 26.0]	27.3 [22.8, 32.0]	8.8 [4.9, 13.3]	24.6 [15.1, 34.0]
Also retains the same scope	3.8 [2.6, 5.2]	3.8 [2.3, 5.6]	1.4 [0.4, 2.8]	8.2 [3.3, 14.5]
Also retained, allowing narrowing	14.9 [12.1, 17.8]	20.5 [16.7, 24.5]	2.7 [0.9, 4.9]	11.3 [5.5, 18.0]
Supported before, retained but not fully after	0.7 [0.3, 1.1]	0.9 [0.4, 1.5]	0.2 [0.0, 0.7]	0.4 [0.0, 1.2]
Fully supported before, removed after	1.4 [0.7, 2.2]	1.7 [0.6, 2.9]	0.6 [0.0, 1.6]	1.6 [0.0, 4.0]
Needed answer information lost	6.5 [4.8, 8.2]	7.0 [4.9, 9.3]	3.9 [1.7, 6.3]	9.0 [3.2, 15.4]
Content no longer asserted (any reason)	13.4 [10.9, 16.2]	12.3 [9.3, 15.5]	13.8 [8.4, 20.0]	18.0 [9.9, 26.2]
J1 credited repairs, n	360	281	32	47
Fully supported after	90.3 [87.2, 93.1]	91.3 [87.8, 94.5]	78.1 [68.1, 88.9]	92.6 [86.0, 98.6]
Support κ (four-way; fully vs. rest)	0.69; 0.72	0.75	0.55	0.67
Paired outcome κ (all; same scope vs. rest)	0.68; 0.68	–	–	–

Before versions and reasons. Across all 973 outcomes, annotators judge 39.1% [34.0, 44.3] fully supported before editing, against 93.2% [90.1, 95.8] of the accepted controls. The share is 40.2% on Legal, 43.8% on Open-domain QA, and 33.7% on Documents. Both annotators label 38 of the 156 distinct Documents before versions not supported at all. Among the 511 distinct before versions, missing conditions account for fewer than half of the reasons for withholding full support. Annotator A identifies 141 missing conditions, 95 inferences that go beyond the source, and 89 wrong facts, and cannot judge nine claims. Annotator B gives 120, 86, and 90 such reasons, plus one wrong citation. J1's own verdicts on these claims are 385 PARTIALLY_VERIFIED, 48 FALSE, 41 INCONSISTENT, 24 VERIFIED_NEARBY, and 13 NOT_VERIFIED.

After-edit validity. Among J1's 360 credited repairs, annotators label 90.3% [87.2, 93.1] of after versions fully supported (78.1% on Documents). This measure includes cases already judged supported before editing. Counting a repair additionally requires a human-recognized problem in the before version. Table 10 gives every configuration. On fixed Qwen3.8-27B-FP8 Legal drafts, the Opus editor improves 37–42% of missed claims, Sonnet 21–27%, and Qwen3.5-9B 19–26%, while same-scope improvement stays at or below 8% for every editor. For Gemma4-12B on Documents, only 8.2% of missed claims are read as fully supported before editing and 2.2% improve, consistent with a repair-limited configuration.

Table 10: **Human-recognized improvements across configurations.** Percentages use all claims rejected by J1 but never selected in each configuration, averaging both annotators. Before: fully supported before editing. Improve: partly or not supported before and fully supported after. Same scope and Narrowing: nested subsets of Improve that retain the point at the same or a narrower scope, with the same denominator. Loss: needed answer information lost, not a subset of Improve. Arrows mark editor swaps on fixed Qwen3.8-27B-FP8 drafts. Mechanism cells hold 100 answers, extension cells the remaining Qwen3.8-27B-FP8 self-editor test items, and confirmation cells 273 Legal or 143 Documents answers. Different drafts and sources prevent reading rows as a paired editor ranking. Intervals accompany the data.

Configuration	Block	n	Before	Improve	Same scope	Narrowing	Loss
Legal							
Opus 5	mechanism	44	68.2	18.2	6.8	13.6	12.5
4-26B	mechanism	15	40.0	33.3	6.7	23.3	0.0
3.8-27B	mechanism	33	37.9	33.3	6.1	27.3	10.6
3.8-27B→Opus 5	mechanism	33	37.9	42.4	4.5	30.3	3.0
3.8-27B→Qwen3.5-9B	mechanism	33	37.9	25.8	0.0	22.7	13.6
3.8-27B→Sonnet 5	mechanism	33	37.9	27.3	3.0	16.7	1.5
Sonnet 5	mechanism	40	32.5	31.2	6.2	16.2	12.5
3.8-27B	extension	71	41.5	23.9	4.9	22.5	7.0
Gemma4-12B	confirmation	30	43.3	31.7	8.3	25.0	6.7
3.8-27B→Opus 5	confirmation	90	37.2	37.2	3.3	27.8	2.8
3.8-27B→Qwen3.5-9B	confirmation	90	37.2	19.4	2.2	15.0	11.1
3.8-27B→Sonnet 5	confirmation	90	37.2	21.1	1.1	15.0	2.8
Documents							
Opus 5	mechanism	8	56.2	12.5	0.0	6.2	6.2
4-26B	mechanism	11	50.0	18.2	0.0	0.0	4.5
3.8-27B	mechanism	15	53.3	0.0	0.0	0.0	6.7
3.8-27B→Opus 5	mechanism	15	53.3	23.3	0.0	6.7	0.0
3.8-27B→Qwen3.5-9B	mechanism	15	53.3	0.0	0.0	0.0	6.7
3.8-27B→Sonnet 5	mechanism	15	53.3	23.3	10.0	16.7	0.0
Sonnet 5	mechanism	18	44.4	5.6	2.8	5.6	0.0
3.8-27B	extension	16	34.4	6.2	3.1	3.1	0.0
Gemma4-12B	confirmation	67	8.2	2.2	0.0	0.0	6.7
3.8-27B→Opus 5	confirmation	21	33.3	21.4	2.4	2.4	0.0
3.8-27B→Qwen3.5-9B	confirmation	21	33.3	2.4	2.4	2.4	9.5
3.8-27B→Sonnet 5	confirmation	21	33.3	14.3	0.0	0.0	0.0
Open-domain QA							
3.8-27B	mechanism	32	43.8	26.6	9.4	12.5	4.7
3.8-27B→Opus 5	mechanism	32	43.8	37.5	15.6	18.8	12.5
3.8-27B→Qwen3.5-9B	mechanism	32	43.8	4.7	0.0	0.0	9.4
3.8-27B→Sonnet 5	mechanism	32	43.8	29.7	7.8	14.1	9.4

Agreement. On 1,408 support units, agreement is 86.2% raw, $\kappa = 0.69$ four-way and 0.72 for fully versus the rest (0.75 on ⚖ Legal, 0.55 on 📄 Documents, 0.67 on 🌐 Open-domain QA). On 474 paired comparisons, outcome agreement is $\kappa = 0.68$, as is agreement on same scope versus any change. For the joint repair-and-scope endpoint, annotators credit 40 and 34 of the 973 outcomes, with fourteen in common ($\kappa = 0.35$). Agreement on which outcomes satisfy both requirements is lower than agreement on scope alone. These outcomes can share claims and are not independent trials. The annotators' relative strictness differs by task. The missed-unit and control subsets have κ 0.68 and 0.52, respectively. About 94% of control labels are fully supported, which raises chance agreement and lowers κ relative to raw agreement.

How reliable are declared content losses? The 971 retention units come from all eight confirmation cells under re-inspection, sampled within judge strata. Humans confirm 74.8% [67.3, 81.6] of the 163 judge-declared supported-content losses. In the same stratum, 6.9% lose answer-critical information. Loss confirmation ranges from 35.0% for the Opus editor on ⚖ Legal to 100% for the Opus and Sonnet editors on 📄 Documents (Table 11). Across the 701 judge-kept narrowed or rewritten units, two of the 1,402 human labels indicate content loss. Among 107 sampled deletions of claims rejected by J1, 31.4% [20.4, 42.5] lose needed answer information. Agreement is $\kappa = 0.73$ for the five-way outcome and 0.92 for lost versus kept. Scope agreement is $\kappa = 0.65$ among the 754 units that both annotators judge retained. The broad retention task gives 0.64 on its 106 jointly retained units (Appendix C). These stratum-specific observations do not establish a corpus loss rate.

Table 11: **How reliable are declared content losses?** Draft-claim/final-answer units from all eight confirmation cells under re-inspection, sampled within judge strata (design-weighted rates, 95% source-cluster intervals). Content gone: the annotators read the original content as no longer asserted or as losing needed answer information. Scope changed: still asserted but narrowed. Stratum rows are precisions within the sampled strata, not corpus loss rates.

Stratum	n	Content gone	Needed info lost	Scope changed
Judge-declared supported-content loss	163	74.8 [67.3, 81.6]	6.9 [3.9, 10.5]	14.7 [9.9, 20.2]
Judge-kept, narrowed	317	0.1 [0.0, 0.4]	0.1 [0.0, 0.4]	60.8 [54.9, 66.5]
Judge-kept, rewritten	384	0.1 [0.0, 0.2]	0.0 [0.0, 0.0]	10.6 [7.2, 14.6]
Deleted unsupported claim	107	78.2 [70.2, 86.4]	31.4 [20.4, 42.5]	13.1 [7.8, 18.3]
Judge-declared losses read as content gone, by confirmation cell				
📄 Documents, Gemma4-12B	25	80.0 [57.9, 93.1]	–	–
📄 Documents, 3.8-27B→ Opus 5	4	100.0 [100.0, 100.0]	–	–
📄 Documents, 3.8-27B→Qwen3.5-9B	11	59.1 [26.9, 91.7]	–	–
📄 Documents, 3.8-27B→ Sonnet 5	3	100.0 [100.0, 100.0]	–	–
⚖ Legal, Gemma4-12B	18	55.6 [28.6, 77.8]	–	–
⚖ Legal, 3.8-27B→ Opus 5	10	35.0 [10.0, 65.0]	–	–
⚖ Legal, 3.8-27B→Qwen3.5-9B	80	84.4 [75.6, 91.8]	–	–
⚖ Legal, 3.8-27B→ Sonnet 5	12	37.5 [12.5, 62.5]	–	–
Agreement: five-way κ / lost vs. kept / scope among both-retained	971	0.73 / 0.92 / 0.65	–	–

C.5 Human audit of judge-credited repairs

The missed-error census covers claims that never entered an edit queue. A separate stratified audit samples changes credited as repairs by J1 across selective continuations, published-method adaptations, and mechanical hedges. A credited repair has an unsupported draft claim and a supported final counterpart. Within each cohort, the sampling unit is a distinct before version with a fixed representative outcome chosen by condition, family, and item order. This avoids counting many copies of the same before version within that cohort. It does not sample all outputs or all changes uniformly.

The sample contains 153 selective-continuation pairs, 121 published-method pairs, and 52 hedge pairs. The latter includes all eight eligible 📄 Documents and fourteen 🌐 Open-domain QA units, below the planned allocation because the eligible pools are exhausted. Sampling is stratified by task and reason category. Two annotators judge before and after versions independently and blind, then compare

randomized A/B pairs for scope and needed answer information. They work without discussion or adjudication. Document units include the cited page image. The guide explicitly instructs them to assess the factual elements that remain asserted when a hedge is added. The annotation guide and per-annotator labels accompany the supplementary data.

Table 12: **Human checks of judge-credited repairs.** Every sampled pair is J1-unsupported in the draft and J1-supported afterward. Rates average two independent annotators and weight by inverse sampling probability. Intervals resample source clusters. Cohorts select distinct before versions with one representative outcome. These rates do not estimate the prevalence of errors in all generated answers.

Task	Units	Fully supported before	Fully supported after
Selective continuations			
⚖ Legal	50	37.0 [21.7, 54.1]	96.8 [92.4, 99.8]
🗎 Documents	53	4.9 [1.3, 9.3]	93.3 [86.4, 98.5]
🌐 Open-domain QA	50	12.6 [3.9, 23.4]	95.7 [89.0, 100.0]
Published-method adaptations			
⚖ Legal	50	38.1 [27.3, 49.2]	90.6 [83.3, 96.8]
🗎 Documents	39	5.1 [0.0, 12.5]	96.2 [89.8, 100.0]
🌐 Open-domain QA	32	29.5 [15.3, 45.8]	98.4 [94.8, 100.0]
Mechanical hedges			
⚖ Legal	30	40.3 [24.8, 57.2]	43.8 [29.4, 58.2]
🗎 Documents	8	43.8 [11.1, 85.7]	93.8 [78.6, 100.0]
🌐 Open-domain QA	14	64.3 [40.0, 85.7]	67.9 [46.2, 88.5]

High after-edit agreement does not establish a repair. Across the selective and published-method samples, weighted full support after editing is 95.7% and 97.1%, respectively. Yet 19.8% [12.1, 28.3] and 29.0% [17.4, 41.9] are already fully supported before editing. The corresponding before rates are about 37–38% on ⚖ Legal and 5% on 🗎 Documents, with different 🌐 Open-domain QA rates for the two cohorts. A single discount from the missed-error census would therefore be inappropriate. Full-support agreement is $\kappa = 0.78$ on 652 support units, four-way agreement is 0.73, and paired-outcome agreement is 0.63 on 326 pairs.

Narrowing and information loss are separate outcomes. Both annotators classify all 52 mechanical hedge pairs as narrower assertions. Weighted full support rises from 44.4% before to 52.1% after in that cohort. Some narrowing can restore support, so the scope finding does not imply that every hedge is an invalid correction. Needed answer information is lost in 72/306 selective-continuation judgments and 47/242 baseline judgments. These are unweighted counts of annotator judgments, not percentages of pairs on which at least one annotator reports loss. Design-weighted joint support, scope, and loss endpoints are provided separately in the data. They are conditional on the sampled credited-repair cohorts.

C.6 Sensitivity of credited counts to before-label disagreement

We use the human before labels for an exploratory sensitivity analysis of 238 configuration-level credited-repair counts. A Qwen3.8-27B-FP8 classifier assigns J1's rationales to the human reason categories, with agreement $\kappa = 0.75$ and 0.77 against the two annotators. Within each task and reason category, the analysis estimates the share p judged fully supported before editing and discounts a credited count by $1 - p$. Categories with fewer than twenty annotation records use the pooled reason-category rate, followed by a broader fallback. The input contains 837 before-label records from the missed-error census and the credited-repair audit.

This calculation asks how many credited transitions remain after an assumed before-label correction. It does not establish human-validated repair totals. Its transfer assumes that task and reason category capture the relevant differences between never-selected, queued, and baseline claims. The implementation averages annotation records within a stratum without design weighting or model-specific adjustment, and some before versions occur in more than one cohort. It leaves after-label error and

content loss uncorrected. Source-cluster intervals reflect uncertainty in the human records while holding the recorded credited counts fixed.

The prespecified analysis pools the cohorts. After inspecting the human results, a second analysis separates the hedge cohort because its selection and before-label distribution differ. Both versions are included in the supplementary data. Under the separated calculation, the first-edit-plus-check counts for the four main ⚖ Legal models retain roughly 47–71% of J1 credit, versus 81–87% on 🗎 Documents and 71–72% on 🌐 Open-domain QA. These are sensitivity estimates under the transfer assumptions. They do not replace the matched claim histories, the direct human rates, or the original support scores.

