THE MISSING COMPLEMENT: STATE-CONDITIONED MINIMAL SUFFICIENT EVIDENCE FOR CODING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval assembles repository context by ranking passages for relevance to the current query. A coding agent halfway through an issue has already read much of what such a ranker returns. Relevance is scored per passage, but sufficiency belongs to the set: independently scored passages can fill the budget with support for one requirement while another goes unmet. We formulate *state-conditioned minimal sufficient evidence recovery*: given a captured agent state, recover a compact evidence combination supplying what its next decision still lacks. SERBENCH measures this on 500 held-out states from 45 repositories, recording what the agent has seen, crediting only sets that satisfy every annotated evidence requirement of the current decision, and separating set recovery from candidate discovery. MSS-COMPLEMENT treats acquisition as set construction, not ranking. Three semantic calls propose a jointly sufficient set, search for what it lacks, and return 4–8 intact source units within 6,144 tokens. One configuration, fixed on calibration data, recovers a complete set for 73.0% of those states at five items and 80.6% at eight, against 61.4% and 72.4% for Qwen3 embedding with reranking. A matched control ranking by similarity alone recovers fewer complete sets, placing the margin over it in the set-level policy, not the computation. The lead persists from frozen repository source with no gold-derived pool. On AMA-Bench it answers from a 76.2% smaller answer prompt, with accuracy 2.08 points above that benchmark's own memory agent. Removing one required group from a complete set costs repair-localization precision under two executors. Retrieval for agents is better posed as recovering what a decision lacks than re-ranking what an issue resembles.

1 INTRODUCTION

A coding agent searches for a symbol, opens the implementation it finds, watches a test fail, and revises its hypothesis (Jimenez et al., 2024; Yang et al., 2024; Xia et al., 2024). The issue never changes; what the agent needs to see next changes at every step. The site that mattered a moment ago is now history, and a caller or configuration branch is the missing fact. What the next decision still needs is what the trajectory has not established, and that target moves even when the query does not.

One response is to keep everything, but addressability is not selection. Lost in the Middle shows that use depends on where supporting text sits; RULER and NoLiMa separate nominal window size from what a model sustains (Liu et al., 2024; Hsieh et al., 2024; Modarressi et al., 2025). At each step, the agent must select evidence from a growing history under a fixed context budget. The other response is to retrieve (Lewis et al., 2020). Embedding, reranking, and software-specific retrievers judge relevance to an issue accurately (Zhang et al., 2025b; Shao et al., 2025; Reddy et al., 2026), and agent-aware variants condition on the reasoning trace (Chen et al., 2026). Two properties of a decision-time query stay outside that objective. The first is redundancy: conditioning relevance on the reasoning trace does not by itself require the returned evidence to add anything beyond what the agent has already observed. A passage can therefore be maximally relevant and add nothing because the agent already has it, and in a 21-state pilot 53.3% of an unmasked state-conditioned query's top ten is already observed. The second is joint sufficiency: independent relevance scores carry no information about which requirements the selected evidence already covers, so the ordering they induce says nothing about whether the set is sufficient. A repair turning on a call site, its caller, and a configuration constraint needs all three, and a ranker may spend the budget on variants of the first. Figure 1 shows both.

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

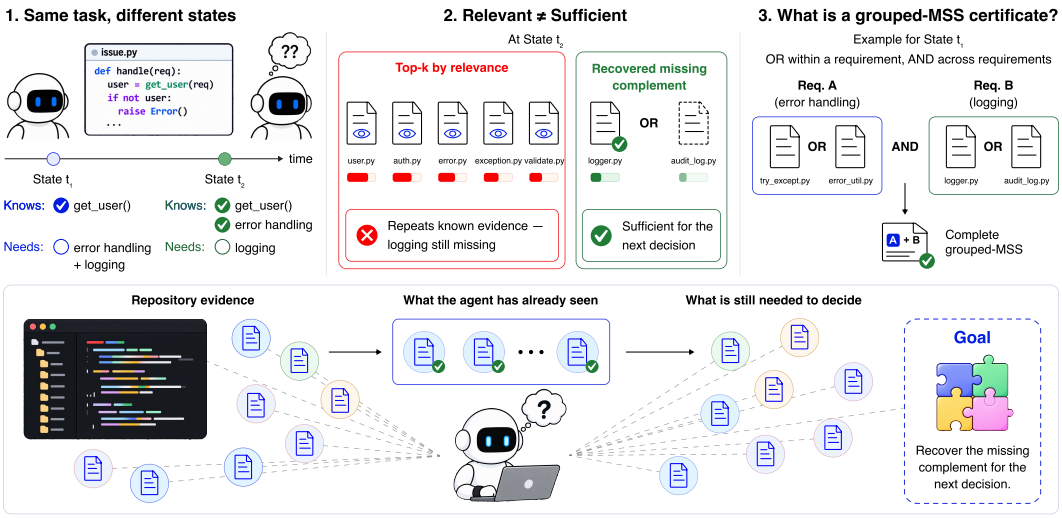

Figure 1: The missing evidence changes with the decision state. At t_1, the task needs error-handling and logging evidence; at t_2, only logging remains unresolved. The t_1 certificate accepts either source for each requirement and requires both requirements to be satisfied. The lower panel distinguishes the growing evidence space, observed evidence, and the complement needed for the next decision.

We call the resulting target *state-conditioned minimal sufficient evidence recovery*. Acquisition is conditioned on the visible state and judged on whether the returned combination covers what the next decision still lacks. Sources may substitute inside a requirement, while distinct requirements must be covered together, so additional evidence for one requirement cannot compensate for leaving another unmet. Minimality constrains the certificate, not the returned set: a requirement is recorded only if removing it would weaken the decision's support (Section 3). Small working contexts and sufficiency checks are both familiar; neither has been anchored to a decision state whose observations are recorded and a certificate fixing which sets count.

Repository-retrieval benchmarks score retrieval against an issue or workflow signal. Evaluating this target instead requires a captured decision state and an explicit specification of which evidence combinations satisfy its unresolved requirements. SERBENCH provides that setting. It turns trajectory prefixes into portable state cards, records their observed evidence, and scores returned prefixes against private grouped certificates of unresolved requirements; the endpoint, Complete-MSS@k, gives partial support zero credit. Cal500 supplies 500 development states, and Test500 another 500 from 45 repositories disjoint from Cal500's. Released candidate pools isolate set recovery, and a gold-blind track from frozen source measures the discovery stage supplying them.

To recover such sets under a fixed context budget, we introduce MSS-COMPLEMENT. It follows from the formulation, so every decision it makes is about a set. Three semantic calls over a fused ranking propose a jointly sufficient set, search deeper candidates for what it lacks, and finalize 4–8 intact source units under a 6,144-token ceiling. Expansion and finalization read the current selection, which lets the method ask what is missing, not what is similar. One controller configuration, fixed on Cal500, transfers unchanged to every main evaluation.

On Test500, MSS-COMPLEMENT recovers a complete evidence set for 73.00% of states at five items, against 61.40% for Qwen3-Embedding-8B with Qwen3-Reranker-8B, and 80.60% against 72.40% at eight. The gain concentrates where the formulation predicts: 14.79 points on the 284 states requiring at least two groups. A control keeping the three calls and the source ceiling but ranking by independent similarity reaches 66.60%, placing 6.40 points with the set-level policy. The advantage carries past the released pools to gold-blind discovery, at 5.00 points. On AMA-Bench it answers from an answer prompt 76.2% smaller, at 2.08 points higher accuracy, and on Action52 its evidence localizes repairs more precisely than the reranked baseline under two executors.