C.7 Applying the human support rule with the judge

To probe the criterion gap, GPT-5.6 Sol applies the annotators' element-by-element rule to the same 1,408 support units from the missed-error audit, with the question, cited source, and document image when applicable (Prompt 27). Agreement with the two annotators on full support is $\kappa = 0.62$ and 0.61, compared with 0.56 and 0.54 for the original J1 verdict. Among 511 distinct never-selected before versions, humans judge 38.3% fully supported and the human-rule judge 26.6%. The original J1 rate is zero by selection.

The direction and size of disagreement vary by task. On ⚖ Legal, the corresponding human and human-rule judge rates are about 43% and 23%. On 🗎 Documents they are 28% and 29%, and on 🌐 Open-domain QA 44% and 50%. The clearer rule reduces aggregate disagreement on this evaluation set without eliminating it. It also changes the judge's instructions and available question context together. This is a same-cohort measurement check, not independent validation of a newly calibrated judge. The original J1 remains the reported instrument for the machine comparisons.

C.8 Changing the support threshold

At S_1, 462 of 6,445 final claim records are PARTIALLY_VERIFIED, accounting for 79.5% of the 581 strict-rule errors present across all eighteen configurations. Partial support is therefore central to the quantity we measure. Table 13 relaxes the rule without changing an action, claim, citation, or raw verdict. Across the nine same-model traces, accepting partial support reduces present errors from 278 to 71. Of the remaining 71, 53 (74.6%) are outside the next queue, compared with 247/278 (88.8%) under the strict rule. Initially missed errors account for 38/71, a different quantity from the 53/71 outside the *next* list. Also accepting neighboring-source support leaves 45 errors, 34 outside (75.6%), and relaxes citation accuracy as well.

Table 13: **Fewer errors under a looser rule, often still outside the next list.** Descriptive claim counts across nine same-model 100-answer traces at S_1, excluding deletion. Missed means never selected. Outside counts all present errors outside Q_2 and gives their fraction of Present. These are different quantities. The last rule also relaxes citation location. Configurations share sources, and some have no residual errors after relaxation. The aggregate is not a per-configuration guarantee. Full counts and intervals accompany the data.

Accepted as supported	Present	Missed	Failed	New	Outside (%)
Full support only	278	216	56	6	247 (88.8%)
Also accept partial support	71	38	27	6	53 (74.6%)
Also accept neighboring-source support	45	31	10	4	34 (75.6%)

These are descriptive sums across configurations that can share sources and drafts. The aggregate does not imply a majority outside every nonempty queue: under the partial-accepted rule, Gemma4-26B-A4B-FP8 on 🗎 Documents has nine of 25 errors outside. Several other configurations have zero residual errors, making their outside fraction undefined. The size and composition of the counted population change with the rule. This sensitivity is not a replacement for human evaluation.

The 77 missed ⚖ Legal claims behind the 16/33 and 26/44 results carry 76 PARTIALLY_VERIFIED verdicts and one FALSE. Under the partial-accepted rule, no missed errors remain in the former configuration and one remains in the latter. Neither yields a missed-error repair. These results do

not support an equally large claim about unequivocally false statements. Source qualifications and citation support remain useful evaluation targets, but should be named explicitly. All sensitivity counts exclude deleted content from present errors, use the same per-item accounting as the main trace, and provide cell-level cluster intervals in the accompanying data.

C.9 Second judge on matched traces

J2 re-labels all five arms of all eighteen trace configurations, with 100 matched items per configuration. Queues, rewritten text, and operational verdicts remain fixed. In the sixteen configurations where J2 differs from the generator, 65.7–100% of present errors lie outside the next edit list, compared with 70–100% under J1 on the same items. Initially missed claims are the largest residual group in all sixteen under both judges. The two Gemma4-26B-A4B-FP8 configurations remain in the supplementary records as non-independent controls, not in these ranges.

Table 14: **Both judges credit some overlooked repairs, but disagree on how many.** Never-selected ⚖ Legal claims becoming fully supported after re-inspection, on matched 100-answer cohorts. Denominators are each judge's own initially missed claims, not an identical error set for paired repair-rate differences. Complete matched traces and intervals are supplied as data.

Generator and editor	J1	J2
Qwen3.8-27B-FP8	16/33 (48%)	7/25 (28%)
Opus 5	26/44 (59%)	20/47 (43%)

Both judges credit some re-inspection repairs on ⚖ Legal, but J2 credits fewer: 7/25 for Qwen3.8-27B-FP8 and 20/47 for Opus. Each denominator uses that judge's own missed-error set, so the fractions are not a paired comparison over identical error claims. Shared source-support rubrics and disagreement over partial support limit the independence of this check. Exact cell counts, denominators, intervals, and all 180 arm-level coverage records accompany the data. Every present-error total is also checked directly against the final claims, excluding deletion.

C.10 Second judge on action selection

The same open judge scores the broad support packet and confirmation outputs under fixed policy decisions. The two Gemma4-12B confirmation configurations share J2's family, though not its checkpoint. They do not provide validation from an independent family. J2 accepts six of the 55 labels marked no by a human, versus one for J1. These are labels, not necessarily 55 distinct claims. Re-inspection-minus-stop signs agree in seven of eight configurations, while the best feasible action changes in four. Under J2, the policy's support gain over always stopping is $+0.48$ [$+0.18$, $+0.77$]. This sensitivity analysis supplements the primary J1 result.

Table 15: Held-out support (%) under J1 and J2 on the same generated answers. Each mean uses the action's own answers with defined scores. n counts test items. A dagger marks a quality-guard violation. Bold best-action entries identify changes relative to J1.

Judge	**Stop**	**Retry**	**Re-inspect**	**Best feasible**
⚖ Legal / 9B editor, $n = 273$				
J1	92.03	91.99	92.30†	stop
J2	92.63	92.66	95.03†	**retry**
⚖ Legal / Sonnet editor, $n = 273$				
J1	92.39	92.31	94.27	re-inspect
J2	92.38	92.69	93.46	re-inspect

Continued on next page.

Table 15 continued.

Judge	Stop	Retry	Re-inspect	Best feasible
⚖ **Legal / Opus editor,** $n = 273$				
J1	92.63	92.63	96.62	re-inspect
J2	92.75	92.80	94.58	re-inspect
⚖ **Legal / Gemma-12B,** $n = 273$				
J1	96.21	96.00	97.20	re-inspect
J2	97.30	97.30	97.93	re-inspect
📄 **Documents / 9B editor,** $n = 143$				
J1	91.69	91.69	91.76†	stop
J2	90.56	90.56	91.42	**re-inspect**
📄 **Documents / Sonnet editor,** $n = 143$				
J1	91.75	91.75	91.42	stop
J2	90.45	90.45	91.35	**re-inspect**
📄 **Documents / Opus editor,** $n = 143$				
J1	91.86	91.92	93.75	re-inspect
J2	90.40	90.40	91.53	re-inspect
📄 **Documents / Gemma-12B,** $n = 143$				
J1	61.07	61.00	59.63†	stop
J2	59.44	60.32	59.07†	**retry**

Full in-loop/held-out cross-tabs and stratum-level retention results are retained in the supplementary data. They describe evaluator disagreement. Human-validated error rates for every method would require further annotation.

D Additional Experimental Results

This section answers three questions: how uncertain are the operation effects, how do the baselines trade support against preservation and cost, and how much does the verifier family matter? The supplementary data retain all endpoints and paired counts. The matched ladder and baseline comparisons use different item sets where noted. Their marginal means must not be silently mixed.

D.1 Every matched operation contrast

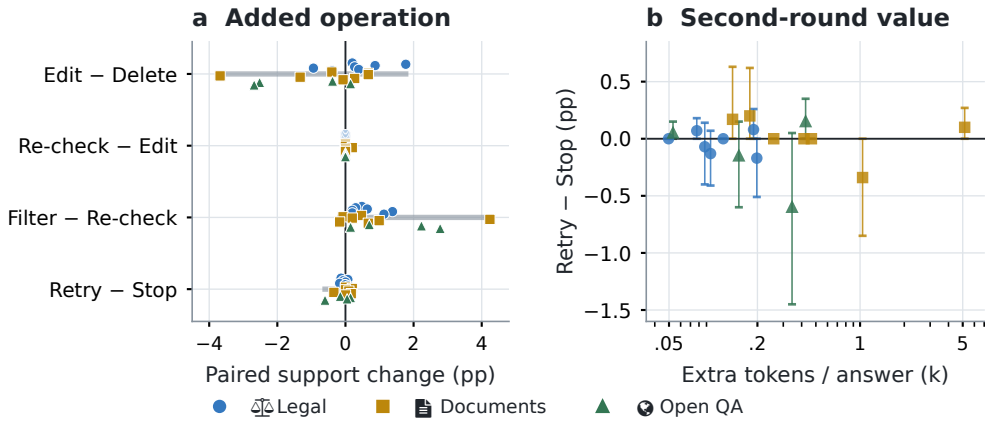

Figure 6: **A second round adds cost with small support changes.** Left: adding an edit instead of deleting flags, adding a check, filtering failures, and retrying versus stopping. Lines span point estimates, not intervals. Each point is a task–model configuration (18 comparisons per step). Right: all 18 retry-minus-stop support effects versus extra tokens, with 95% source-cluster intervals. The token axis is logarithmic. Here, k means thousands and pp means percentage points.

Table 16: Paired support changes for all 18 configurations (percentage points). Each entry gives the effect, its 95% source-cluster interval, and paired sample size. Intervals use 1,000 resamples. All 72 operation comparisons are included. Qwen-to-editor rows keep the Qwen generator and verifier fixed. All endpoints and arm-level outcomes are retained in the supplementary data.

Configuration	Edit minus delete	Re-check minus edit	Stop minus re-check	Retry minus stop
⚖ Legal				
Qwen	+0.20 [-0.44, +0.81] $n = 247$	+0.00 [+0.00, +0.00] $n = 249$	+0.48 [+0.03, +1.05] $n = 249$	-0.13 [-0.41, +0.07] $n = 249$
Opus	+1.77 [+0.72, +2.88] $n = 248$	+0.00 [+0.00, +0.00] $n = 250$	+0.30 [+0.07, +0.59] $n = 250$	+0.07 [+0.00, +0.18] $n = 250$
Sonnet	+0.87 [-0.56, +2.44] $n = 245$	+0.00 [+0.00, +0.00] $n = 250$	+0.64 [+0.07, +1.40] $n = 249$	-0.07 [-0.40, +0.14] $n = 249$
Gemma	+0.27 [-0.37, +0.93] $n = 246$	+0.00 [+0.00, +0.00] $n = 249$	+0.20 [+0.00, +0.63] $n = 249$	+0.00 [+0.00, +0.00] $n = 249$
Qwen → 9B	-0.94 [-3.13, +0.67] $n = 99$	+0.00 [+0.00, +0.00] $n = 99$	+1.38 [+0.08, +3.35] $n = 99$	-0.17 [-0.51, +0.00] $n = 99$
Qwen → Opus	+0.39 [-0.56, +1.46] $n = 99$	+0.00 [+0.00, +0.00] $n = 99$	+0.20 [+0.00, +0.64] $n = 99$	+0.00 [+0.00, +0.00] $n = 99$
Qwen → Sonnet	-0.39 [-1.60, +0.58] $n = 99$	+0.00 [+0.00, +0.00] $n = 100$	+1.13 [+0.24, +2.15] $n = 99$	+0.08 [+0.00, +0.26] $n = 99$
Documents				
Qwen	-0.40 [-1.05, +0.14] $n = 246$	+0.00 [+0.00, +0.00] $n = 250$	+0.47 [+0.00, +1.13] $n = 249$	+0.00 [+0.00, +0.00] $n = 249$
Opus	+0.67 [-0.07, +1.56] $n = 249$	+0.00 [+0.00, +0.00] $n = 250$	-0.07 [-0.58, +0.28] $n = 250$	+0.20 [+0.00, +0.62] $n = 250$
Sonnet	+0.67 [-0.24, +1.72] $n = 247$	+0.00 [+0.00, +0.00] $n = 250$	+0.20 [+0.00, +0.52] $n = 250$	+0.00 [+0.00, +0.00] $n = 250$
Gemma	-3.68 [-5.62, -1.67] $n = 236$	+0.20 [+0.00, +0.63] $n = 250$	+4.24 [+2.40, +6.00] $n = 241$	+0.10 [+0.00, +0.27] $n = 241$
Qwen → 9B	-1.33 [-3.80, +0.34] $n = 99$	+0.00 [+0.00, +0.00] $n = 100$	+0.99 [-0.16, +2.45] $n = 99$	-0.34 [-0.85, +0.00] $n = 99$
Qwen → Opus	+0.27 [-0.29, +0.86] $n = 99$	+0.00 [+0.00, +0.00] $n = 100$	-0.17 [-0.63, +0.00] $n = 100$	+0.17 [+0.00, +0.63] $n = 100$
Qwen → Sonnet	-0.07 [-0.70, +0.49] $n = 99$	+0.00 [+0.00, +0.00] $n = 100$	+0.67 [+0.00, +1.87] $n = 100$	+0.00 [+0.00, +0.00] $n = 100$

Continued on next page.

Table 16 continued.

Configuration	Edit minus delete	Re-check minus edit	Stop minus re-check	Retry minus stop
Open-domain QA				
Qwen	-0.38 [-1.38, +0.55] $n = 100$	+0.00 [+0.00, +0.00] $n = 100$	+0.70 [+0.15, +1.50] $n = 100$	-0.15 [-0.60, +0.15] $n = 100$
Qwen → 9B	-2.52 [-4.27, -0.92] $n = 100$	+0.00 [+0.00, +0.00] $n = 100$	+2.23 [+1.13, +3.47] $n = 100$	+0.15 [+0.00, +0.35] $n = 100$
Qwen → Opus	+0.15 [-0.98, +1.05] $n = 100$	+0.00 [+0.00, +0.00] $n = 100$	+0.15 [+0.00, +0.45] $n = 100$	+0.05 [+0.00, +0.15] $n = 100$
Qwen → Sonnet	-2.68 [-4.68, -0.80] $n = 100$	+0.00 [+0.00, +0.00] $n = 100$	+2.78 [+1.20, +4.63] $n = 100$	-0.60 [-1.45, +0.05] $n = 100$

D.2 RETRY VERSUS STOPPING ON ANSWER QUALITY

Table 17 summarizes all eighteen continuation configurations. The accompanying data retain each estimate, paired sample size, and interval. Sixteen correctness point estimates are zero, which does not imply identical answers: the ⚖ Legal Qwen3.8-27B-FP8–Sonnet contrast is 0.00 [-3.03, $+2.91$] points. The largest correctness gain is $+1.00$ [0.00, $+3.12$] with the 📄 Documents Qwen3.5-9B editor. The continuous quality/coverage endpoint ranges from -0.20 to $+0.50$ points, with all intervals including zero. For ⚖ Legal, this is normalized answer-quality score, not a separately measured completeness judgment.

Table 17: **Small average retry effects do not imply no rescued content.** Retry minus stopping across 18 configurations. Ranges contain point estimates, not confidence intervals. The third column counts paired 95% cluster intervals excluding zero. Quality/coverage means legal rubric score divided by five, document gold-span presence, or open-domain short-answer recall. These are task-specific endpoints, not a universal completeness score. Complete paired estimates, intervals, and exclusions are supplied as data.

Endpoint	Change (points)	CIs excluding 0	Paired n
Source support	-0.60 to +0.20	0/18	99–250
Answer correctness	-0.40 to +1.00	0/18	100–250
Answer quality/coverage	-0.20 to +0.50	0/18	100–250
Kept supported content	-0.26 to +0.33	0/18	96–249
Same-scope kept (all draft)	-0.50 to +0.57	0/18	100–250

Repairs that averages can hide. The 3,000 paired answer observations contain 10,066 S_1 claim records: 980 unsupported and 9,086 supported. Configurations share source items and sometimes drafts, so these totals are descriptive. Retry rescues 37 unsupported claims that stopping removes, and makes three supported claims unsupported. Stopping removes 188 claims in total, while retry removes 113. Of the 75 kept only by retry, 64 end supported and eleven unsupported. Thirty were supported at S_1 and 45 were not. The 64 supported rescues, about 6.4 per 1,000 S_1 claims, include more than the 37 newly supported claims. This is why retention of initially supported content alone misses some value of retry.

Useful repairs can leave support unchanged. All five configurations with $[0, 0]$ support intervals have exactly zero differences before rounding on every jointly defined answer. Each nevertheless contains a text-changing J1 rescue that stopping deletes. Table 18 shows the final claim counts for those answers. Both arms have 100% support despite retaining different amounts of supported content. These rescues need not change retention of initially supported draft claims, because the rescued claims were initially unsupported. Shared questions across editor configurations are not independent examples.

Table 18: **Repairs hidden by zero support differences.** Each row is a configuration with a $[0, 0]$ interval. n is the number of valid support pairs. The last two columns describe one text-changing repair under J1 in that configuration, not its average answer. Stop/Retry claim counts differ despite identical support. The Qwen and Qwen-to-Sonnet document examples share a question.

Configuration	n	Claims Stop / Retry	Support (%) Stop / Retry
⚖ Legal / 4-26B	249	1 / 2	100 / 100
⚖ Legal / Qwen → Opus	99	4 / 5	100 / 100
📄 Documents / 3.8-27B	249	1 / 2	100 / 100
📄 Documents / Sonnet 5	250	3 / 4	100 / 100
📄 Documents / Qwen → Sonnet	100	1 / 2	100 / 100

Answer-bearing content. Source-based answer-bearing tags are available for 2,478 S_1 claims from 📄 Documents and 🌐 Open-domain QA, but not ⚖ Legal. Retry rescues eight tagged unsupported

claims, loses support on none, and removes 32 tagged claims versus 46 for stopping. Ten claims removed by stopping are both kept and supported under retry. These tags are task-specific proxies, not a complete measure of useful information in every answer.

Empty answers and undefined endpoints. Stopping produces sixteen empty final claim sets and retry fifteen across these paired blocks. The largest individual empty count is nine for Gemma4-26B-A4B-FP8 Documents. There are no judge abstentions. Assigning empty claim sets zero support instead of excluding undefined pairs changes the retry support range to -0.60 to $+0.33$ points, still with all intervals including zero. The data include endpoint-specific exclusions and the full sensitivity, rather than silently treating an empty answer as perfectly supported.

D.3 CONTINUATION ON THE FULL OPEN-DOMAIN BLOCK

The eighteen-configuration ladder retains its original cohorts so that operation effects and mechanism traces can be read together. Table 19 separately examines all four models on the existing 850-item Open-domain QA block. The paired changes remain small. In the recorded-prefix sensitivity, Opus gains $+0.17$ [$+0.06$, $+0.29$] points. The interval excludes zero, indicating a small positive effect on these saved outputs. This extends the continuation comparison to the larger block. The 100-answer claim-history analysis retains its original coverage.

Table 19: **Small continuation effects in the larger open-domain block.** Retry minus stop support points [95% question-cluster interval], using 850 saved runs per model. Full-cohort intervals reproduce the original 1,000-draw export. The same protocol is used for the recorded-prefix restriction defined below, which excludes eight Opus and ten Sonnet items. Gemma has one undefined support pair. These checks are separate from Figure 4's eighteen configurations and its 100-answer mechanism traces.

Model	Full n	Full cohort	Restr. n	Recorded-prefix match
3.8-27B	850	-0.02 [-0.12, +0.08]	850	-0.02 [-0.12, +0.08]
Opus 5	850	+0.09 [-0.05, +0.24]	842	+0.17 [+0.06, +0.29]
Sonnet 5	850	+0.10 [-0.13, +0.35]	840	+0.13 [-0.10, +0.35]
4-26B	849	+0.00 [-0.12, +0.12]	849	+0.00 [-0.12, +0.12]

Check matching beyond the edited text. Two Opus items and one Sonnet item have different first edits between stop and retry. Excluding only these gives Opus $+0.14$ [$+0.03$, $+0.28$]. The records also contain initial-check or re-check discrepancies without a changed first edit. The table therefore uses a stricter restriction: matching draft and first-edit text/citations, initial queue/check/action summaries, and re-check counts. It excludes eight Opus and ten Sonnet items in total. The audit covers these recorded fields. Unrecorded provider responses and other answer fields remain outside its scope.

The full-cohort outcomes remain in the left comparison and in the data. No records are corrected or discarded. The eighteen-configuration main ladder's open-domain cells use Qwen3.8-27B-FP8 drafts and are unaffected by these Opus/Sonnet discrepancies. Question ids, pair availability, separate match indicators, and both exclusion sensitivities accompany the data. The larger block should not be treated as perfectly matched without this qualification.

D.4 BASELINES: SUPPORT, PRESERVATION, AND COST

Each task has a complete table of seventeen methods and four models: 204 method–model–task results in total. Each cell reports support, kept supported content, tokens, and sample size. Confidence intervals, paired loop-minus-baseline effects, correctness, and other endpoints accompany the data. RARR's uncertain-kept variant remains visible even where it matches the base variant. No method is assumed to dominate on support, preservation, and cost simultaneously.

The main overview selects fourteen methods. The uncertain-kept RARR variant, RARR with re-check and filtering, and best-of-three are included here. Their omission from the overview does not imply weaker support. In particular, the high-support Gemma result below illustrates why we inspect the operations and denominators behind a method's mean.

Table 20: ⚖ Legal: every baseline, with support and preservation. Each entry gives support (kept supported content), in percent. The second line gives total operational tokens in thousands and the arm sample size. Every method is evaluated in every configuration. Methods use the fixed evidence pool. Intervals, paired sample sizes, correctness, and call counts are in the supplementary data. These marginal means are not paired contrasts.

Method	**Opus 5**	**Sonnet 5**	**3.8-27B**	**4-26B**
Draft only	73.3 (100.0) 2.1k; 250	78.0 (100.0) 2.0k; 250	83.6 (100.0) 2.1k; 250	90.6 (100.0) 1.9k; 250
Delete flags	85.4 (96.7) 3.9k; 250	84.0 (98.2) 3.1k; 250	90.8 (97.7) 3.7k; 250	93.6 (98.3) 3.0k; 250
Self-check	91.7 (98.8) 6.3k; 250	87.6 (99.0) 5.0k; 250	91.1 (94.3) 6.6k; 250	93.9 (99.4) 5.6k; 250
Self-Refine	82.3 (93.5) 10.2k; 250	84.4 (94.6) 6.9k; 250	84.8 (93.0) 6.3k; 250	92.2 (94.3) 8.6k; 250
CoVe	78.7 (99.1) 13.2k; 250	80.5 (99.2) 9.8k; 250	85.6 (98.9) 12.1k; 250	92.5 (98.4) 9.4k; 250
RARR	77.9 (99.9) 4.3k; 250	83.4 (99.6) 4.0k; 250	85.8 (99.6) 4.4k; 250	90.8 (100.0) 3.9k; 250
RARR + verifier	86.6 (99.6) 5.0k; 250	85.8 (99.9) 3.8k; 250	90.7 (98.8) 4.3k; 250	93.7 (99.8) 3.4k; 250
RARR + keep uncertain	86.6 (99.6) 5.0k; 250	85.8 (99.9) 3.8k; 250	90.7 (98.8) 4.3k; 250	93.7 (99.8) 3.4k; 250
RARR + re-check	86.9 (99.3) 5.3k; 250	85.9 (99.9) 3.8k; 250	91.2 (98.7) 4.4k; 250	93.9 (99.7) 3.5k; 250
VeriCite	87.4 (96.3) 8.7k; 250	82.4 (97.8) 6.8k; 250	91.0 (91.9) 8.0k; 250	93.5 (97.8) 6.8k; 250
Delete/regenerate	85.3 (98.9) 6.2k; 250	85.2 (97.8) 5.2k; 250	90.4 (97.7) 5.7k; 250	93.4 (99.1) 5.0k; 250
ReAct	86.8 (97.5) 20.1k; 250	81.4 (99.7) 8.8k; 250	89.5 (96.4) 16.8k; 250	90.9 (99.3) 15.3k; 250
Best of three	84.6 (98.3) 11.7k; 250	82.3 (99.3) 9.5k; 250	92.3 (99.7) 11.1k; 250	94.2 (99.5) 9.1k; 250
Edit once	87.3 (98.9) 4.9k; 250	84.8 (99.7) 3.7k; 250	91.0 (98.9) 4.4k; 250	94.0 (99.8) 3.5k; 250
Stop after filtering	87.6 (98.8) 5.5k; 250	85.3 (99.5) 3.9k; 250	91.5 (98.8) 4.5k; 250	94.1 (99.7) 3.5k; 250
Loop, two rounds	87.7 (98.9) 5.6k; 250	85.3 (99.5) 4.0k; 250	91.3 (99.0) 4.6k; 250	94.1 (99.7) 3.6k; 250
Self-check + loop	92.7 (98.2) 6.7k; 250	89.7 (98.8) 5.9k; 250	93.6 (96.0) 6.4k; 250	95.4 (99.3) 5.6k; 250

Table 21: Documents: every baseline, with support and preservation. Each entry gives support (kept supported content), in percent. The second line gives total operational tokens in thousands and the arm sample size. Every method is evaluated in every configuration. Methods use the fixed evidence pool. Intervals, paired sample sizes, correctness, and call counts are in the supplementary data. These marginal means are not paired contrasts.

Method	**Opus 5**	**Sonnet 5**	**3.8-27B**	**4-26B**
Draft only	90.0 (100.0) 16.5k; 250	80.8 (100.0) 16.5k; 250	89.7 (100.0) 16.5k; 250	81.2 (100.0) 16.6k; 250
Delete flags	94.3 (97.4) 24.9k; 250	91.0 (98.8) 25.8k; 250	93.2 (99.4) 25.3k; 250	93.3 (96.9) 27.7k; 250
Self-check	92.9 (99.2) 36.5k; 250	90.6 (99.6) 37.5k; 250	89.9 (94.2) 45.8k; 250	80.9 (98.2) 37.3k; 250
Self-Refine	87.6 (99.0) 47.6k; 250	86.7 (96.7) 54.4k; 250	87.5 (91.2) 49.8k; 250	80.1 (94.9) 41.3k; 250
CoVe	90.4 (99.4) 69.6k; 250	86.2 (96.4) 70.9k; 250	90.4 (97.9) 74.0k; 250	79.5 (96.0) 79.0k; 250
RARR	90.7 (100.0) 30.2k; 250	88.3 (99.9) 33.3k; 250	89.5 (99.3) 33.6k; 250	81.2 (99.6) 34.1k; 250
RARR + verifier	96.3 (99.6) 28.1k; 250	93.1 (99.6) 31.7k; 250	93.5 (98.7) 28.0k; 250	86.2 (97.8) 34.5k; 250
RARR + keep uncertain	96.1 (99.6) 28.0k; 250	92.1 (99.6) 31.3k; 250	92.2 (99.2) 27.3k; 250	86.4 (98.4) 34.2k; 250
RARR + re-check	96.3 (99.6) 28.9k; 250	93.2 (99.6) 32.8k; 250	93.9 (99.2) 27.9k; 250	97.0 (96.4) 37.7k; 250
VeriCite	96.8 (98.5) 70.2k; 250	92.7 (89.9) 58.1k; 250	94.8 (81.6) 61.3k; 250	97.7 (90.9) 57.9k; 250
Delete/regenerate	90.4 (99.4) 41.5k; 250	91.1 (98.3) 42.2k; 250	95.0 (99.3) 41.8k; 250	90.9 (89.6) 44.1k; 250
ReAct	96.4 (96.9) 95.2k; 250	89.4 (99.8) 61.3k; 250	92.8 (99.2) 88.6k; 250	81.5 (99.7) 113.0k; 250
Best of three	95.3 (99.3) 74.5k; 250	92.0 (100.0) 75.9k; 250	95.2 (99.7) 76.1k; 250	95.2 (98.9) 83.5k; 250
Edit once	95.0 (99.6) 26.5k; 250	91.8 (99.6) 27.9k; 250	91.9 (99.4) 26.8k; 250	86.0 (99.1) 33.2k; 250
Stop after filtering	94.9 (99.6) 27.6k; 250	92.0 (99.6) 29.5k; 250	92.7 (99.2) 27.4k; 250	93.5 (98.2) 36.0k; 250
Loop, two rounds	95.1 (99.6) 27.8k; 250	92.0 (99.6) 29.7k; 250	92.7 (99.2) 27.8k; 250	93.6 (98.5) 41.2k; 250
Self-check + loop	96.0 (99.2) 42.9k; 250	93.2 (99.2) 44.4k; 250	89.6 (94.5) 43.9k; 250	93.9 (96.5) 58.2k; 250

Table 22: Open-domain QA: every baseline, with support and preservation. Each entry gives support (kept supported content), in percent. The second line gives total operational tokens in thousands and the arm sample size. Every method is evaluated in every configuration. Methods use the fixed evidence pool. Intervals, paired sample sizes, correctness, and call counts are in the supplementary data. These marginal means are not paired contrasts.