We define state-conditioned minimal sufficient evidence recovery, introduce SERBENCH to evaluate complete-set recovery at captured decision states, and present MSS-COMPLEMENT, evaluated under matched controls, gold-blind repository discovery, and downstream repair localization.

2 Related Work

Repository context for coding agents. SWE-bench, SWE-agent, and Agentless evaluate or perform repository-level issue resolution, and RepoCoder studies repository-level completion by iterating retrieval and generation (Jimenez et al., 2024; Yang et al., 2024; Xia et al., 2024; Zhang et al., 2023). ContextBench, Agent Retrieval Bench, CORE-Bench, and SWE-Explore evaluate context and file retrieval from issue or workflow signals (Li et al., 2026a; Qin & Xie, 2026; Zhang et al., 2026a;b). SERBench specifies an exact intermediate decision, records the evidence established by its trajectory, and scores joint coverage of its remaining requirements.

Reasoning-aware and set retrieval. Qwen3 Embedding, ReasonIR, and SweRank study general embedding and reranking, reasoning-intensive retrieval, and software-issue localization (Zhang et al., 2025b; Shao et al., 2025; Reddy et al., 2026). AgentIR uses retrieval intent from an agent's reasoning trace (Chen et al., 2026). Set-aware acquisition includes NEST's selection under retrieval noise, S2G-RAG's sufficiency prediction and evidence-gap generation, Evidence Tree Search's combination search, and search control in RAAC and AutoSearch (Verma et al., 2026; Li et al., 2026b; Sun et al., 2025; Soudani et al., 2026; Sun et al., 2026). PACE prioritizes complementary evidence and adapts reranking depth in multi-hop QA (Cai & Zafarani, 2026). Evidence ranking and sufficiency verification also appear in fact verification and long-video QA (Alt et al., 2026; Yan et al., 2026). Our setting couples set-level acquisition to the unresolved requirements of a captured coding-agent state.

Agent memory and context construction. Generative Agents and MemGPT manage long interaction histories, while LongMemEval, LoCoMo, and MemoryAgentBench evaluate conversational or incremental memory (Park et al., 2023; Packer et al., 2023; Wu et al., 2025; Maharana et al., 2024; Hu et al., 2026a). AMA-Bench evaluates long agent-environment trajectories, and τ-Knowledge connects knowledge retrieval to tool-mediated outcomes (Zhao et al., 2026b; Shi et al., 2026). Structured memory methods organize software experience, revise hierarchical memories, or provide tree access (Shen et al., 2026; Hu et al., 2026b; Liu et al., 2026). Router-Mem routes between memory-processing paths using evidence sufficiency, and MESA selects and fuses query-specific memory views (Lin et al., 2026; Zhao et al., 2026a). SERBench measures the source-evidence combination still needed at a decision, and MSS-Complement supplies intact sources as the working context. Appendix A expands the comparison, including long-context evaluation and retrieval foundations.

3 State-Conditioned Evidence Recovery

A decision's working context is its visible state plus the source evidence acquired to support it. We budget that evidence separately from the stored trajectory and from retrieval-controller tokens. Let $\mathcal{C}_r$ be repository r's frozen evidence corpus. At decision time t, the normalized state is $s_t = (x, n_t, h_t, O_t, \psi_t)$: x is the issue, n_t the current information need, h_t the visible-trajectory summary, $O_t \subseteq \mathcal{C}_r$ the observed source evidence, and ψ_t the hypothesis and subgoal. Let $\mathcal{V}_t \subseteq \mathcal{C}_r$ denote the candidate pool available at the decision, and $\mathcal{U}_t = \mathcal{C}_r \backslash O_t$ the unobserved evidence universe. A method returns an ordered sequence $\mathcal{R}_t = (e_1, \ldots, e_{L_t})$ from $\mathcal{V}_t$, subject to an item limit $L_t \leq K$ and a source-token budget B. Writing $\ell(e)$ for a unit's rendered source length, the delivered context admits units in returned order while $\sum \ell(e) \leq B$; a unit that would exceed B is dropped rather than truncated, so every scored unit stays intact. The pool may overlap O_t. Write $\mathcal{P}_t(k) = \{e_1, \ldots, e_{\min(k, L_t)}\}$ for its first k available items. Prefixes are taken from the returned order without removing observed units or backfilling their positions. Observed units cost budget without adding certificate coverage.

Each state has a grouped certificate $\mathcal{G}_t = \{g_1, \ldots, g_{m_t}\}$. Group $g_j = (A_j, q_j, w_j, \rho_j)$ specifies acceptable evidence IDs $A_j \subseteq \mathcal{U}_t$, their required count q_j, a necessity weight w_j, and a semantic role ρ_j. Each unresolved evidence requirement is represented by one certificate group g_j. Each certificate specifies the support still missing at the captured decision: its acceptable evidence and complete alternative branches are drawn from the unobserved portion of the corpus. Its coverage is

$$c_j(\mathcal{R}_t, k) = \mathbb{I}[|\mathcal{P}_t(k) \cap A_j| \geq q_j]\,. \tag{1}$$

The threshold applies within each group and reduces to OR when $q_j = 1$. All required groups are conjoined, so completion is

$$C_t(k) = \prod_{j=1}^{m_t} c_j(\mathcal{R}_t, k). \tag{2}$$

A certificate also enumerates its complete alternative solutions as a family $\mathcal{B}_t$ of evidence sets. Each branch is an inclusion-minimal witness derived from the grouped certificate: every member belongs to at least one acceptable set, and every branch jointly satisfies all group thresholds. A branch therefore records one sufficient combination rather than widening what counts as complete. The released scorer accepts a recovered branch directly, which keeps certificates whose branches are not derived this way scorable without a schema change. Minimality belongs to the certificate's decision-specific requirements, so the metric asks retrieval to cover a sufficient combination within budget, not to return a provably minimal set.

Over N evaluation states, the primary endpoint is

$$\text{Complete-MSS@}k = \frac{1}{N}\sum_{t=1}^{N} C_t(k). \tag{3}$$

Group recall and necessity-weighted recall measure partial coverage. Appendix D specifies these metrics, grouped nDCG, and prediction validation. The supporting evaluations distinguish evidence recovery from repair localization, with the state, model, tools, repository, and evaluator held fixed while the evidence changes.

4 SERBENCH

SERBENCH fixes a coding agent's visible state and evaluates whether retrieval recovers the source evidence needed for its next decision. Figure 2 connects state construction, observed-evidence alignment, and grouped certificates to the recovery and downstream evaluations.

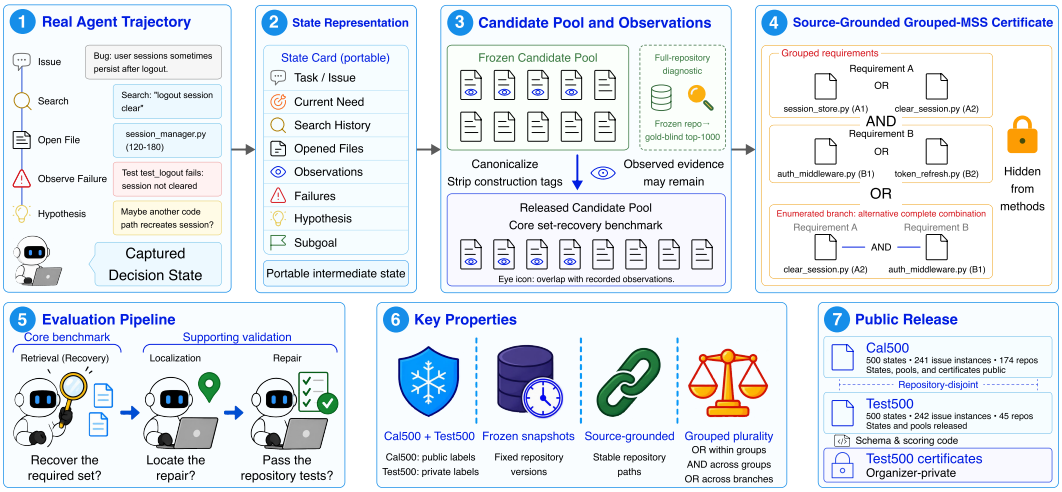

Figure 2: SERBENCH construction and evaluation. A trajectory prefix becomes a portable state card, with observed source evidence aligned to repository units. Private certificates specify the support the decision still requires, encoding requirement coverage and complete alternative branches. Released pools isolate set recovery, while a gold-blind track tests candidate reachability.

4.1 EXACT DECISION STATES

Test500 contains 500 states from 242 real issue instances across 45 repositories. It captures four decision boundaries: before search (200 states), after search (124), after file inspection (104), and before an intended edit (72). Each state has a reconstructable repository snapshot, issue, visible tool history, and source observations.

Trajectories come from an instrumented coding-agent harness run over repository snapshots frozen at each issue's base commit. The agent plans, searches the repository, inspects source files, and regenerates its information need, hypothesis, and subgoal after every observation; the prefix up to one of the four boundaries above is one state. Gold patches, test patches, certificates, and method outputs are never exposed while a trajectory runs. Prefixes are normalized into portable state cards

carrying the current need, searches, opened files, observations, failures, hypothesis, and subgoal. Agent-authored plans remain visible context, while captured tool outputs determine which source evidence has already been acquired. A file path named in a plan therefore enters the observed set only after source confirmation. Appendix B gives the harness configuration, schema, and selection criteria. Repository source is frozen at the trajectory base commit and segmented into provenance-preserving units. File reads and search results are aligned to these units to record O_t. Test500 evaluates each frozen candidate pool as released, retaining observed units where they occur. In a 21-state pilot, 53.3% of an unmasked state-conditioned query's top ten repeated evidence already visible to the agent, against 18.1% for issue retrieval.

4.2 Grouped Sufficiency Certificates

Annotators identify the decision's unresolved requirements, ground each in source units, and specify acceptable alternatives and coverage thresholds. A requirement is retained when its removal would weaken support for the next grounded action. Minimal sufficiency is local to the decision, allowing a repository to support several valid solutions. Groups capture jointly required facts (Equation (2)), while a complete alternative branch records one sufficient combination and counts only when recovered in full. Test500 includes 124 states with multiple enumerated branches. In the certificate panel of Figure 2, sources substitute within requirement A, requirement B must also be covered, and the enumerated branch covers both at once. Appendix C gives the annotation object, branch counts, and review protocol.

4.3 Validation and Release Protocol

Cross-family model audits provide review signals, followed by independent review of all 500 states by two human experts and source-grounded arbitration by a senior expert blinded to the model decisions. Two further experts outside that chain independently audit 80 uniformly sampled release states; they jointly accept 77 of 80 (96.25%) and agree on all eight assessed fields for 151 of 155 groups (97.42%). Appendix C covers construction and adjudication, and Appendix C.4 the audit.

Released candidate pools contain 20–120 source units, mean 81.01. Construction combines gold-blind repository retrieval with units located through subsequent trajectories, gold patches, and test patches, then removes construction-source tags. Every private certificate is nonempty and grounded in its public pool. These recall-complete pools isolate evidence-set recovery. The separate full-repository diagnostic tests discovery from the visible state and frozen source alone. Cal500 provides 500 labeled development states from 174 repositories disjoint from Test500's. Test500 publishes state cards, bounded candidate excerpts, schemas, scoring code, and provenance fields, while certificates and evidence roles stay in a separate organizer evaluator that scores completed predictions.

5 MSS-Complement

MSS-Complement receives a visible state, the supplied candidate pool, and a source budget, and returns an ordered evidence set for the next decision. Fused candidates pass through three semantic calls that propose a set, find its missing support, and finalize the delivered context (Figure 3). Each call evaluates joint sufficiency, and only the finalized source units reach the downstream model.

5.1 State-Conditioned Candidate Pool

The state card supplies the issue, current need, and visible trajectory context. On Test500, reciprocal-rank fusion combines nine lexical, dense, entity, and recency views over the frozen public candidate pool into a deterministic ranking for the frozen state and repository. The pool supplies breadth; the semantic calls decide joint support. Appendix E specifies the retrieval routes, depths, and pool cap.

5.2 Three-Stage Complement Acquisition

The proposal call uses DeepSeek-v4-pro, abbreviated Pro. It reads the state, current question, and leading source cards, then selects an eight-unit set S_0 covering the entities, values, conditions, time points, or causal links required for a precise answer. The expansion call uses DeepSeek-v4-flash, abbreviated Flash. It receives S_0 and further candidate summaries excluding units already selected. It seeks additional sources supporting facts absent from the proposal, including missing endpoints,

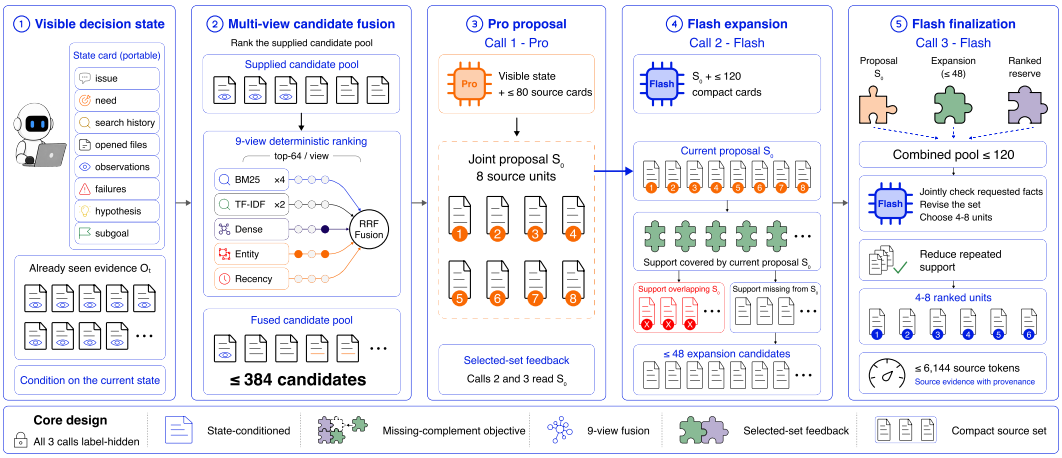

Figure 3: MSS-COMPLEMENT's three semantic stages; expansion and finalization read the current selection. Selected-set feedback concerns evidence acquired within the three-stage run. The intermediate pools are controller inputs; the downstream model receives only the final intact source set.

conditions, and transitions. Flash finalization reads the proposal, expansion candidates, and a ranked reserve, with each candidate tagged by origin. It checks their support jointly, revises the selection, and returns 4–8 ranked units, removing repeated support where possible. The renderer enforces the 6,144-source-token ceiling. The final context is intact source evidence with its provenance preserved.