Method	**Opus 5**	**Sonnet 5**	**3.8-27B**	**4-26B**
Draft only	68.6 (100.0) 2.1k; 850	68.7 (100.0) 2.1k; 850	79.7 (100.0) 2.1k; 850	91.5 (100.0) 1.9k; 850
Delete flags	81.9 (97.2) 4.2k; 850	76.0 (97.5) 3.7k; 850	90.7 (98.6) 3.9k; 850	94.8 (99.2) 3.3k; 850
Self-check	91.9 (98.8) 6.5k; 850	84.1 (99.4) 5.7k; 850	91.7 (95.4) 6.6k; 850	94.0 (97.8) 5.9k; 850
Self-Refine	86.2 (91.4) 11.3k; 850	77.0 (93.4) 8.6k; 850	84.5 (88.3) 8.3k; 850	91.8 (89.0) 9.3k; 850
CoVe	78.5 (99.1) 14.6k; 850	69.4 (97.8) 12.3k; 850	82.5 (97.6) 13.1k; 850	93.0 (97.0) 10.3k; 850
RARR	73.2 (99.9) 4.8k; 850	74.1 (99.9) 4.7k; 850	83.0 (99.6) 4.6k; 850	91.9 (99.8) 3.9k; 850
RARR + verifier	84.0 (99.8) 5.8k; 850	75.8 (99.7) 4.7k; 850	89.9 (99.6) 5.0k; 850	94.1 (99.8) 3.7k; 850
RARR + keep uncertain	84.0 (99.8) 5.8k; 850	75.8 (99.7) 4.7k; 850	89.9 (99.6) 5.0k; 850	94.1 (99.8) 3.7k; 850
RARR + re-check	84.3 (99.8) 6.2k; 850	76.5 (99.5) 4.9k; 850	90.9 (99.4) 5.2k; 850	94.7 (99.7) 3.7k; 850
VeriCite	88.0 (94.9) 12.6k; 850	70.1 (92.9) 9.2k; 850	90.2 (84.8) 9.6k; 850	95.2 (96.9) 8.3k; 850
Delete/regenerate	81.5 (98.6) 6.6k; 850	77.7 (96.4) 5.9k; 850	90.4 (97.7) 5.9k; 850	94.7 (99.1) 5.2k; 850
ReAct	83.6 (97.5) 25.4k; 850	72.4 (99.2) 10.4k; 850	89.1 (98.5) 19.7k; 850	91.8 (99.5) 11.3k; 850
Best of three	80.6 (98.9) 12.7k; 850	74.2 (99.0) 11.2k; 850	89.9 (99.7) 11.8k; 850	94.4 (99.7) 9.8k; 850
Edit once	84.5 (99.7) 5.5k; 850	75.2 (99.4) 4.6k; 850	89.9 (99.5) 5.0k; 850	94.5 (99.8) 3.7k; 850
Stop after filtering	84.6 (99.7) 6.3k; 850	76.2 (99.5) 5.0k; 850	90.7 (99.5) 5.3k; 850	94.8 (99.6) 3.8k; 850
Loop, two rounds	84.7 (99.7) 6.4k; 850	76.3 (99.5) 5.3k; 850	90.7 (99.5) 5.4k; 850	94.8 (99.6) 3.8k; 850
Self-check + loop	91.6 (98.7) 7.2k; 850	84.1 (99.4) 6.9k; 850	94.4 (96.8) 7.1k; 850	96.2 (97.9) 5.8k; 850

A high-support RARR variant, without another rewrite. For Gemma4-26B-A4B-FP8 on Documents, RARR with verifier selection has 86.2% support. Keeping uncertain claims gives 86.4%. Adding re-check and deletion to the latter gives 97.0%. All 250 pairs share draft claims, and the re-check variant removes 106 of the preceding variant's 694 final claims. All 588 survivors have identical text and citations. No further textual repair explains this gain. Empty claim sets rise from one to seventeen, so the nonempty-arm means use different denominators. On the 233 jointly defined pairs, the support gain over the uncertain-kept variant is $+5.31$ points, not the 10.6-point difference of marginal means. Comparing directly to the 86.2% variant also changes the uncertainty policy. This example supports separating repair, removal, and endpoint inclusion rather than treating every support gain as better editing.

More candidates do not guarantee better support. On Documents, best-of-three and the loop have small paired support differences for Opus and Sonnet, with intervals spanning zero. On Open-

domain QA, the loop exceeds best-of-three by 4.12 [3.05, 5.22] points for Opus and 2.09 [0.97, 3.22] for Sonnet. Against RARR with verifier selection and re-check, the loop differs by -0.14 [-0.37, $+0.07$] for Qwen3.8-27B-FP8 and $+0.09$ [-0.12, $+0.31$] for Gemma4-26B-A4B-FP8. These last intervals do not establish equivalence. The Open-domain QA arms have 850 runs. The Gemma support contrast uses 848 defined pairs under the empty-output rule (Appendix B).

Complete outputs can include intermediate abstentions. All four Opus/Sonnet best-of-three arms have complete answer/audit coverage. Their meters record at least one abstained intermediate call in 9/250 and 10/250 Documents runs, and 23/850 and 17/850 Open-domain QA runs, respectively. Complete output coverage therefore does not establish three successful candidates on every item.

D.5 The increment after a fixed self-check pass

The one-pass self-check arm keeps the answer after one whole-answer revision. The composition arm starts with that cached self-check call, then runs the selective verifier, editor, re-check, and final filter. The comparison therefore measures the complete loop's addition to one self-check. It does not isolate which operation inside the added loop supplies the gain, and it does not repeat the selective first-edit experiment from a new starting state.

Prefix and inclusion checks. All 5,400 item pairs share the draft question, answer line, raw draft, claim ids, texts, and citations. Their first self-check action counts, post-pass claim counts, and token usage also agree, including one initially empty draft with no self-check call. The two branches use the same prompt and content-addressed cache. The recorded loop history does not preserve every self-check deletion, so it cannot independently reconstruct every intermediate answer. The analysis replaced an initially planned full-history reconstruction check with input and action-count checks after identifying this logging limitation. We additionally verify complete draft input fields, claim counts, and token usage in the publication audit. Reconstruction agreement is retained as a separate diagnostic in the data. This provenance check is distinct from complete claim-level reconstruction of S_1 in the main ladder.

Each model has 250 Legal, 250 Documents, and 850 Open-domain QA paired records. Endpoint inclusion excludes undefined values separately, giving the support sample sizes in Table 23. All 12 support intervals are positive. The expectation recorded before the added runs, that the Opus and Sonnet Open-domain QA increment would be near zero or negative, is not borne out.

Table 23: **Adding the selective loop after one whole-answer self-check.** Paired changes from keeping that first self-check output. Support and kept supported content are points, cost is thousands of operational tokens per answer. Support uses the displayed endpoint-specific n. Intervals are descriptive 95% source-cluster intervals from 2,000 draws. The loop includes verification, editing, and filtering. Its individual operations are not separated here.

Model	n	Δ support	Δ kept	Δ tokens (k)
Legal				
Opus 5	250	+2.62 [+1.75, +3.58]	-0.48 [-1.14, +0.00]	+2.3 [+2.1, +2.4]
Sonnet 5	250	+3.63 [+2.10, +5.28]	-0.62 [-1.70, +0.14]	+1.7 [+1.6, +1.9]
3.8-27B	247	+3.75 [+2.33, +5.33]	-0.33 [-1.22, +0.24]	+2.0 [+1.8, +2.3]
4-26B	249	+2.62 [+1.48, +3.80]	-0.32 [-0.73, +0.00]	+1.5 [+1.4, +1.7]
Documents				
Opus 5	250	+3.60 [+1.93, +5.56]	+0.00 [+0.00, +0.00]	+9.8 [+8.7, +10.9]
Sonnet 5	249	+3.78 [+2.10, +5.59]	-0.36 [-0.94, +0.00]	+11.3 [+9.9, +12.9]
3.8-27B	240	+2.64 [+1.09, +4.50]	+0.62 [-0.39, +1.94]	+10.8 [+9.2, +12.5]
4-26B	241	+10.09 [+7.10, +13.16]	-1.57 [-3.12, -0.30]	+25.0 [+20.9, +29.3]
Open-domain QA				
Opus 5	850	+2.74 [+2.25, +3.23]	-0.27 [-0.62, -0.01]	+2.8 [+2.7, +2.9]
Sonnet 5	850	+2.57 [+2.00, +3.17]	-0.34 [-0.60, -0.13]	+2.6 [+2.5, +2.7]
3.8-27B	848	+5.33 [+4.58, +6.07]	-0.28 [-0.59, +0.01]	+2.7 [+2.6, +2.8]
4-26B	848	+2.57 [+1.99, +3.22]	-0.42 [-0.88, -0.06]	+1.8 [+1.7, +1.8]

The largest support increment, for Gemma4-26B-A4B-FP8 on Documents, accompanies a 1.6-point decrease in retained initially supported content and a 6.9-point decrease in retained content over all draft claims. These endpoints have different denominators. Cost also varies substantially by task. Correctness and complete retention endpoints are reported alongside support in the supplementary data, without assuming that higher support improves every endpoint.

Table 24: **Two ways to continue after self-check.** Self-check plus the selective loop minus the two-pass self-check baseline in Table 2. Units and intervals match Table 23. This is a comparison of complete continuations with different budgets.

Model	n	Δ support	Δ kept	Δ tokens (k)
Legal				
Opus 5	250	+0.99 [-0.18, +2.17]	-0.54 [-1.41, +0.29]	+0.4 [+0.2, +0.6]
Sonnet 5	250	+2.07 [+0.85, +3.32]	-0.21 [-1.46, +1.06]	+0.9 [+0.7, +1.1]
3.8-27B	247	+1.82 [-0.14, +3.88]	+1.76 [+0.23, +3.30]	-0.2 [-0.4, +0.0]
4-26B	249	+1.44 [+0.02, +2.93]	-0.17 [-0.78, +0.47]	-0.0 [-0.2, +0.2]
Documents				
Opus 5	250	+3.13 [+1.53, +5.01]	+0.00 [+0.00, +0.00]	+6.4 [+5.2, +7.6]
Sonnet 5	249	+2.23 [+0.70, +3.80]	-0.36 [-0.94, +0.00]	+6.9 [+5.3, +8.5]
3.8-27B	238	+0.46 [-2.01, +2.97]	+0.28 [-1.62, +2.26]	-1.9 [-3.7, -0.0]
4-26B	241	+10.17 [+7.16, +13.31]	-1.66 [-3.21, -0.39]	+20.9 [+17.1, +24.8]
Open-domain QA				
Opus 5	850	-0.30 [-0.91, +0.31]	-0.09 [-0.50, +0.29]	+0.7 [+0.6, +0.8]
Sonnet 5	850	+0.07 [-0.64, +0.77]	+0.05 [-0.26, +0.38]	+1.2 [+1.0, +1.3]
3.8-27B	847	+2.62 [+1.91, +3.35]	+1.40 [+0.71, +2.10]	+0.5 [+0.4, +0.6]
4-26B	848	+2.06 [+1.31, +2.84]	+0.13 [-0.45, +0.65]	-0.1 [-0.2, +0.0]

Table 24 compares continuing with the selective loop against the two-pass whole-answer self-check baseline. On Open-domain QA, the Opus and Sonnet support differences are small, with intervals spanning zero. They do not establish equivalence. The loop and the extra self-check have different costs and operations. These results support testing complete alternatives on the target task, without establishing a universally best correction strategy or an optimal fixed-budget allocation.

D.6 CROSS-FAMILY VERIFICATION

Table 25: Cross-family verification on 250 items per task and model pair. All values are percentages. Flag, Delete, and Loop report held-out support. All-draft kept and Correct report the loop's retention and task correctness. Retention uses all draft claims as denominator, including initially unsupported claims. V+/H– is the share of verifier-accepted draft claims rejected by the held-out judge. This measures model disagreement.

Generator / verifier	Flag	Delete	Loop	All-draft kept	Correct	V+/H–
⚖ ***Legal***						
Opus / Opus	73.3	85.4	87.7	95.3	88.4	14.5
Opus / Qwen	73.3	88.1	89.7	94.4	90.8	12.3
Qwen / Qwen	83.6	90.8	91.3	97.2	83.6	8.7
Qwen / Opus	83.6	91.2	91.7	97.3	83.2	8.5
🗎 ***Documents***						
Opus / Opus	90.0	94.3	95.1	97.6	84.0	2.6
Opus / Qwen	90.0	92.0	92.5	98.8	84.4	3.4
Qwen / Qwen	89.7	93.2	92.7	97.3	77.2	4.2
Qwen / Opus	89.7	94.6	94.2	96.4	77.2	3.1

Cross-family verification changes outcomes as well as error rates. On legal, the loop support is 87.7 versus 89.7 for Opus drafts under Opus versus Qwen3.8-27B-FP8 verification. It is 91.3 versus 91.7 for Qwen3.8-27B-FP8 drafts. On documents, changing verification also changes the endpoint. These comparisons probe verifier dependence. They do not isolate the same operation as the shared-prefix ladder. Comparisons to a frontier model's plain draft change both model and method and are contextual, not evidence that the loop is universally superior.

E MECHANISM AND ROBUSTNESS

E.1 PRE-ACTION ACCOUNTING AND DENOMINATOR CHECK

The counting constraint. Let $C_0 = S_0 \cup U_0$ be draft claims supported and unsupported by the held-out judge. A selector changes claims in $A \subseteq C_0$, reaching a fraction $r = |A \cap U_0|/|U_0|$ of the errors when $|U_0| > 0$. With fixed claim identity and unchanged claims outside A, the final unsupported claims follow three paths: missed errors, failed repairs, and newly introduced errors. In particular,

$$\frac{|U_{\text{final}}|}{|C_0|} \;\geq\; (1-r)\frac{|U_0|}{|C_0|}. \tag{2}$$

The right side counts initially unsupported claims never changed. Deletion is recorded separately from repair. The floor follows directly from counting these unchanged errors and constrains unsupported mass. It gives no upper bound on RETRY-minus-STOP support, which averages per-answer fractions whose denominators and inclusion can change after filtering.

The measured state. The pre-action state is the saved answer after the first edit and any re-check, before filtering or a second action. Draft labels come from the flag audit, and subsequent labels from the corresponding held-out audits. Selected claims include initial queue entries and verifier-triggered document re-citations. The second queue is read from the run record and checked against the retry arm.

We distinguish unsupported claims still present from supported draft claims already lost. We record lost supported content separately and exclude it from both the present-error count and the count outside the next edit list. We compute item-level indicators and source-cluster intervals using this definition. An independent match of claim identities into the action outputs checks the aggregate counts, which satisfy

$$U_0 = N_{\text{never}} + N_{\text{fixed}} + N_{\text{dropped}} + N_{\text{failed}}, \qquad U_1 = N_{\text{never}} + N_{\text{failed}} + N_{\text{new}}.$$

All 1,800 item records pass these checks. The corrected totals for the same-model legal configurations are 147 present errors, including 132 never selected. Documents have 93, including 52 never selected. Of these, 30 and 11 come from the Gemma4-26B-A4B-FP8 diagnostic cell. Its outside-list share is 47% [28, 68] because 15 re-flagged failed repairs remain. The available Qwen3.8-27B-FP8 open-domain trace has 38, including 32 never selected. These descriptive totals combine configurations that share sources and sometimes drafts.

Table 26: Complete claim accounting before the second action, 100 items per configuration. $U_0 =$ never selected + fixed + dropped + failed. New errors are initially supported claims that remain but are now unsupported. Lost supported claims are separate. These two categories must not be combined when counting residual errors. Document selection includes citation changes.

Configuration	U_0	**Never selected**	**Fixed**	**Drop**	**Failed**	**New errors**	**Lost** S_0
⚖ Legal / Qwen	61	33	16	9	3	1	2
⚖ Legal / Opus	115	44	60	6	5	0	0
⚖ Legal / Sonnet	72	40	12	17	3	3	1
⚖ Legal / Gemma	24	15	9	0	0	0	0
⚖ Legal / Qwen → 9B	61	33	13	10	5	1	2
⚖ Legal / Qwen → Sonnet	61	33	16	8	4	1	2
⚖ Legal / Qwen → Opus	61	33	21	6	1	1	3
📄 Documents / Qwen	26	15	7	1	3	1	0
📄 Documents / Opus	25	8	10	0	7	0	0
📄 Documents / Sonnet	65	18	34	2	11	0	0
📄 Documents / Gemma	45	11	14	2	18	1	0
📄 Documents / Qwen → 9B	26	15	3	4	4	3	0

Continued on next page.

Table 26 continued.

Configuration	U_0	Never selected	Fixed	Drop	Failed	New errors	Lost S_0
Documents / Qwen → Sonnet	26	15	8	1	2	1	0
Documents / Qwen → Opus	26	15	11	0	0	1	0
Open-domain QA / Qwen	92	32	43	11	6	0	5
Open-domain QA / Qwen → 9B	92	32	39	6	15	2	2
Open-domain QA / Qwen → Sonnet	92	32	36	6	18	0	3
Open-domain QA / Qwen → Opus	92	32	53	3	4	0	0

Table 27: Opportunity at the pre-action state. First recall is the fraction of initial J1 errors selected, including document citation changes. Present errors exclude deleted claims. All queued claims can include false re-flags. Errors in queue counts only J1-rejected claims still present. Ratios and 95% intervals are recomputed from per-item exports with 10,000 source-cluster resamples. Configurations share source items. The queue is not the complete filtering target: Documents/Qwen-to-9B has two unaddressed initial flags removed at termination, beyond its four queued errors (Appendix E.1).

Configuration	First recall (%)	Present errors	Errors in queue	All queued	Outside queue (%)	Error items with empty queue (%)
Legal / Qwen	45.9	37	1	4	97 [90, 100]	97 [88, 100]
Legal / Opus	61.7	49	1	3	98 [93, 100]	97 [91, 100]
Legal / Sonnet	44.4	46	5	6	89 [78, 98]	87 [73, 98]
Legal / Gemma	37.5	15	0	1	100 [100, 100]	100 [100, 100]
Legal / Qwen → 9B	45.9	39	5	10	87 [73, 98]	81 [67, 95]
Legal / Qwen → Sonnet	45.9	38	5	9	87 [76, 97]	84 [72, 96]
Legal / Qwen → Opus	45.9	35	1	4	97 [90, 100]	97 [88, 100]
Documents / Qwen	42.3	19	2	3	89 [74, 100]	88 [67, 100]
Documents / Opus	68.0	15	1	1	93 [76, 100]	93 [75, 100]
Documents / Sonnet	72.3	29	1	2	97 [88, 100]	95 [85, 100]
Documents / Gemma	75.6	30	16	22	47 [28, 68]	48 [27, 68]
Documents / Qwen → 9B	42.3	22	4	5	82 [67, 96]	72 [50, 92]
Documents / Qwen → Sonnet	42.3	18	2	2	89 [73, 100]	87 [65, 100]
Documents / Qwen → Opus	42.3	16	0	1	100 [100, 100]	93 [75, 100]
Open-domain QA / Qwen	65.2	38	4	6	89 [79, 98]	87 [74, 97]
Open-domain QA / Qwen → 9B	65.2	49	13	18	73 [61, 84]	66 [50, 80]
Open-domain QA / Qwen → Sonnet	65.2	50	15	16	70 [56, 84]	68 [53, 83]
Open-domain QA / Qwen → Opus	65.2	36	1	2	97 [91, 100]	97 [88, 100]

The supplementary claim-action records trace each of these groups through stopping, retry, and re-inspection, retaining supported, unsupported, and removed outcomes separately. The second-round queue can contain false re-flags of supported claims. For legal Qwen3.8-27B-FP8, it contains four claims, only one of which is a present error under J1. Counting queued claims as repair opportunities would overstate the available opportunity.

The next edit list is not the complete filtering target. If the editor leaves initial flags unaddressed and no rewrite remains to check, correction terminates without a second queue. The final filter still removes claims labeled unsupported or wrong-page by the in-loop checker. For Documents with Qwen3.8-27B-FP8 drafts and the Qwen3.5-9B editor, 22 errors are present at S_1 and four are in Q_2. Two more initially flagged claims in one answer remain unaddressed, then are removed at termination. Stop therefore removes six errors and leaves sixteen, as Figure 4c reports. The raw action records preserve these two removals. "Failed, outside next queue" does not necessarily mean accepted by the checker. In general,

$$|U_{\text{final}}| = |U_1| - N_{\text{repaired}} - N_{\text{errors removed}} + N_{\text{new}}.$$

Subtracting only errors in Q_2 would give eighteen in this configuration and miss the two terminal removals. Filtering does not supply another generative attempt on those claims.

Why seven unsuccessful changes can coexist with zero eligible final errors. In the 100-item document Opus trace, seven selected initial errors remain unsupported at the pre-action state. Six received only a page-citation correction and never entered the original edit queue. All six remain present under ordinary retry with unchanged text and citation. Their held-out verdict remains VERIFIED_NEARBY. The seventh is a queued rewrite, is re-flagged, and becomes supported under ordinary retry. The final-answer controls below restrict eligibility to the original edit queue, so they have zero eligible Opus document errors. The six citation-only failures are not failed rewrites or disappearing judge labels.

What a repaired missed claim means. Re-inspection makes 16/33 initially missed Legal Qwen3.8-27B-FP8 claims supported, leaves thirteen unsupported, and removes four. For Opus, the corresponding counts are 26/44, seventeen, and one. Fifteen Qwen3.8-27B-FP8 successes change text and one changes only the citation. Opus uses twenty rewording and six softening actions. These transitions restore full support under J1's strict rule. Same-scope correction of false facts requires a separate assessment. Table 9 separates human-recognized before-to-after improvements from simply having a supported after version. Appendix C.8 shows why relaxing partial support changes the missed-error population.

E.2 Residual paths and selector sweeps

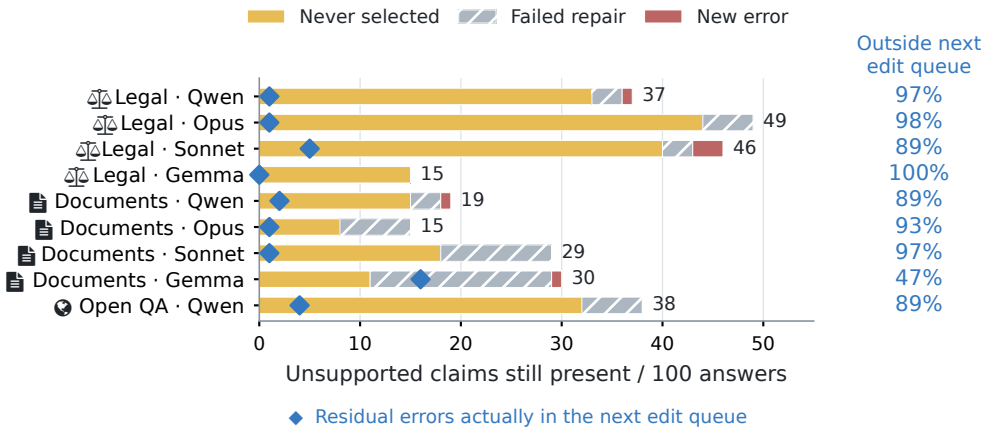

Figure 7: **Most remaining errors never entered the edit list.** All nine traces using the same model for generation, verification, and editing. Each bar counts unsupported claims present after one edit and re-check, before filtering, across 100 answers. Colors distinguish missed errors, failed repairs, and new errors. Blue diamonds count those selected for another edit. Percentages give the share outside that list. Counts use individual claims as their unit.

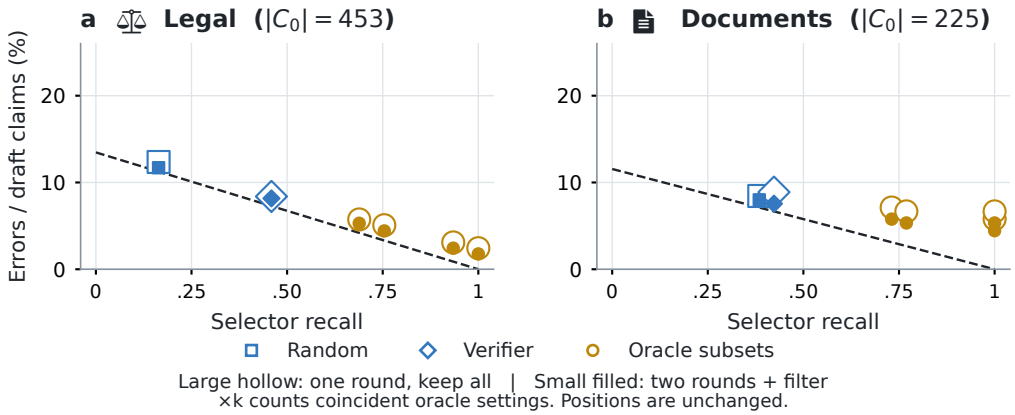

Figure 8: **Some errors survive even when their locations are known.** Qwen3.8-27B-FP8 drafts and editor, 100 answers per task. C_0 is the draft-claim denominator. Both model figures use the same axes. The dashed floor counts errors never reached. Large hollow markers use one round with re-check and keep all surviving claims. Small filled markers allow two rounds and final filtering. Overlapping oracle outcomes share a marker, with multiplicity shown. Random and oracle variants omit reasons. The verifier pair also changes to reason-bearing feedback, so its difference is not an isolated round effect. Oracle subsets provide error locations without corrected text.

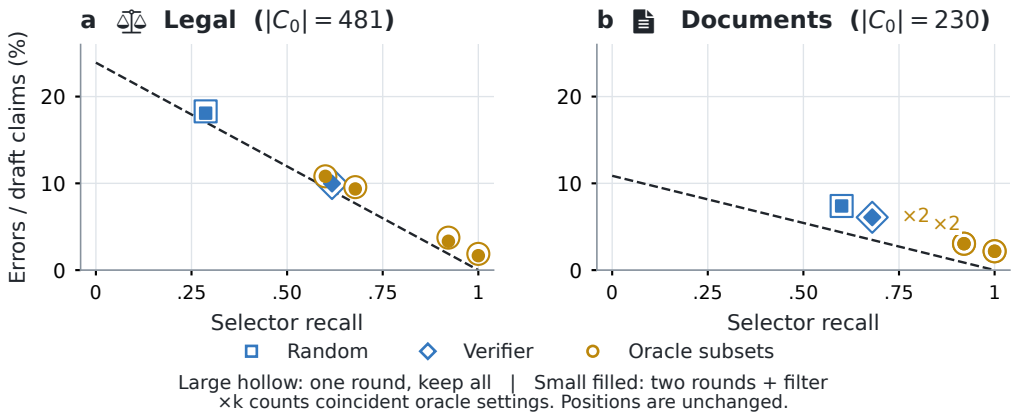

Figure 9: **Selection and repair with Opus.** Opus drafts and editor, 100 answers per task, with the same axes as Figure 8. C_0 gives the draft-claim denominator. Large hollow markers keep surviving claims after one round and re-check. Small filled markers allow two rounds and filtering. Concentric markers indicate equal outcomes, with overlapping oracle multiplicity shown. Random and oracle variants omit reasons. The verifier pair additionally changes feedback. Oracle locations yield 101/115 repairs on Legal and 20/25 on Documents, higher than for Qwen3.8-27B-FP8 on its different drafts.

Complete residual-path counts for the selector, feedback, and budget interventions are provided in the supplementary data. The curves above show the measured-recall comparisons without repeating a long path matrix for each intervention.

E.3 Editor gains across exploration and confirmation

Table 28 places the two positive exploratory Documents editor results beside their source-disjoint confirmation outcomes. Sonnet changes from a positive exploratory estimate to a slightly negative confirmation estimate, while Opus remains positive. We do not attribute this contrast to a particular cause or infer an editor ranking across tasks. The results support checking each candidate on held-back examples, rather than choosing an editor from the exploratory gain alone.

Table 28: **An exploratory gain need not carry over.** Re-inspect minus stop on Documents with fixed Qwen3.8-27B-FP8 drafts and verifier, changing the editor. Entries are support percentage points [95% source-cluster interval]. Exploratory intervals retain the original export. Confirmation point estimates use unchanged source-disjoint action outcomes, with descriptive intervals computed retrospectively from 10,000 document-cluster draws. A sign change is not a direct test of a difference between cohorts. These comparisons do not establish a task-wide editor ranking.

Editor	Expl. n	Exploration	Conf. n	Confirmation
Sonnet 5	100	+1.67 [+0.33, +3.50]	143	-0.33 [-3.82, +2.19]
Opus 5	100	+1.67 [+0.00, +4.00]	143	+1.89 [+0.62, +3.50]

E.4 Second-round controls

Table 29: Second-round controls: counts. Forced retry is d-force. Independent re-check is d-ind. Slash-separated edit counts show ordinary/control. Eligible errors are defined using the ordinary loop's final answer. Harm includes supported claims lost or made unsupported.

Configuration	Control	n	Items edited	Claims edited	Errors	Fixed	Drop	Harm
legal / Qwen3.8	d-force	100	4 / 22	4 / 28	4	2	1	0
legal / Qwen3.8	d-ind	100	4 / 5	4 / 5	4	2	1	0
document / Qwen3.8	d-force	100	3 / 10	3 / 13	2	0	2	1
document / Qwen3.8	d-ind	100	3 / 2	3 / 2	2	0	0	0
legal / Opus 5	d-force	100	3 / 59	3 / 93	4	4	0	2
legal / Opus 5	d-ind	100	3 / 5	3 / 5	4	1	0	0
document / Opus 5	d-force	100	1 / 14	1 / 15	0	0	0	0
document / Opus 5	d-ind	100	1 / 1	1 / 1	0	0	0	0

Table 30: Effects of second-round controls on the same 100 items per configuration. Forced retry (d-force) re-edits all initial flags. Independent re-check (d-ind) changes the verifier. Differences are control minus ordinary retry. Support changes are percentage points with 95% source-cluster intervals. Token changes are mean input plus output tokens per answer.

Configuration	Control	Δ support [CI]	Δ tokens
legal / Qwen3.8	d-force	+0.6 [+0.0, +1.3]	+506
legal / Qwen3.8	d-ind	+0.6 [+0.0, +1.4]	+20
document / Qwen3.8	d-force	-0.3 [-2.2, +1.4]	+1723
document / Qwen3.8	d-ind	-0.2 [-0.6, +0.0]	-83
legal / Opus 5	d-force	-0.0 [-1.0, +0.9]	+1553
legal / Opus 5	d-ind	+0.1 [-0.6, +0.8]	+41
document / Opus 5	d-force	+0.0 [+0.0, +0.0]	+3177
document / Opus 5	d-ind	+0.0 [+0.0, +0.0]	+27

The eligibility set for these controls is round-0 queued claims still present and unsupported in the ordinary loop's *final* answer: 4/2 errors for Qwen3.8-27B-FP8 on legal/documents and 4/0 for Opus. It is not a prospective pre-action opportunity set. Zero eligible errors is not a zero repair rate. The repaired Opus legal audit re-ran 15 abstained correctness calls that had previously appeared as failures. The displayed control outcomes use the repaired audit.

E.5 Feedback, round limits, and whole-answer variants

For Qwen3.8-27B-FP8 on the exploration block, support with location alone, a reason, an evidence quote, and a shuffled reason is 91.7/91.7/91.7/92.0 on legal and 92.4/92.5/92.5/92.6 on documents. These contrasts do not establish that reason content improves correction. On the larger document block, hiding reasons costs about 1.6 points for Sonnet and Gemma4-26B-A4B-FP8, showing that feedback sensitivity depends on editor and modality. The small pilot's suggested reason-channel gaming is not retained as a finding.

Round limits 1, 2, and 3 change little in the Opus and Qwen3.8-27B-FP8 legal/document configurations. This is not a universal claim about all models or all queues. Full redrafting can obtain similar support while dropping more content. Empty outputs are especially important there: 38 of 250 Sonnet legal outputs and 24 of 250 Gemma4-26B-A4B-FP8 document outputs end with no claims. The targeted loop also has nine empty document outputs from the latter model. Thus, some configurations have more than one empty answer.

Self-check alone reaches farther into legal drafts than the selective loop but can alter supported claims. It is weaker on page-image questions. Giving the same Gemma4-26B-A4B-FP8 document items as

OCR text changes both grounding and correctness: the plain draft is better supported but less often correct. These controls caution against attributing every difference to model size or to selector recall alone.

Self-check is simple to implement but can cost more. The baseline permits two whole-answer revision passes and stops early when no claims change. Its mean token-to-draft ratio is 2.64, versus 2.05 for retry, over the same eight ⚖ Legal/📄 Documents configurations. Self-check followed by the loop starts after one pass and is a separate complete strategy (2.94). The shared-prefix ladder starts from the selective procedure's first edit. Adoption should follow target-task comparisons of support, retention, and cost.