5.3 OPERATING POINT AND INFORMATION ACCESS

Controller allocation, prompts, candidate-pool and stage-input limits, and output sizes are fixed on Cal500 and transferred to the primary Test500, repository-discovery, AMA, prospective, and action evaluations. Declared ablations and controller-family comparisons vary their stated factors separately. Appendix E records stage capacities, card limits, validation, fallback behavior, and configuration identities. Acquisition uses the public state and candidate source evidence. Each cohort supplies its own candidate inputs, whose construction and observed-evidence handling are documented separately. It receives no certificate, required group count, gold evidence ID, answer, reward, or judge output. Private certificates enter only when the organizer scores completed predictions.

6 EXPERIMENTS

Section 6.2 measures recovery of a complete evidence set under a tight budget. Section 6.3 asks whether completing that certificate changes what an agent does next, at a repair-localization decision on Action52 and at executed repository tests on Fresh23. Section 6.4 varies the discovery stage, task family, and controller family. Section 6.5 attributes the gain under matched budgets.

6.1 COMMON SETUP

All Test500 methods are evaluated on the same public states and frozen candidate pools. External retrievers choose their query representation on a repository-disjoint 75-state subset of Cal500, and MSS-COMPLEMENT uses the Cal500 configuration of Section 5. The supporting evaluations also use SSR, our one-pass semantic set-selection comparator (Appendix D.2). The primary endpoint is Complete-MSS@5, with group and necessity-weighted recall, larger prefixes, and matched source-token budgets. All intervals below are 95%; those for Test500 and repository discovery come from 20,000 paired resamples clustered by repository. Model endpoints, cohort-specific resampling, seeds, and trace accounting are in Appendices D.1, E, F.2, and F.4.

6.2 RECOVERING A COMPLETE EVIDENCE SET UNDER TIGHT BUDGETS

At five source units, MSS-COMPLEMENT reaches 73.00% Complete-MSS, 11.60 points above Qwen3-Embedding-8B plus Qwen3-Reranker-8B, $[+6.89, +17.21]$ (Table 1). Group recall improves by 9.57 points and necessity-weighted recall by 9.51, both intervals excluding zero. The same five

slots thus cover more unresolved requirements and complete a combination more often. Under a matched three-call budget, removing the set-sufficiency objective, selected-set feedback, and adaptive sizing costs 6.40 points, so the gain is not a by-product of spending controller calls (Section 6.5). These pools isolate selection from discovery, and the headroom inside them is wide: uniform selection of five units recovers 5.17%, and 38 points separate the strongest baseline from the oracle ceiling, of which MSS-COMPLEMENT closes 11.60 (Appendix B).

At eight items the gain is 8.20 points, $[+4.91, +13.85]$, and matching each state's source tokens gives 11.40 points. Appendix D gives the full prefix and token-budget comparisons. The gain is larger where several requirements must be supported together: at five items it is 14.09 points on the 149 states needing at least three groups, $[+6.25, +21.67]$, against 7.41 on the 216 single-group states. Appendix D.5 reports every stratum at both budgets.

Partitioning Test500 by recorded candidate/observation ID overlap, MSS-COMPLEMENT ranks first among the twelve primary methods on all five metrics in both strata, leading Qwen3 embedding with reranking by 8.28 points of Complete-MSS@5 in the 314 states without overlap and 17.20 in the 186 with it. The strata differ in trajectory stage, so the wider lead under overlap is descriptive rather than causal (Appendix D.4).

Table 1: SERBENCH-Test500 at five and eight source units. Every method covers all 500 states. MMR's calibrated coefficient of 1.0 recovers the reranker ordering, so the two rows coincide (Appendix D.2). Later tables abbreviate Qwen3-Embed + Reranker-8B as Qwen3 + Reranker.

	Complete		Group recall		Necessity
Method	@5	@8	@5	@8	@5
Lexical, dense, sparse, and late interaction					
BM25	9.00	13.00	13.12	18.43	13.16
BGE-large	31.00	40.80	40.79	51.83	40.84
SPLADE++	23.60	33.40	31.91	42.29	32.18
ColBERTv2	16.80	24.00	22.92	31.47	23.07
Modern general and software retrieval					
ReasonIR-8B	28.00	37.60	36.60	47.97	36.72
SweRankEmbed-Large	52.40	62.40	64.07	73.14	64.50
Qwen3-Embedding-8B	52.60	64.20	63.91	74.49	63.98
Qwen3-Embed + Reranker-8B	61.40	72.40	72.45	81.41	72.95
Qwen3-Embed + Reranker-8B + MMR	61.40	72.40	72.45	81.41	72.95
Agent-aware retrieval					
AgentIR-4B	38.60	47.40	47.51	57.01	47.71
Agent-ModernColBERT	34.00	43.00	44.00	54.07	44.45
State-conditioned set acquisition					
MSS-COMPLEMENT	**73.00**	**80.60**	**82.03**	**87.89**	**82.46**

6.3 DOES CERTIFICATE COMPLETION MATTER?

Test500 scores a returned set against a certificate. Whether completing it changes what an agent does next is a separate question, and two cohorts answer it. Action52 scores a repair-localization decision against validated reference repairs; Fresh23 scores an executed repair against the repositories' own tests. Both are built by Test500's construction and screening procedure from separately frozen issue cohorts that share repositories with it (Appendix B).

Action52 holds 52 frozen states, one from each of 52 repositories so that paired resampling has independent units. Each carries a source-grounded certificate, and 32 of the references are constructed minimal residual repairs while 20 retain the validated historical patch. Each state is paired with five evidence conditions: Qwen3-Embedding-8B, that embedding reranked, MSS-COMPLEMENT, Oracle-MSS, and Oracle evidence with one required group's alternatives removed. DeepSeek-v4-flash and Claude Sonnet 5 receive identical bundles and propose local code targets without repository browsing, and source-grounded adjudication labels each target precise, coarse, or wrong, with fixed-budget precision giving unused and non-precise slots zero credit (Appendix F.4).

Adding the reranker recovers much more evidence (Table 2, left), yet neither executor's target precision separates from the embedding baseline, at $+2.92$ points for DeepSeek, $[-5.14, +11.45]$,

Table 2: *Left*, Action52 on 52 tasks: per-task evidence counts are matched within each block under a 6,144-source-token ceiling, evidence metrics are shared by both executors, and target precision uses a 171-slot denominator. *Right*, Fresh23: Tokens is the mean per-task API total, prompt plus completion. Oracle rows are diagnostics, not deployable, and strict resolution needs the fail-to-pass (F2P) tests to pass while the selected pass-to-pass neighborhood stays green; the F2P column counts tasks whose fail-to-pass tests pass. Bold marks the best deployable value (Appendices F.4 and F.5).

	Evidence		Target prec. (%)	
Condition	Complete MSS (%)	Group recall (%)	DeepSeek	Claude
Qwen3 Embed	21.15	41.60	48.54	54.39
Qwen3 + Reranker	32.69	52.44	51.46	54.39
MSS-COMPLEMENT	**51.92**	**70.74**	**64.33**	**58.48**
Oracle − 1 group	0.00	48.27	52.63	55.56
Oracle-MSS	100.00	100.00	64.91	66.67

	Resolved of 23		
Condition	Strict	F2P	Tokens
BM25	5	8	133,775
Qwen3 + Reranker	6	**10**	126,261
MSS-COMPLEMENT	**8**	**10**	**118,351**
Oracle − 1 group	7	8	98,121
Oracle-MSS	8	11	99,655

and 0.00 for Claude, $[-6.83, +6.63]$. Against this stronger baseline MSS-COMPLEMENT gains 12.87 points of precision under DeepSeek, $[+5.39, +20.36]$, and 4.09 under Claude, $[-1.89, +10.29]$. Under DeepSeek, reference-file recall improves by 4.44 points as well, $[+0.12, +10.44]$, with no loss across the 52 paired cases. Removing one required group from otherwise complete evidence, at the same evidence count, costs 12.28 and 11.11 points of precision while leaving 48.27% group recall in place: the conditions separate on certificate completion, not on how much evidence is supplied. Action52 uses its own certificates and returned-set budgets, so this tests the grouped-MSS target rather than restating a Test500 result.

Fresh23 moves the endpoint to executed tests. It is a census, not a sample: every qualifying task frozen after the main release is in it, so its size is set by what was runnable. Each task runs the whole loop, in which a retrieval method supplies evidence, an executor proposes a patch, and the repository's own tests decide the outcome, with the deployable conditions reading no organizer-private label at inference time. Two organizer-constructed diagnostics bound the target: Oracle-MSS resolves 11 of 23 tasks on the fail-to-pass (F2P) criterion, and removing one required group from that same evidence costs three of them and returns the condition to BM25's eight, and under that criterion the incomplete certificate resolves no task that the complete one misses. MSS-COMPLEMENT reaches the Oracle's strict resolution, 8 of 23, from the lowest mean API token usage of any deployable condition (Table 2, right). It resolves two more tasks than the reranked baseline under the strict criterion and matches it on fail-to-pass; every paired interval in Table 25 includes zero.

6.4 DOES IT TRANSFER? DISCOVERY, LONG-TRAJECTORY QA, AND CONTROLLER FAMILY

The full-repository diagnostic starts from frozen source using only the visible state and observed mask, over a fixed 500-state cohort that shares 434 state IDs with Test500 and carries its own certificate snapshot, in which a complete certificate is reachable for 333 states inside the shared BM25 top-1,000 pool. It gives a 5.00-point Complete-MSS@5 gain over all states, $[+1.62, +9.44]$, and 7.51 points on the coverable subset, $[+2.73, +13.88]$, from a smaller five-item source context (Table 17). The first number composes discovery with selection; the second conditions on discovery having succeeded.

AMA-Bench moves the task family: 208 real agent episodes and 2,496 questions (Zhao et al., 2026b), with the Cal500 configuration transferred without AMA-specific tuning and a single answer model and judge shared by every acquisition method. MSS-COMPLEMENT answers 1,399 correctly against 1,347 for AMA-Agent (Table 3), a difference of $+2.08$ points, episode-clustered $[-0.16, +4.37]$.

The clearest gain is a 76.2% smaller answer prompt alongside the higher accuracy point estimate. Constructing it consumes retrieval-controller tokens, so the online total is 33.57K against 13.95K, billed cumulatively across the retained source runs (Table 3). The result trades acquisition computation against answer-visible context. Appendix F.2 reports paired outcomes, deployment latency, and the full token breakdown.

Replacing the DeepSeek controller family with Claude on the 44-state prospective cohort, with the candidate input, prompts, evidence limits and scorer held fixed, gives 52.27% Complete-MSS@5

Table 3: AMA-208 under one answer model and judge, where token counts are means over all 2,496 questions. Answer prompt is the context visible to the answer model, answer total adds completion tokens, and online total adds controller tokens (Appendix F.2).

Method	Accuracy (%)	Correct	Tokens per question: Answer prompt	Answer total	Controller	Online total
BM25	39.58	988	2.56K	2.67K	0	2.67K
LongContext	45.35	1,132	5.56K	5.60K	0	5.60K
Qwen3-Embedding-8B	43.51	1,086	9.05K	9.17K	0	9.17K
Qwen3-Embed + Reranker-8B	49.12	1,226	10.09K	10.20K	0	10.20K
SSR	50.04	1,249	2.34K	2.45K	6.45K	8.90K
AMA-Agent	53.97	1,347	13.06K	13.95K	0	13.95K
MSS-COMPLEMENT	**56.05**	**1,399**	3.10K	3.95K	29.62K	33.57K

against 45.45%, so the policy carries across families. Appendices F.1–F.3 and E report that comparison, the prospective evaluation, and the parameter-neighborhood checks.

6.5 WHERE THE GAIN COMES FROM: ATTRIBUTION AND ABLATION

The independent-similarity control matches the three-call policy in controller configuration and call budget, removing the set-sufficiency objective and selected-set feedback and fixing the output to seven units (Appendix E). It reaches 66.60% Complete-MSS@5, and the full policy improves on it by 6.40 points, $[+3.42, +10.46]$ (Table 4). The separation appears where the formulation says it should: at a single returned item the two do not separate, -1.40 points, $[-3.26, +1.05]$, and the ordering reverses as soon as the prefix has room for more than one requirement.

The no-feedback variants separate the objective from output sizing. The objective row adds 3.20 points, $[-0.25, +8.26]$, adaptive sizing a further 3.00, $[+1.06, +6.50]$, and feedback a further 0.20, $[-4.00, +3.14]$. The overall contrast against independent similarity establishes the policy-level gain. Expansion and finalization add 3.00 points over the single-stage proposal, and among the configurations reaching 73.00%, the adaptive final policy delivers the smallest mean source context.

Table 4: Attribution and ablation on the same 500 Test500 states, from each variant's frozen predictions. The reference row makes no controller call, the single-stage row one; "Returned completion" scores each declared output and context is its mean returned source-token count (Appendix E).

Method	MSS objective	Selection feedback	Complete@5	Returned completion	Group@5	Context
Qwen3 + Reranker (fixed 7)	–	–	61.40	68.80	72.45	3,781
Independent similarity (fixed 7)	×	×	66.60	74.40	76.55	3,595
No feedback (fixed 7)	✓	×	69.80	78.20	80.07	3,797
No feedback (adaptive 4–8)	✓	×	72.80	81.80	82.12	3,907
Single-stage MSS proposal	✓	n/a	70.00	79.40	79.28	4,139
Fixed top-8 finalization	✓	✓	**73.00**	81.40	82.03	4,113
MSS-COMPLEMENT	✓	✓	**73.00**	80.60	82.03	3,663

7 CONCLUSION

The evidence useful to a coding agent changes as its trajectory develops. State-conditioned minimal sufficient evidence recovery makes that need explicit: the source-evidence combination still required for the current decision. SERBENCH evaluates this target through captured states, observed-evidence alignment, and grouped sufficiency certificates. MSS-COMPLEMENT recovers more complete combinations from the released pools at matched budgets, with gains extending to gold-blind discovery. Its selected evidence supports downstream answering and improves repair localization under two executors, and on executed repository tests, removing one required group from complete evidence costs resolutions. These results support constructing an agent's working context around the requirements its next decision still lacks, with recovery and downstream use evaluated separately.

Ethics Statement

External experts who annotated, adjudicated, or audited evaluation labels were paid for their work and agreed to the use of their judgments in this research. We assessed the need for ethics review under our institution's criteria: the experts judged public source code and recorded agent trajectories rather than providing information about themselves, so the annotation does not constitute human subjects research and institutional review was not required. Release sanitization is described in Appendix G.