The supplementary tables retain all feedback, round-limit, OCR, and self-check conditions, including negative and null outcomes. Here we summarize their implications rather than repeat endpoints already shown in the main results.

Following the same claims through each action. Figure 4c counts all unsupported claims after each alternative action, not the errors selected beforehand. Exact claim ids join each of the eighteen 100-answer S_1 cohorts to stop, retry, and re-inspection. Initial errors split into still unsupported, now fully supported, and removed. Final errors equal still unsupported plus newly unsupported claims. Supported claims removed are counted separately, not as errors or repairs. On ⚖ Legal with Qwen3.8-27B-FP8, stop/retry/re-inspect leave 36/37/22 errors. Re-inspection makes 16 of the 37 initial errors supported, removes five, and leaves sixteen. It also introduces six errors and removes nineteen supported claims. On 📄 Documents with Gemma4-26B-A4B-FP8, it makes four of thirty errors supported, removes sixteen, and leaves ten, with no new errors but thirteen supported claims removed. These are J1 claim-state counts, not human-certified repairs or semantic retention scores. Complete outcomes for all 54 action/configuration pairs, 5,400 item/action records, and their claim-level joins accompany the paper.

E.6 EDITING PERMISSIONS AND MECHANICAL WORDING CONTROLS

All controls start from the same pre-action state as re-inspection. Eligible claims are accepted and never queued, and the same checker and final filtering policy apply afterward. Qualify-only permits softening, adding a condition, or narrowing while keeping the source and claim count fixed (Prompt 25). Reword-or-drop permits changing content or deleting a claim, with instructions against qualification (Prompt 26). These are prompt and parser restrictions, not human-certified categories of the resulting edits. Their comparison also changes deletion permissions.

The mechanical controls prefix every eligible claim with either "It appears that" or "In some cases,". The prefix operation does not inspect evidence or select a correction, and it leaves the citation fixed. The ordinary verifier then checks the changed claims against evidence and can remove failures. These controls test the complete consequence of a fixed wording change followed by checking and filtering. Their support gain cannot be attributed solely to the scorer's response to unchanged substantive content, since scope and final inclusion can change.

Coverage. Seven configurations contain all four additional controls and unrestricted re-inspection. The Qwen3.8-27B-FP8 self-editor cohorts contain 320 ⚖ Legal, 240 📄 Documents, and 100 🌐 Open-domain QA answers. The Opus and Sonnet editor cohorts contain 323 ⚖ Legal and 194 📄 Documents answers, with Qwen3.8-27B-FP8 drafts and checker. These latter cohorts pool the earlier calibration and confirmation items for a new diagnostic follow-up. They are not a second held-out confirmation of the original policy. Four further configurations have only the unrestricted reference arm and remain in the supplementary export. Thus, seven configurations supply the complete editing-permission comparison.

Table 31: **Editing permissions on ⚖ Legal.** Changes from stopping at the same pre-action state. Point units and 95% source-cluster intervals. All drafts and checkers use Qwen3.8-27B-FP8. Both kept-content columns use initially supported draft claims. The last also requires the same scope. Each continuation ends with the usual re-check and filtering. Endpoint-specific n is for support. All endpoints, cost, and four additional unrestricted reference configurations are in the supplementary data.

Continuation	n	Δ support	Δ kept	Δ kept (same scope)
3.8-27B editor, 320 shared items				
Re-inspect	318	+2.98 [+1.73, +4.31]	-1.54 [-2.44, -0.78]	-11.39 [-13.55, -9.36]
Qualify only	319	+2.35 [+1.26, +3.46]	-0.66 [-1.17, -0.23]	-12.50 [-14.90, -10.31]
Reword or drop	318	+2.59 [+1.33, +3.80]	-1.87 [-2.86, -1.05]	-10.78 [-12.97, -8.78]
Add 'It appears that'	318	+2.47 [+1.41, +3.61]	-0.36 [-0.72, -0.08]	-13.67 [-17.47, -10.19]
Add 'In some cases'	314	+3.63 [+2.35, +4.91]	-10.20 [-12.34, -8.14]	-79.65 [-82.50, -76.82]
Sonnet 5 editor, 323 shared items				
Re-inspect	321	+2.02 [+1.18, +2.89]	-1.17 [-2.07, -0.42]	-2.50 [-3.68, -1.44]
Qualify only	321	+2.78 [+1.71, +3.84]	-0.73 [-1.31, -0.24]	-6.37 [-8.08, -4.85]
Reword or drop	321	+1.70 [+0.93, +2.55]	-0.62 [-1.12, -0.18]	-1.76 [-2.61, -0.99]
Add 'It appears that'	320	+2.14 [+1.11, +3.22]	-0.45 [-0.82, -0.13]	-16.61 [-20.51, -13.08]
Add 'In some cases'	315	+2.90 [+1.38, +4.51]	-9.01 [-11.19, -6.91]	-68.72 [-72.07, -65.32]
Opus 5 editor, 323 shared items				
Re-inspect	323	+3.91 [+2.76, +5.14]	-0.66 [-1.18, -0.22]	-5.76 [-7.32, -4.33]
Qualify only	323	+4.92 [+3.69, +6.22]	-1.51 [-2.38, -0.75]	-12.98 [-15.27, -10.96]
Reword or drop	323	+2.85 [+1.87, +3.98]	-0.28 [-0.76, +0.16]	-2.81 [-3.84, -1.88]
Add 'It appears that'	323	+2.05 [+1.02, +3.07]	-0.50 [-0.93, -0.16]	-15.97 [-19.89, -12.49]
Add 'In some cases'	318	+2.85 [+1.41, +4.41]	-8.06 [-9.96, -6.11]	-68.39 [-72.06, -64.98]

Table 32: **Editing permissions on Documents.** Changes from stopping at the same pre-action state. Point units and 95% source-cluster intervals. All drafts and checkers use Qwen3.8-27B-FP8. Both kept-content columns use initially supported draft claims. The last also requires the same scope. Each continuation ends with the usual re-check and filtering. Endpoint-specific n is for support. All endpoints, cost, and four additional unrestricted reference configurations are in the supplementary data.

Continuation	n	Δ support	Δ kept	Δ kept (same scope)
3.8-27B editor, 240 shared items				
Re-inspect	228	-1.86 [-4.06, +0.08]	-5.37 [-7.57, -3.36]	-6.94 [-9.41, -4.60]
Qualify only	239	-2.70 [-4.96, -0.67]	-2.71 [-4.54, -1.18]	-5.09 [-7.37, -3.00]
Reword or drop	229	-0.40 [-2.18, +1.16]	-3.56 [-5.36, -1.92]	-6.73 [-9.51, -4.27]
Add 'It appears that'	239	+0.45 [-0.53, +1.65]	-0.30 [-0.84, +0.00]	-50.30 [-57.39, -43.38]
Add 'In some cases'	237	-0.63 [-2.38, +0.79]	-5.37 [-8.33, -2.88]	-85.09 [-89.33, -80.73]
Sonnet 5 editor, 194 shared items				
Re-inspect	194	-0.24 [-2.79, +1.51]	-1.60 [-3.27, -0.28]	-1.77 [-3.44, -0.46]
Qualify only	194	+1.12 [-0.20, +2.64]	-1.14 [-2.48, -0.17]	-2.02 [-3.92, -0.56]
Reword or drop	194	+0.62 [-1.07, +2.14]	-1.60 [-3.27, -0.28]	-2.12 [-3.85, -0.63]
Add 'It appears that'	194	+0.46 [-0.48, +1.42]	+0.00 [+0.00, +0.00]	-32.22 [-39.63, -24.73]
Add 'In some cases'	194	-0.45 [-2.18, +0.74]	-2.04 [-3.84, -0.55]	-87.44 [-91.56, -83.11]
Opus 5 editor, 194 shared items				
Re-inspect	194	+1.15 [-0.15, +2.48]	-1.70 [-3.37, -0.36]	-2.82 [-4.82, -1.25]
Qualify only	194	+1.32 [-0.13, +2.93]	-2.19 [-4.14, -0.64]	-4.25 [-6.45, -2.27]
Reword or drop	194	+1.92 [+0.73, +3.32]	-1.60 [-3.26, -0.34]	-1.77 [-3.48, -0.50]
Add 'It appears that'	194	+0.51 [-0.47, +1.48]	+0.00 [+0.00, +0.00]	-31.69 [-38.88, -24.66]
Add 'In some cases'	194	-0.40 [-2.15, +0.81]	-2.04 [-3.84, -0.55]	-86.91 [-91.38, -82.32]

Table 33: **Editing permissions on Open-domain QA.** Changes from stopping at the same pre-action state. Point units and 95% source-cluster intervals. All drafts and checkers use Qwen3.8-27B-FP8. Both kept-content columns use initially supported draft claims. The last also requires the same scope. Each continuation ends with the usual re-check and filtering. Endpoint-specific n is for support. All endpoints, cost, and four additional unrestricted reference configurations are in the supplementary data.

Continuation	n	Δ support	Δ kept	Δ kept (same scope)
3.8-27B editor, 100 shared items				
Re-inspect	100	+1.40 [-0.55, +3.40]	-6.20 [-9.27, -3.57]	-10.87 [-15.33, -7.00]
Qualify only	100	+0.80 [-0.75, +2.50]	-3.52 [-5.70, -1.70]	-9.58 [-13.38, -6.20]
Reword or drop	100	+2.25 [+0.70, +4.05]	-4.08 [-6.63, -1.88]	-8.27 [-12.08, -5.25]
Add 'It appears that'	100	+2.32 [+0.72, +3.97]	-1.23 [-2.48, -0.25]	-35.53 [-43.33, -28.05]
Add 'In some cases'	100	-1.97 [-4.33, +0.22]	-5.07 [-8.23, -2.65]	-73.75 [-79.17, -67.92]

What the controls establish. The "It appears that" control yields positive paired support intervals on all three Legal configurations and on the Open-domain QA configuration. On Documents, its intervals include zero. The alternative prefix yields positive legal effects but a negative Open-domain QA point estimate. Qualify-only exceeds reword-or-drop in legal support for Opus and Sonnet but not for the Qwen3.8-27B-FP8 editor. Conversely, reword-or-drop has higher same-scope kept content than qualify-only in all three legal configurations. The results expose a trade-off between changing the assertion and preserving it, rather than a generally superior editing permission.

The prespecified prediction that a mechanical hedge recovers at least half of unrestricted re-inspection's support gain is met by the point estimates for the two commercial legal editors, but not for the Opus document editor. The latter's unrestricted gain itself has an interval spanning zero. Ratios of these small gains are descriptive and are not evidence of statistical equivalence. The

prediction that qualification matches or exceeds content rewriting in support also fails for the legal Qwen3.8-27B-FP8 editor. The appendix tables preserve those contrary cases. Claim-level credited counts in the data use draft-to-final transitions, including improvements before the follow-up. They must not be read as additional repairs made by the wording intervention alone.

F Held-out Action Comparisons and Policy Analysis

F.1 Support and information loss on held-out sources

The eight confirmation configurations use source clusters absent from development. The ⚖ Legal pool has 50 calibration and 273 test items, with 158 test contract clusters. The 🗎 Documents pool has 51 calibration and 143 test items, with 38 test document clusters. Four configurations share each task's items. Three editors use fixed Qwen3.8-27B-FP8 drafts and checker, while Gemma4-12B generates and edits its own drafts. Table 34 reports re-inspection effects in all eight configurations, including those where a policy chooses to stop. Re-inspection improves support in several configurations while exceeding the 3-point content-loss guard in three. Complete action scores, choices, and costs accompany the supplementary data.

Table 34: Paired re-inspection minus stop effects, with 95% source-cluster intervals and endpoint-specific n. The first metric is a gain, while the other three are losses, in points. † marks a recorded guard violation, not an interval test. Quality duplicates correctness on ⚖ Legal and uses gold-span coverage on 🗎 Documents, where correctness is lenient. The four editors share each task's items, with Qwen drafts except for Gemma. These descriptive intervals use 10,000 synchronized draws. Complete marginal action scores and calibrated policy choices accompany the supplementary data. Different endpoint inclusion can make a paired gain differ from the marginal-score difference.

Editor	Support gain	Content loss	Correctness loss	Quality loss
⚖ Legal				
3.5-9B†	+0.3 $[-1.1, 1.7]$ $n = 271$	+10.74 $[8.2, 13.5]$ $n = 273$	-1.1 $[-4.0, 1.8]$ $n = 273$	-1.1 $[-4.0, 1.8]$ $n = 273$
Sonnet 5	+1.9 $[1.0, 2.8]$ $n = 273$	+1.19 $[0.4, 2.3]$ $n = 273$	-2.2 $[-5.0, 0.4]$ $n = 273$	-2.2 $[-5.0, 0.4]$ $n = 273$
Opus 5	+4.0 $[2.8, 5.3]$ $n = 273$	+0.65 $[0.2, 1.2]$ $n = 273$	-1.5 $[-4.4, 1.3]$ $n = 273$	-1.5 $[-4.4, 1.3]$ $n = 273$
4-12B	+1.0 $[-0.6, 2.5]$ $n = 271$	+2.42 $[0.8, 4.4]$ $n = 272$	-0.4 $[-1.6, 0.7]$ $n = 273$	-0.4 $[-1.6, 0.7]$ $n = 273$
🗎 Documents				
3.5-9B†	+0.1 $[-2.0, 1.9]$ $n = 142$	+3.78 $[0.9, 7.4]$ $n = 139$	0.0 $[0.0, 0.0]$ $n = 143$	+0.7 $[0.0, 2.2]$ $n = 143$
Sonnet 5	-0.3 $[-3.8, 2.2]$ $n = 143$	+1.80 $[0.0, 4.0]$ $n = 139$	-0.7 $[-2.2, 0.0]$ $n = 143$	0.0 $[0.0, 0.0]$ $n = 143$
Opus 5	+1.9 $[0.6, 3.5]$ $n = 143$	+1.37 $[0.0, 3.2]$ $n = 139$	-1.4 $[-3.6, 0.0]$ $n = 143$	-0.7 $[-2.2, 0.0]$ $n = 143$
4-12B†	-1.3 $[-5.0, 2.0]$ $n = 104$	+10.44 $[4.7, 17.2]$ $n = 75$	+2.8 $[0.7, 5.5]$ $n = 143$	+0.7 $[-1.5, 3.0]$ $n = 143$

Different valid-answer sets can change the comparison. On Documents with Gemma4-12B, stop has 107 defined support scores and retry has 110 among 143 answers. Their marginal-mean difference is -0.075 points, while the 107 complete pairs give $+0.701$. Policy tests use marginal means and descriptive action intervals use complete pairs. Empty outputs have no defined support ratio.

F.2 A SECONDARY COMPARISON OF CALIBRATED CHOICES

The guarded rule estimates the gains from retry and re-inspection using detection and repair measurements. Direct selection chooses the highest-support action observed in calibration. Both apply the same 3-point loss margins on supported content, correctness, and quality. The last two margins duplicate thresholded correctness on Legal (Appendix B). This extension tests whether the diagnostic measurements help choose an action. Table 35 shows fewer guard violations than always re-inspecting, with no established support advantage over direct selection. The guards have no independent utility validation.