Reproducibility Statement

The anonymous supplementary material submitted with this paper provides repository-disjoint Cal500 and Test500, the scorer, method code, prompts, configurations, and aggregate predictions; Test500 certificates stay in a separate organizer evaluator that is not part of the submitted archive. The Fresh23 end-to-end run records are included in the same archive as a separate frozen package described in Appendix G. Appendices B–H describe construction, annotation, evaluation, reproduction, and release scope. The accompanying ledger maps reported results to their configurations and source artifacts. It also includes state-level observed-overlap assignments and the script that reproduces the post-hoc subgroup summaries from frozen per-state scores.

AI Use Statement

The authors made all research decisions. Generative AI tools assisted in implementing methods, baselines, and experiment orchestration; in constructing and formatting state cards and certificates, with human expert review as described in Appendix C; in giving feedback on experimental design and in analyzing and interpreting results; in translating and polishing text drafted by the authors; and in finding and checking related work and references. Models used within the research itself, including the cross-family certificate audit, the studied methods, and the models used in evaluation, are specified in Appendices C, D.1, and E. The authors reviewed all AI-assisted work and take full responsibility for this submission.

References

Guy Alt, Eran Hirsch, Serwar Basch, Ido Dagan, and Oren Glickman. User-centric evidence ranking for attribution and fact verification. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 7215–7237. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.340.

Weibin Cai and Reza Zafarani. Less can be more: Relieving RAG bottlenecks via evidence frontloading and pressure-adaptive budgeting. *arXiv preprint arXiv:2608.25115*, 2026. doi: 10.48550/arXiv.2608.25115. URL https://arxiv.org/abs/2608.25115.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

Antoine Chaffin. Agent-ModernColBERT, 2026. URL https://huggingface.co/lightonai/Agent-ModernColBERT. Hugging Face model release.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Jimmy Lin, Akari Asai, and Victor Zhong. AgentIR: Reasoning-aware retrieval for deep research agents. *arXiv preprint arXiv:2603.04384*, 2026. doi: 10.48550/arXiv.2603.04384.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 758–759, 2009.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*,

pp. 2353–2359, 2022. doi: 10.1145/3477495.3531857. URL https://arxiv.org/abs/2205.04733.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=DT7JyQC3MR.

Zhanghao Hu, Qinglin Zhu, Runcong Zhao, Di Liang, Hanqi Yan, Yulan He, and Lin Gui. Beyond RAG for agent memory: Retrieval by decoupling and aggregation. *arXiv preprint arXiv:2602.02007*, 2026b. doi: 10.48550/arXiv.2602.02007.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020.

Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

Han Li, Letian Zhu, Bohan Zhang, Rili Feng, Jiaming Wang, Yue Pan, Earl T. Barr, Federica Sarro, Zhaoyang Chu, and He Ye. ContextBench: A benchmark for context retrieval in coding agents. *arXiv preprint arXiv:2602.05892*, 2026a. doi: 10.48550/arXiv.2602.05892.

Minghan Li, Junjie Zou, Xinxuan Lv, Chao Zhang, and Guodong Zhou. S2G-RAG: Structured sufficiency and gap judging for iterative retrieval-augmented QA. *arXiv preprint arXiv:2604.23783*, 2026b. doi: 10.48550/arXiv.2604.23783.

Yidan Lin, Kaixiang Wang, Jiong Lou, and Jie Li. Stop when memory suffices: Evidence-conditioned progressive execution for LLM agents. *arXiv preprint arXiv:2608.01285*, 2026. doi: 10.48550/arXiv.2608.01285. URL https://arxiv.org/abs/2608.01285.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Yifan Simon Liu, Ruifan Wu, Liam Gallagher, Jiazhou Liang, Armin Toroghi, and Scott Sanner. Semantic XPath: Structured agentic memory access for conversational AI. *arXiv preprint arXiv:2603.01160*, 2026. doi: 10.48550/arXiv.2603.01160.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schuetze. NoLiMa: Long-context evaluation beyond literal matching. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 44554–44570. PMLR, 2025. URL https://proceedings.mlr.press/v267/modarressi25a.html.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. doi: 10.48550/arXiv.2310.08560. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023.

Bowen Qin and Yi Xie. Agent Retrieval Bench: Evaluating repository context retrieval for coding agents. *arXiv preprint arXiv:2607.24882*, 2026. doi: 10.48550/arXiv.2607.24882.

Prithvi Rajasekaran, Ethan Dixon, Carly Ryan, and Jeremy Hadfield. Effective context engineering for AI agents. Anthropic Engineering, 2025. URL https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents.

Revanth Gangi Reddy, Tarun Suresh, JaeHyeok Doo, Ye Liu, Xuan Phi Nguyen, Yingbo Zhou, Semih Yavuz, Caiming Xiong, Heng Ji, and Shafiq Joty. SweRank: Software issue localization with code ranking. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OnkRqbNhe3.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009. doi: 10.1561/1500000019.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272/.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. ReasonIR: Training retrievers for reasoning tasks. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=kkBCNLMbGj.

Kangning Shen, Jingyuan Zhang, Chenxi Sun, Wencong Zeng, and Yang Yue. Structurally aligned subtask-level memory for software engineering agents. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=2CoRS45Ucj.

Quan Shi, Alexandra Zytek, Pedram Razavi, Karthik R. Narasimhan, and Victor Barres. τ-Knowledge: Evaluating conversational agents over unstructured knowledge. In *International Conference on Machine Learning*, 2026. doi: 10.48550/arXiv.2603.04370. URL https://openreview.net/forum?id=XHZK5abtw2.

Heydar Soudani, Elizabeth Lingg, Faegheh Hasibi, and Navid Rekabsaz. When deep research agents stagnate: Enhancing reasoning with retrieval-aware agent control. *arXiv preprint arXiv:2608.15191*, 2026. doi: 10.48550/arXiv.2608.15191.

Hao Sun, Hengyi Cai, Yuchen Li, Xuanbo Fan, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. Enhancing retrieval-augmented generation via evidence tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24116–24127. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1175. URL https://aclanthology.org/2025.acl-long.1175/.

Jingbo Sun, Wenyue Chong, Songjun Tu, Qichao Zhang, Yaocheng Zhang, Jiajun Chai, Xiaohan Wang, Wei Lin, Guojun Yin, and Dongbin Zhao. AutoSearch: Adaptive search depth for efficient agentic RAG via reinforcement learning. *arXiv preprint arXiv:2604.17337*, 2026. doi: 10.48550/arXiv.2604.17337.

Akshay Verma, Siddharth Pillai, Prateek Sircar, and Deepak Gupta. NEST: Nested evidence survival for retrieval. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 507–516, 2026. doi: 10.18653/v1/2026.acl-industry.35.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.10813.

Chunqiu Steven Xia, Yinlin Deng, Soren Dunn, and Lingming Zhang. Agentless: Demystifying LLM-based software engineering agents. *arXiv preprint arXiv:2407.01489*, 2024. URL https://arxiv.org/abs/2407.01489.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2023. doi: 10.48550/arXiv.2309.07597. URL https://arxiv.org/abs/2309.07597.

Caijun Yan, Yang Zhou, Meixing Shi, Haoran Sun, Yichen Li, Yuxiang Cai, and Yankai Jiang. REVEAL: A rubric-guided agent for explicit evidence sufficiency verification in long-video question answering. *arXiv preprint arXiv:2608.08612*, 2026. doi: 10.48550/arXiv.2608.08612.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R. Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Qijie You, Wenkai Yu, Hao Liang, Zhen Hao Wong, and Wentao Zhang. AgenticRAGTracer: A hop-aware benchmark for diagnosing multi-step retrieval reasoning in agentic RAG. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 1301–1335, 2026. doi: 10.18653/v1/2026.findings-acl.66.