Table 35: **Secondary policy comparison after calibration correction.** Support changes are percentage points with 95% intervals from 2,000 synchronized source-cluster draws. They use task-balanced differences of marginal means. Violation counts give rule versus reference across eight configurations. Both the gain over stopping and the unestablished advantage over direct selection are shown.

Reference policy	Rule minus reference support	Guard violations
Always stop	$+0.72$ [$+0.36$, $+1.05$]	0/8 vs. 0/8
Direct selection	-0.01 [-0.28, $+0.21$]	0/8 vs. 0/8
Always re-inspect	-0.19 [-1.14, $+0.70$]	0/8 vs. 3/8

Protocol and reporting scope. Initial decisions were written and hashed before test scoring under a protocol specifying actions, guards, splits, and two primary comparisons. A transport error counted failed calibration judgments as incorrect answers. We corrected failed calls uniformly and recomputed affected choices before viewing those configurations' test scores, although one unchanged configuration had already been previewed. The choice to report corrected-input results was made after both analyses were available, so this is not a new prospective confirmation. Initial decisions and the correction record are retained for audit.

The two primary comparisons concern guard violations against always re-inspecting and support against stopping. Both meet the prespecified Holm threshold under 2,000 source-cluster draws synchronized within task. J2 gives $+0.48$ [$+0.18$, $+0.77$] support points over stopping with the same choices. These intervals describe source-item uncertainty in the tested configurations. Eight configurations across two tasks do not establish generalization to future tasks or models.

The error-accounting predictions followed the operation ladder and preceded the selector, feedback-content, and budget comparisons. The shuffled-reason prediction fails, as does the fixed-constant rule's development regret target. Supplementary notes retain the rule equations, development results, protocol amendments, and the restricted VRR-Stop adaptation, which selects stop in all eight configurations. The original VRR-Stop interface is not tested here.

G WORKED EXAMPLES

These examples illustrate mechanisms rather than estimate their frequency. Counts and comparisons are in Appendix E.

Repairing an omitted qualification. A LegalBench-RAG question answered by Opus asks whether a receiving party may share confidential information with employees. The draft's claim c1 cites clause S3 and treats internal disclosure as permitted. Both verifiers reject it because the claim omits the need-to-know restriction. The rewrite adds that qualification while keeping S3. Re-check and held-out evaluation accept the revision, and the short answer stays yes. Deleting the flagged claim would remove it instead. The edit restores useful content by adding the missing condition. Whether a revision preserves the original scope requires a separate judgment.

Narrowing a document claim. A document answer cites page 10 for both anthrax and chickenpox vaccine information statements. The text and image checks support the anthrax reference but not the chickenpox reference on that page. The rewrite keeps anthrax and drops chickenpox. The revised claim is accepted, and the answer remains yes. A strict retention judgment rejects it as narrower than the draft. Support and same-scope retention can therefore move in opposite directions.

A second rewrite that still fails. For the Street Stream NDA question about retaining confidential information after return or destruction, a draft claim treats exclusions and compelled disclosures as evidence about a retention right. The cited clause does not establish that inference. The first and second revisions still fail verification, so the claim is removed at the round limit. Another claim in the same answer is repaired successfully. This illustrates why final support alone cannot distinguish successful editing from filtering after an unsuccessful retry.

H EXTENDED DISCUSSION

Scope of inference. Shared-prefix contrasts identify operation effects within the tested procedure. Cross-configuration traces change models and drafts, so they do not isolate recall's causal effect. Oracle rates use different error sets, and editor swaps include interface differences. Re-inspection jointly changes scope, prompt, and role (Appendix A.2). Human bounds condition on fixed cohorts and returned labels, without correcting annotation error or estimating population prevalence (Appendix C.3). Source-cluster intervals quantify uncertainty conditional on saved outputs, without independent run-level replication or an equivalence test (Appendix B). Preservation remains separately judged, with configuration-dependent human agreement (Appendix C.4).

The new self-check comparison tests a different starting point and adds the full selective loop. It does not show that unrestricted full-answer rechecking at every round would stall. Wording controls vary editing permissions while retaining re-inspection's inspection role, and therefore do not isolate scope alone. The human transfer calculation remains an assumption-dependent sensitivity analysis. Direct before/after judgments carry the human repair claims (Appendix C.5). All experiments retain the structured, short, single-citation-per-claim answer format. Free-form long answers and multi-source claims remain outside the evaluated setting.

Measure the errors available for another edit. A queue is observable before another edit, but its size is not the number of actual errors available to repair. False re-flags inflate it, while missed errors never enter it. The useful diagnostic is the relationship between the edit set and independently assessed residual errors. It requires evaluation data and cannot be treated as a perfect online signal.

Separate a guard from a guarantee. A calibration guard screens candidate actions against a chosen loss margin. Its usefulness depends on the measurement rule and the task. The held-out comparisons support evaluating support and retention together, while the secondary policy analysis leaves its advantage over direct selection unestablished (Appendix F).

What might change the result? A selector with higher recall could move errors into a retry's reach. A stronger or differently trained editor could improve repair on page images. Retrieval could

make previously unsupported claims supportable by adding evidence. The current fixed-evidence study does not rule out these possibilities.

I PROMPTS AND VERIFIER INTERFACE

The following pages reproduce the exact prompt templates, one template per page. Braced placeholders are filled with item-specific questions, evidence, answers, or verdicts. Prompt formatting here does not change the strings used in the experiments.

Prompt revision. The self-check prompt was amended to require a short answer in the draft's format, matching the restriction already imposed on the editor. Without this constraint, a document answer could contain the correct span but receive a lower ANLS score because it was expanded into a sentence. All self-check and self-check-plus-loop runs were repeated under the amended prompt. The versions reproduced here are those used for the reported comparisons.

I.1 VERIFIER INPUTS AND DECISIONS

The frozen ATOMCITE interface is shared across in-loop verification, evidence-based baselines, and held-out evaluation. The model serving that interface differs by role. Each call receives one claim and its cited source. It does not see the claim's position, the rest of the answer, or uncited sources. Legal and open-domain claims are checked against the cited clause or passage. Document claims use separate image and OCR-text channels.

A failed document citation is also checked against the two neighboring pages. This distinguishes unsupported text from a claim supported on a nearby page. Equal channel verdicts stand. A direct conflict between supported and contradicted becomes uncertain. Otherwise the image channel is authoritative, and a failed channel defers to the other.

From prompt labels to verdicts. Prompt 21 returns `SUPPORTED`, `PARTIAL`, `NOT_SUPPORTED`, or `CONTRADICTED`. Before channel merging, these map to VERIFIED, PARTIALLY_VERIFIED, NOT_VERIFIED, and FALSE, respectively. The rules above then produce the citation-level verdict used below.

Verifier verdict	Actionable status
VERIFIED	Supported.
VERIFIED_NEARBY	Wrong page. Move the citation to the supporting page.
PARTIALLY_VERIFIED, NOT_VERIFIED, FALSE, HALLUCINATION, or nonexistent citation	Unsupported and eligible for editing or removal.
INCONSISTENT or ABSTAIN	Uncertain. Kept, but not counted as supported.

Held-out support uses the same verdict-to-status mapping with a separate model and accepts only VERIFIED. It does not move citations: VERIFIED_NEARBY is not support at the supplied page. INCONSISTENT is not accepted, while ABSTAIN is excluded from support scoring. A claim's identifier is never evidence of support or content preservation.

PROMPT 1: DRAFT GENERATION: LEGAL

Generate a source-cited answer to a question about a legal clause.

Exact template. Braced fields are filled for each item.

```
You answer a question about a specific contract using only
the numbered sources below. Each source is labeled with the
document it comes from; some sources come from OTHER
contracts and are not about the document in the question.
Write the answer as a short list of claims.

Rules:
- Each claim is ONE sentence stating ONE fact or rule, and
cites exactly ONE source id in square brackets, e.g. [S3].
- After the citation, write "::" and a short reason why
that source supports the claim.
- Cite only ids from the list, and only sources from the
document named in the question. If no such source supports
a statement, do not make it.
- Write 2 to 6 claims. Do not repeat the sources verbatim;
state what they establish.
- End with a one-sentence direct answer to the question.

Output format (exactly):
ISSUE: <one sentence stating what the question asks>
1. <claim> [<source id>] :: <reason>
2. <claim> [<source id>] :: <reason>
...
ANSWER: <one sentence answer>

Question: {question}

Sources you may cite:
{evidence}
```

PROMPT 2: DRAFT GENERATION: DOCUMENTS

Answer a question using the supplied document pages.

Exact template. Braced fields are filled for each item.

```
You answer a question about the attached multi-page
document. The pages are attached IN ORDER and are numbered
from 0: the first attached image is page p0, the second is
p1, and so on.
{image_map}

Rules:
- Each claim is ONE sentence stating ONE fact taken from
the document, and cites exactly ONE page id in square
brackets, e.g. [p4].
- After the citation, write "::" and a short reason why
that page supports the claim.
- Cite only page ids p0 to p{last}. Cite the page where the
fact actually appears.
- Write 1 to 5 claims that lead to the answer. Do not make
statements the document does not support.
- End with the ANSWER: the shortest span that answers the
question (a name, number, date, or phrase), copied verbatim
from the document, with no extra words.

Output format (exactly):
ISSUE: <one sentence stating what the question asks>
1. <claim> [<page id>] :: <reason>
2. <claim> [<page id>] :: <reason>
...
ANSWER: <short answer>

Question: {question}
```

PROMPT 3: DRAFT GENERATION: OPEN-DOMAIN QA

Generate a long-form answer grounded in the supplied passages.

Exact template. Braced fields are filled for each item.

```
You answer a question using only the numbered sources below
(Wikipedia passages, each labeled with its article title).
The question may be ambiguous: it can have several valid
readings with different answers. Cover every reading that
the sources support, and give each its own answer. Write
the answer as a short list of claims.

Rules:
- Each claim is ONE sentence stating ONE fact, and cites
exactly ONE source id in square brackets, e.g. [S3].
- After the citation, write "::" and a short reason why
that source supports the claim.
- Cite only ids from the list. If no source supports a
statement, do not make it.
- Write 2 to 6 claims. Do not repeat the sources verbatim;
state what they establish.
- End with a one- or two-sentence direct answer that names
the answer for each reading of the question.

Output format (exactly):
ISSUE: <one sentence stating what the question asks, and
its readings if ambiguous>
1. <claim> [<source id>] :: <reason>
2. <claim> [<source id>] :: <reason>
...
ANSWER: <direct answer>

Question: {question}

Sources you may cite:
{evidence}
```

PROMPT 4: SELECTIVE REWRITING

Rewrite only the claims flagged by the checker. In the no-reason condition, the checker text states only that the cited source does not support the claim.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. An independent checker read the
cited source of every claim. The claims listed under FAILED
were not supported by the source they cite.

For each failed claim choose ONE action:
- "reword": keep the same source; restate the claim so that
it says only what that source actually says.
- "soften": keep the claim's point but hedge or narrow it
to what the source supports, or cite one of the allowed
sources that does support it.
- "drop": no allowed source supports it. Remove the claim.

Do not change any claim that is not listed. Do not add new
claims. Cite only the allowed source ids given for that
claim.
If your changes alter the final answer, give the revised
answer; otherwise repeat it.

Return one JSON object per failed claim, one per line, then
one final JSON object with the answer:
{{"claim_id": "<id>", "action": "reword|soften|drop",
"text": "<new claim text, empty if drop>", "cited":
"<source id>"}}
{{"answer": "<final one-sentence or short answer>"}}

Question: {question}

Current answer:
{answer}

FAILED:
{failed}

{sources}
```

PROMPT 5: FULL REDRAFTING

Revise the entire answer using the verification feedback.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. An independent checker read the
cited source of every claim. The claims listed under FAILED
were not supported by the source they cite.

Rewrite the WHOLE answer in the same format, fixing the
failed claims (restate, narrow, re-cite an allowed source,
or remove them) and keeping the rest accurate. Cite only
allowed source ids. Do not add claims that no source
supports.

Output format (exactly):
ISSUE: <one sentence>
1. <claim> [<source id>] :: <reason>
...
ANSWER: <final answer>

Question: {question}

Current answer:
{answer}

FAILED:
{failed}

{sources}
```

PROMPT 6: WHOLE-ANSWER SELF-CHECK

Inspect the answer directly and retain the required short-answer format.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. Re-read the source cited by
each claim and check, claim by claim, whether the cited
source actually supports the claim as written. Use only the
sources; do not use outside knowledge.

For every claim that is NOT fully supported by its cited
source choose ONE action:
- "reword": keep the same source; restate the claim so that
it says only what that source actually says.
- "soften": keep the claim's point but hedge or narrow it
to what the source supports, or cite a source that does
support it.
- "drop": no source supports it. Remove the claim.

Do not list claims that are fully supported. Do not add new
claims. Cite only source ids from the list.
If your changes alter the final answer, give the revised
answer; otherwise repeat it unchanged. Keep the ANSWER line
in the same form as before: a short answer, not an
explanation (for extractive questions, the shortest span
from the document).

Return one JSON object per changed claim, one per line,
then one final JSON object with the answer. If every claim
is supported, return only the answer object.
{{"claim_id": "<id>", "action": "reword|soften|drop",
"text": "<new claim text, empty if drop>", "cited":
"<source id>"}}
{{"answer": "<final one-sentence or short answer, same form
as before>"}}

Question: {question}

Current answer (claim ids in angle brackets):
{answer}

{sources}
```

PROMPT 7: SELF-REFINE: FEEDBACK

Produce feedback on the draft before the refinement call.

Exact template. Braced fields are filled for each item.

```
You are reviewing a draft answer. Read the question, the
sources, and the draft. Give feedback on the draft as a
short list: for each problem, name the claim number, say
what is wrong (the cited source does not support it, wrong
source cited, the claim is inaccurate, an important reading
of the question is missing, the ANSWER line is wrong), and
what to change. Base every point on the sources. If the
draft has no problems, output exactly: NO ISSUES.

Question: {question}

Draft:
{answer}

{sources}
```

PROMPT 8: SELF-REFINE: REVISION

Revise the draft using the preceding feedback.

Exact template. Braced fields are filled for each item.

```
Rewrite the draft answer using the feedback. Keep what the
feedback did not criticise. Use only the sources listed;
every claim cites exactly one source id in square brackets,
followed by "::" and a short reason.

Output format (exactly):
ISSUE: <one sentence stating what the question asks>
1. <claim> [<source id>] :: <reason>
2. <claim> [<source id>] :: <reason>
...
ANSWER: <direct answer>

Question: {question}

Draft:
{answer}

Feedback:
{feedback}

{sources}
```

PROMPT 9: CHAIN-OF-VERIFICATION: QUESTIONS

Create verification questions about claims in the draft.

Exact template. Braced fields are filled for each item.

```
A draft answer is given as numbered claims. For each claim,
write ONE verification question whose answer, found in the
sources, would confirm or refute the claim. Output one line
per claim in the form "<claim id>: <question>", nothing
else.

Question: {question}

Draft:
{answer}
```

PROMPT 10: CHAIN-OF-VERIFICATION: INDEPENDENT ANSWER

Answer a verification question independently before final revision.

Exact template. Braced fields are filled for each item.

```
Answer the question below using only the sources. Give a
one- or two-sentence answer and name the source id(s) it
comes from in square brackets. If the sources do not answer
it, say "Not answerable from the sources".

Question: {vquestion}

{sources}
```

PROMPT 11: CHAIN-OF-VERIFICATION: REVISION

Revise the answer using the independently answered verification questions.

Exact template. Braced fields are filled for each item.

```
Revise the draft answer using the verification results.
Each result is a verification question about one claim and
its answer found independently in the sources. Fix or drop
claims the results contradict, fix wrong source ids, keep
claims the results confirm, and update the ANSWER line.
Every claim cites exactly one source id in square brackets,
followed by "::" and a short reason.

Output format (exactly):
ISSUE: <one sentence stating what the question asks>
1. <claim> [<source id>] :: <reason>
2. <claim> [<source id>] :: <reason>
...
ANSWER: <direct answer>

Question: {question}

Draft:
{answer}

Verification results:
{results}

{sources}
```

PROMPT 12: RARR: ATTRIBUTION

Identify unsupported claims and explain the attribution problems.

Exact template. Braced fields are filled for each item.

```
For each claim of the draft, find the source that best
supports it and decide whether that source actually
supports the claim as written. Output one JSON object per
line, one per claim, nothing else:
{{"claim_id": "<id>", "best_source": "<source id>",
"supported": "yes" or "no", "note": "<one line: what the
source says, or what is wrong>"}}

Question: {question}

Draft:
{answer}

{sources}
```

PROMPT 13: RARR: EDITING

Edit using the attribution report. The verifier-matched RARR variant uses this same prompt with the report replaced by the frozen verifier's failed claims and reasons.

Exact template. Braced fields are filled for each item.

```
Some claims of the draft are not supported by their best
source (attribution report below). Edit each listed claim
so that it agrees with what the sources say, changing as
little as possible, or drop it if no source supports it. Do
not touch the other claims. Output one JSON object per line
for each listed claim, nothing else:
{{"claim_id": "<id>", "action": "reword" or "drop", "text":
"<new claim text, or empty for drop>", "cited": "<source
id>"}}
You may also output one line {{"answer": "<revised ANSWER
line>"}} if the ANSWER line must change.

Question: {question}

Draft:
{answer}

Attribution report (unsupported claims):
{report}

{sources}
```

PROMPT 14: VERICITE: SOURCE UTILITY

Assess whether a source is useful for answering the question.

Exact template. Braced fields are filled for each item.

```
For each source below, decide whether it contains
information useful for answering the question
(VeriCite-style utility check). A source is useful if it
states a fact that answers the question or part of it; a
source that is only on the same topic is not useful.

Output one line per source id, nothing else:
<source id>: useful
<source id>: not useful

Question: {question}

{sources}
```

PROMPT 15: VERICITE: EVIDENCE EXTRACTION

Extract relevant statements from one source. The extracted statements are then verified.

Exact template. Braced fields are filled for each item.

```
Using ONLY the single source below, write the statements
that answer the question or part of it (VeriCite-style
per-source answer). Every statement must be directly stated
by this source; do not use outside knowledge and do not use
other sources. If the source answers nothing, output
exactly: NONE

Output format (exactly):
1. <statement>
2. <statement>
...

Question: {question}

{source}
```

PROMPT 16: VERICITE: SYNTHESIS

Synthesize the answer from the retained evidence statements.

Exact template. Braced fields are filled for each item.

```
You are the synthesis stage of a VeriCite-style pipeline.
Every statement below has been verified against the source
it cites: the verified draft claims come from the initial
answer, and the verified evidence statements were extracted
from individual sources. Write the final answer from these
statements only: reorganise them into a coherent answer,
remove redundancy (merge statements that say the same thing
and keep the citation of the one you keep), and keep every
citation as given. Do not add facts that are not in the
statements, do not change what a statement asserts, and
cite only the source ids shown.

Output format (exactly):
ISSUE: <one sentence>
1. <claim> [<source id>]
...
ANSWER: <final answer>

Question: {question}

Verified draft claims:
{draft}

Verified evidence statements:
{evidence}
```

PROMPT 17: DELETE-AND-REGENERATE ABLATION

Regenerate an answer after failed draft claims have been removed.

Exact template. Braced fields are filled for each item.

```
An independent checker has verified every claim of the
draft below against its cited source; the claims that
failed have already been removed. Write the final answer
from the verified claims and the sources: keep every
verified claim and its citation, integrate them into a
coherent answer, and you may add a claim only if a source
states it directly (cite that source). Do not weaken or
drop verified claims. Cite only allowed source ids.

Output format (exactly):
ISSUE: <one sentence>
1. <claim> [<source id>]
...
ANSWER: <final answer>

Question: {question}

Verified draft:
{answer}

{sources}
```

PROMPT 18: REACT: NEXT ACTION

Choose one tool action at a time. The visible transcript grows after each action.

Exact template. Braced fields are filled for each item.

```
You are an agent improving a draft answer so that every
claim is supported by the source it cites. You have these
actions:
- verify(<claim id>): an independent checker reads the
cited source and reports whether it supports the claim,
with a one-line reason.
- read(<source id>): shows the text of a source.
- edit(<claim id> | <new claim text> | <source id>):
rewrites a claim and its citation.
- drop(<claim id>): removes a claim.
- answer(<text>): replaces the ANSWER line.
- finish: stop and return the current answer.
Use at most {budget} actions. Reply with exactly one
"Thought:" line followed by one "Action:" line each turn,
and nothing after the Action line. You decide which claims
to verify and what to change.

Question: {question}

{sources}

Current answer:
{answer}
{transcript}
```

PROMPT 19: LEGAL-ANSWER EVALUATION

Judge task-level correctness separately from claim-level source support.

Exact template. Braced fields are filled for each item.

```
You are evaluating the CORRECTNESS of an answer to a
question about a legal document.

QUESTION:
{question}

GROUND TRUTH (the clause of the document that answers the
question, or the reference answer):
{ground_truth}

CANDIDATE ANSWER:
{candidate}

Score the candidate on a 1-5 scale:
  5 = Reaches the same conclusion as the ground truth
  supports, with sound reasoning.
  4 = Reaches the same conclusion; minor reasoning gaps or
  one weak or irrelevant statement.
  3 = Reaches the same conclusion but with notable
  reasoning errors or two or more weak statements.
  2 = Wrong or missing conclusion but partially correct
  reasoning.
  1 = Wrong conclusion and wrong reasoning, or no answer.

When the question asks whether the document permits,
requires, or prohibits something, judge whether the
candidate's operative yes/no holding matches what the
ground-truth clause supports. When the question asks for a
fact (a party, a date, a term), judge whether the candidate
states the same fact.

Output JSON only:
{{"score": <1-5>, "rationale": "<one sentence>"}}
```

PROMPT 20: CONTENT-PRESERVATION EVALUATION

Compare the content of draft claims with the final answer. Claim identifiers do not establish that their content survives.

Exact template. Braced fields are filled for each item.

```
You are checking which statements from a DRAFT answer are
still asserted in a FINAL answer. The final answer may
reword, merge, split, quote, or reorder statements; what
matters is whether the same factual content (the same claim
about the same thing) is still stated. A claim that was
weakened to a vague statement, dropped, or contradicted is
NOT kept.

QUESTION:
{question}

FINAL ANSWER:
{final}

DRAFT CLAIMS:
{claims}

For every draft claim id, decide whether its content is
still asserted in the final answer.

Output JSON only:
{{"kept": {{"<id>": true|false, ...}}}}
```

PROMPT 21: CLAIM-SOURCE VERIFICATION

Check a single claim against its cited source, using the text or page-image channel.

Exact template. Braced fields are filled for each item.

```
You are verifying whether a cited source supports one
atomic claim.

ATOMIC CLAIM:
{atom}

CITED SOURCE ({channel} of {target_desc}):
{evidence}

Judge ONLY from the cited source above. Do not use outside
knowledge to bridge
gaps. Every operative element of the claim (entities,
numbers, dates,
qualifiers) must be present or directly entailed for full
support.

Answer with STRICT JSON, nothing else:
{"verdict": "SUPPORTED" | "PARTIAL" | "NOT_SUPPORTED" |
"CONTRADICTED",
 "quote": "<the exact supporting/contradicting span, or
 empty>",
 "rationale": "<= 25 words"}
```

PROMPT 22: NEIGHBORING-PAGE VERIFICATION

For a failed document citation, check whether a neighboring page supplies support.

Exact template. Braced fields are filled for each item.

```
An atomic claim was cited to {cited_target}, but may
actually be supported by
a nearby page. For each candidate page below, say whether
it fully supports
the claim.

ATOMIC CLAIM:
{atom}

CANDIDATE PAGES:
{pages}

Answer with STRICT JSON, nothing else:
{"supporting_pages": ["<page number>", ...]}
Empty list if none fully supports it.
```

PROMPT 23: RE-INSPECTION OF ACCEPTED CLAIMS

Re-examine only accepted claims that have never entered an edit queue. The editor sees the answer context but may change only the listed ids.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. Re-check ONLY the claims with
these ids: {ids}. For each of them, re-read the source it
cites and check whether the cited source actually supports
the claim as written. Leave every other claim unchanged.
Use only the sources; do not use outside knowledge.

For every claim that is NOT fully supported by its cited
source choose ONE action:
- "reword": keep the same source; restate the claim so that
it says only what that source actually says.
- "soften": keep the claim's point but hedge or narrow it
to what the source supports, or cite a source that does
support it.
- "drop": no source supports it. Remove the claim.

Do not list claims that are fully supported. Do not add new
claims. Cite only source ids from the list.
If your changes alter the final answer, give the revised
answer; otherwise repeat it unchanged. Keep the ANSWER line
in the same form as before: a short answer, not an
explanation (for extractive questions, the shortest span
from the document).

Return one JSON object per changed claim, one per line,
then one final JSON object with the answer. If every claim
is supported, return only the answer object.
{{"claim_id": "<id>", "action": "reword|soften|drop",
"text": "<new claim text, empty if drop>", "cited":
"<source id>"}}
{{"answer": "<final one-sentence or short answer, same form
as before>"}}

Question: {question}

Current answer (claim ids in angle brackets):
{answer}

{sources}
```

PROMPT 24: SAME-SCOPE PRESERVATION EVALUATION

Judge whether the original statement remains asserted at the same scope. This judgment is separate from Prompt 20. Qualification can improve support while failing this strict preservation rule.

Exact template. Braced fields are filled for each item.

```
You are checking which statements from a DRAFT answer are
still asserted in a FINAL answer, under a STRICT rule. The
final answer may reword, merge, split, quote, or reorder
statements; what matters is whether the SAME statement,
with the SAME scope, is still made.

Under the strict rule a draft claim is NOT kept if the
final answer:
- drops it or contradicts it;
- weakens it to a vaguer statement;
- narrows or qualifies it: adds a condition, exception,
limitation or "need-to-know"-style restriction, restricts
the set of things it applies to, or changes an
unconditional statement into a conditional one;
- broadens it or changes what it is about.
Only a restatement that asserts the same content with the
same scope counts as kept. A more precise, more correct, or
better-hedged version of the claim is NOT kept under this
rule (that is a correction, and corrections are counted
elsewhere).

QUESTION:
{question}

FINAL ANSWER:
{final}

DRAFT CLAIMS:
{claims}

For every draft claim id, decide whether the same statement
with the same scope is still asserted in the final answer.

Output JSON only:
{{"kept": {{"<id>": true|false, ...}}}}
```

PROMPT 25: RE-INSPECTION WITH QUALIFICATION ONLY

Qualify eligible claims without changing the source or removing claims. The ordinary verifier checks the edited claims afterward.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. Re-check ONLY the claims with
these ids: {ids}. For each of them, re-read the source it
cites and check whether the cited source actually supports
the claim as written. Leave every other claim unchanged.
Use only the sources; do not use outside knowledge.

For every claim that is NOT fully supported by its cited
source there is ONE action:
- "soften": keep the claim's point and the same source;
hedge, qualify or narrow the claim to what that source
supports (add the condition, exception or limitation the
source states, or weaken an unconditional statement). Do
not restate it with different content, do not change the
source, and do not remove it.

Do not list claims that are fully supported. Do not add or
remove claims. Keep each claim's source id.
If your changes alter the final answer, give the revised
answer; otherwise repeat it unchanged. Keep the ANSWER line
in the same form as before: a short answer, not an
explanation (for extractive questions, the shortest span
from the document).

Return one JSON object per changed claim, one per line,
then one final JSON object with the answer. If every claim
is supported, return only the answer object.
{{"claim_id": "<id>", "action": "soften", "text": "<new
claim text>", "cited": "<same source id>"}}
{{"answer": "<final one-sentence or short answer, same form
as before>"}}

Question: {question}

Current answer (claim ids in angle brackets):
{answer}

{sources}
```

PROMPT 26: RE-INSPECTION WITH REWORDING OR REMOVAL

Reword or remove eligible claims, with instructions against hedging or qualification.

Exact template. Braced fields are filled for each item.

```
You wrote the answer below. Re-check ONLY the claims with
these ids: {ids}. For each of them, re-read the source it
cites and check whether the cited source actually supports
the claim as written. Leave every other claim unchanged.
Use only the sources; do not use outside knowledge.

For every claim that is NOT fully supported by its cited
source choose ONE action:
- "reword": keep the same source; restate the claim as a
plain statement of what that source actually says (correct
the fact, figure, party or condition). Do not hedge or
qualify: no "may", "generally", "in some cases", "appears",
"typically" or similar words, and do not turn a statement
into a conditional one.
- "drop": no source supports it. Remove the claim.

Do not list claims that are fully supported. Do not add new
claims. Cite only source ids from the list.
If your changes alter the final answer, give the revised
answer; otherwise repeat it unchanged. Keep the ANSWER line
in the same form as before: a short answer, not an
explanation (for extractive questions, the shortest span
from the document).

Return one JSON object per changed claim, one per line,
then one final JSON object with the answer. If every claim
is supported, return only the answer object.
{{"claim_id": "<id>", "action": "reword|drop", "text":
"<new claim text, empty if drop>", "cited": "<source id>"}}
{{"answer": "<final one-sentence or short answer, same form
as before>"}}

Question: {question}

Current answer (claim ids in angle brackets):
{answer}

{sources}
```

PROMPT 27: SUPPORT EVALUATION WITH THE HUMAN RULE

Apply the human element-by-element support rule to one claim and its cited evidence. This measurement check does not replace the primary judge.

Exact template. Braced fields are filled for each item.

```
You are checking one claim against the single source it
cites. Use only the source shown; do not use outside
knowledge. The question the claim was written to help
answer is shown for context only.

S1: is the claim, as written, supported by that source
alone? Decide element by element: split the claim into its
factual elements (which party, what is asserted, which
condition or exception, which number, date or name).
  fully = every element is stated in the source or follows
  directly from it. Paraphrase, reordering and leaving out
  source content that does not change what the claim
  asserts never make a claim less than fully supported.
  partly = at least one element is not in the source, or
  the source states a condition, exception, limit or
  different figure that the claim drops, overstates or
  mis-states, while the gist is supported.
  not_supported = the source does not support the claim or
  contradicts it.
  insufficient_evidence = the source is unreadable,
  missing, or does not concern the claim at all.
S2: the main reason when S1 is not "fully": wrong_fact (the
source says something else), missing_condition (a
condition, exception or qualifier in the source is
dropped), over_strong_inference (the claim goes beyond what
the source says), wrong_citation (the source is about
something else; the claim may be true elsewhere),
cannot_judge; "none" when S1 is fully.

Question: {question}

Claim: {claim}

Source ({cited}):
{evidence}
{page_note}
Output JSON only:
{{"S1": "fully|partly|not_supported|insufficient_evidence",
"S2": "wrong_fact|missing_condition|over_strong_inference|w
rong_citation|cannot_judge|none", "note": "<one
sentence>"}}
```