Daoguang Zan, Zhirong Huang, Wei Liu, Hanwu Chen, Shulin Xin, Linhao Zhang, Qi Liu, Aoyan Li, Lu Chen, Xiaojian Zhong, Siyao Liu, Yongsheng Xiao, Liangqiang Chen, Yuyu Zhang, Jing Su, Tianyu Liu, Rui Long, Ming Ding, and Liang Xiang. Multi-SWE-bench: A multilingual benchmark for issue resolving. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2111.

Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. RepoCoder: Repository-level code completion through iterative retrieval and generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2471–2484, 2023. URL https://aclanthology.org/2023.emnlp-main.151/.

Fuwei Zhang, Yanzhao Zhang, Mingxin Li, Dingkun Long, Lexiang Hu, Pengjun Xie, Zhao Zhang, and Fuzhen Zhuang. CORE-Bench: A comprehensive benchmark for code retrieval in the era of agentic coding. *arXiv preprint arXiv:2606.11864*, 2026a. doi: 10.48550/arXiv.2606.11864.

Linghao Zhang, Shilin He, Chaoyun Zhang, Yu Kang, Bowen Li, Chengxing Xie, Junhao Wang, Maoquan Wang, Yufan Huang, Shengyu Fu, Elsie Nallipogu, Qingwei Lin, Yingnong Dang, Saravan Rajmohan, and Dongmei Zhang. SWE-bench goes live! In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-4923.

Shaoqiu Zhang, Yuhang Wang, Jialiang Liang, Yuling Shi, Wenhao Zeng, Maoquan Wang, Shilin He, Ningyuan Xu, Siyu Ye, Kai Cai, and Xiaodong Gu. SWE-Explore: Benchmarking how coding agents explore repositories. *arXiv preprint arXiv:2606.07297*, 2026b. doi: 10.48550/arXiv.2606.07297.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b. doi: 10.48550/arXiv.2506.05176.

Beidi Zhao, Yaoqi Chen, Yuru Feng, Menghao Li, Qianxi Zhang, Baotong Lu, Jianan Lu, Zhirui Wang, Xinjiang Wang, Shusen Xu, Zengzhong Li, Xiaoxiao Li, and Qi Chen. MESA: Task-adaptive multi-structure evidence selection for long-horizon agent memory. *arXiv preprint arXiv:2608.10108*, 2026a. doi: 10.48550/arXiv.2608.10108. URL https://arxiv.org/abs/2608.10108.

Yujie Zhao, Boqin Yuan, Junbo Huang, Haocheng Yuan, Zhongming Yu, Haozhou Xu, Lanxiang Hu, Abhilash Shankarampeta, Zimeng Huang, Wentao Ni, Yuandong Tian, and Jishen Zhao. AMA-Bench: Evaluating long-horizon memory for agentic applications. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b. URL https://arxiv.org/abs/2602.22769.

Appendices A and B position SERBENCH and document how it was built; Appendix C covers certificate annotation and audit. Appendix D gives the evaluation protocol, the baselines, and the extended Test500 analyses; Appendix E specifies MSS-COMPLEMENT and its attribution. Appendix F.1 onward reports the cohorts outside the released pools, and Appendices G and H the artifact and its scope.

A BENCHMARK POSITIONING AND SPECIFICATION

SERBench lies at the intersection of repository-context evaluation, process-grounded retrieval, and agent-memory diagnosis. ContextBench, Agent Retrieval Bench, CORE-Bench, and SWE-Explore examine context retrieval, file retrieval, and repository exploration from issue or workflow signals (Li et al., 2026a; Qin & Xie, 2026; Zhang et al., 2026a;b). Table 5 compares the properties that define its evaluation target: an exact intermediate decision as the query, already observed evidence recorded as IDs in the released candidate pool, a grouped sufficiency target, private evaluation certificates, and a paired test of evidence use.

Table 5: Positioning against closely related evaluation settings. The State column records the trajectory granularity used as the retrieval condition. The Observed IDs column marks whether each decision state stores the evidence its trajectory has already observed as IDs in the released candidate pool; SERBENCH additionally audits that overlap. ContextBench instead traces the code an evaluated agent inspects, and Agent Retrieval Bench records query-supplied files as given context. Grouped MSS requires alternatives within a requirement and conjunction across requirements.

Setting	Code	State	Observed IDs	Grouped MSS	Private	Action test
ContextBench (Li et al., 2026a)	✓	process	–	–	–	analysis
Agent Retrieval Bench (Qin & Xie, 2026)	✓	workflow	–	–	–	pilot
MemoryAgentBench (Hu et al., 2026a)	–	interaction	–	–	–	–
AgenticRAGTracer (You et al., 2026)	–	hop trace	–	–	–	diagnostic
SERBench	✓	exact	✓	✓	✓	paired

SERBench builds on established retrieval primitives (BM25, dense dual encoders, late interaction, reciprocal-rank fusion, and maximal marginal relevance) and on context engineering practice (Robertson & Zaragoza, 2009; Karpukhin et al., 2020; Khattab & Zaharia, 2020; Cormack et al., 2009; Carbonell & Goldstein, 1998; Rajasekaran et al., 2025). Long-context evaluations document position, scale, and lexical-overlap limits (Liu et al., 2024; Hsieh et al., 2024; Modarressi et al., 2025); SERBench instantiates these concerns at a captured software-agent decision and scores a compact source-grounded set against its unresolved requirements.

B SERBENCH CONSTRUCTION AND GOVERNANCE

B.1 STATE SELECTION AND PUBLIC SCHEMA

The builder begins from trajectories that expose an issue identifier, repository and base commit, typed tool actions, observations, opened files, search queries and results, test output when present, and a state update. Candidate boundaries are extracted before search, after search, after file inspection, and before an edit. A state enters Test500 only when the repository snapshot is reconstructable, the prefix contains genuine task progression, the information need belongs to the selected decision, observed source spans can be aligned, the content fingerprint is unique, near-duplicate prompts are removed, and the source instance belongs to the frozen Test500 inventory.

State fields are normalized without rewriting source text to simplify annotation. The public interface contains the issue, current information need, current observation, opened files, search queries, latest test output, current diff, hypothesis, subgoal, and visible trajectory prefix. Framework-specific fields can be retained in a namespaced metadata object. A stable state ID and content fingerprint allow another framework to emit an equivalent state card without reproducing the original private runtime.

Trajectory generation. Trajectories are produced by `adaptive_llm_trace_v3_agent`, an instrumented harness written for this benchmark so that every tool action, observed source span, and state update is recorded. For each issue the repository is materialized at its base commit and indexed into at most 1,600 provenance-preserving chunks served by path-aware BM25. The agent drafts a plan before it sees any repository content, issues two searches returning eight chunks each, reads up to three previously unopened files in windows of roughly 120 lines, and regenerates its retrieval intent, information need, hypothesis, and subgoal after every observation under the frozen `observation_refresh_v3` policy. One trajectory is run per issue and stops at the intended edit, which isolates the evidence need of a decision before any repair is committed; where a trajectory contains several searches or file reads, the last state matching each boundary type is kept. Test500 trajectories use DeepSeek-v4-Flash at temperature 0.1 with thinking disabled. Cal500 uses the same harness and refresh policy across a mix of planners: 399 states from a locally served Qwen3-Coder-30B, 99 from DeepSeek-v4-Flash, and two from a deterministic offline planner. Released state cards omit the planner identity, so no evaluated method can condition on it.

Source issues. The Test500 issue pool is frozen before any trajectory is generated: 362 candidate issues over 70 repositories, 320 of them from Multi-SWE-bench (Zan et al., 2025) and 42 from SWE-bench-Live (Zhang et al., 2025a), each bound to a real repository and a fixed base commit, with a reserve held for replacements. Trajectory completeness, state quality, candidate-pool scorability, and certificate review reduce this to the released 500 states over 242 issue instances and 45 repositories. The pool is frozen without reading any method result, and it shares no repository with Cal500.

Cohort overlap. Action52 and Fresh23 follow the same harness, screening and certificate protocol as Test500 but are frozen as separate issue cohorts rather than drawn from the Test500 pool above, so the three evaluation cohorts are not repository-disjoint from one another. Candidate materialization and observed-evidence handling are cohort-specific and are documented with each evaluation. Action52 and Fresh23 share eight repositories, Action52 and Test500 five, and Fresh23 and Test500 six, with `lektor/lektor` the only repository common to all three. At the instance level the overlaps are eight, three and three, and `lektor__lektor-1224` is the only instance shared by all three. The cohorts score different objects against different private certificates: a returned set against a sufficiency certificate, a repair-localization decision, and an executed test outcome.

B.2 Calibration and Test Composition

Table 6: Final SERBench composition.

Collection	Repositories	Issues	States
Cal500	174	241	500
Test500	45	242	500

State type	States	Graded
Before search	200	200
After search	124	124
After file inspection	104	104
Before edit	72	72
Total	500	500

Table 6 gives the final composition. SERBench has two fixed partitions. Cal500 publishes 500 states and their certificates for development and calibration. Test500 publishes 500 test queries, while its certificates remain in the organizer evaluator and alone determine leaderboard results. The partitions are disjoint by state, issue instance, and repository. Reserve and excluded review records remain outside both released partitions.

Test500 certificates contain 1, 2, 3, 4, 5, and 6 required groups for 216, 135, 121, 22, 5, and 1 states, respectively, for a mean of 1.936 requirements per state. A majority of states, 284 of 500, require at least two independently grounded facts. The single-group stratum retains decisions whose unresolved need is atomic.

B.3 Evidence Segmentation and Observed Alignment

Repository snapshots are segmented into bounded evidence units. A unit stores a stable evidence and corpus ID, repository-relative path, line or structural span, symbol and evidence-type metadata

when available, intact L2 text, compact L1 content, an L0 navigation card, token length, hashes, and reconstruction provenance. Units preserve one behavior or constraint while remaining independently retrievable. Overlapping chunks resolve to one canonical unit before scoring. When several source units independently support the same requirement, the certificate records them as alternatives instead of rewarding duplicates.

Observed intervals come from file reads and search results in the trajectory prefix. Exact path and overlapping line ranges are matched first. Normalized-content alignment handles tool output without precise spans. The aligned evidence IDs record O_t. Test500 evaluates the frozen candidate pools as released, including observed units where present, and prefix scores retain their original positions. Appendix D.4 reports performance by exact candidate/observation ID overlap. In a separate 21-state pilot that ranked a full repository index before any filtering, 53.3% of a state-conditioned query's top ten repeated visible evidence, compared with 18.1% for issue retrieval.

B.4 Candidate Discovery and Task Boundary

SERBench separates candidate discovery from evidence-set recovery. The public pool for each state is formed from a frozen union of gold-blind full-repository retrieval routes and source units discovered through the future trajectory, gold patch, and test patch. Units are canonicalized and construction-source tags are stripped before release. All reported methods use the same frozen candidate pool for each state; the pools retain recorded observed units in 186 states, enabling a direct robustness diagnostic. The organizer certificate is then grounded in this released pool, but certificate fields and source-origin labels are unavailable to retrieval methods.

The primary task is candidate-pool evidence-set recovery: once candidate discovery supplies a recall-complete search region, the method must identify a set covering the decision's residual requirements. In the separate gold-blind full-repository diagnostic, BM25 top-1000 covers a complete certificate for 66.6% of states and 75.96% of required groups. Qwen3 reranking raises complete-certificate coverage at depth 100 from 36.8% to 61.8%, while sharing the same 66.6% top-1000 coverage ceiling. Within the released pools, an oracle reaches 99.80% Complete-MSS@5. The remaining state requires six distinct units and has no admissible branch of at most five units. Uniform selection of five distinct units from each released candidate pool has an exact expected Complete-MSS@5 of 5.17%. These values use the current Test500 certificates, group thresholds, and complete alternative branches.

C Grouped-MSS Annotation and Governance

C.1 Annotation Object

Each private certificate conforms to `serbench-mss-v1`. Required fields identify the state, retrieval-needed status, evidence groups, release scope, and confidence. Every evidence group stores a semantic role, acceptable evidence IDs, a minimum required count, a necessity weight, a rationale, and source quotations. Optional fields record observed evidence, source-first repair history, and enumerated complete alternatives.

The schema represents two different forms of plurality. Interchangeable source units for one fact are recorded inside the same group, with the group's declared threshold q_j determining how many acceptable IDs are needed. When $q_j = 1$, the group is an OR over those IDs; larger thresholds require the stated count. Distinct complete solutions are stored in `alternative_minimal_sets`. Each entry is an AND branch, and satisfying any complete branch is sufficient. The scorer evaluates this nested structure directly rather than flattening the union of branch members, which would accept a mixture drawn from different solutions. In Test500 every enumerated branch is derived from the grouped requirements and satisfies them.

Observed-evidence annotation follows the provenance of a claim rather than lexical overlap with the state card. File contents and search results captured by the trajectory, together with explicit `observed_evidence_ids`, count as acquired evidence. Agent-authored plans and hypotheses remain part of the query but require source confirmation. Reviewers first establish this epistemic boundary, then identify the unresolved requirements, ground them in source units, and test whether every retained requirement contributes nonredundant value to a concrete next search, edit, or verification action.

C.2 Independent Review and Source-First Resolution

Gold production and validation use two sequential safeguards. First, a cross-family model calibration audit has Grok 4.6 and Gemini 3.6 Flash independently inspect all 500 certificates without method predictions or human opinions; Claude Sonnet 5 provides source-constrained arbitration for 455 selected rows. All three reviewers are accessed through OpenRouter. These model outputs are review signals rather than automatic label changes, and Table 7 records what each reviewer recommended. Second, a blinded human closure has two expert annotators independently review all 500 states, followed by a senior expert who arbitrates substantive anonymized A/B disagreements without receiving model decisions. Source-first human decisions are then applied to the release: every adjudicated certificate repair is incorporated and every inserted replacement is human-reviewed. This closure produces the final 500-state Test500 release and its aggregate governance ledger. Certificate changes require support from the visible state, repository source, quotations, and public candidate pool.

Table 7: Cross-family model audit recommendations before source-first adjudication.

Reviewer	Rows	Accept	Revise	Rebuild	Unscorable	Uncertain
Grok 4.6	500	23	167	285	25	–
Gemini 3.6 Flash	500	117	363	5	15	–
Claude Sonnet 5	455	93	54	1	53	254

The different marginal distributions reveal distinct review thresholds, especially for whether a planning-state need is sufficiently grounded and whether a certificate is minimal. Source inspection separates genuine certificate plurality from four other causes of disagreement: the observed/unobserved boundary, quotation or provenance fidelity, joint sufficiency and minimality, and the action boundary represented by the state. Alternative branches resolve genuine plurality. Grounding and state defects instead require repair, reconstruction, or replacement.

Both independent human reviewers completed every assigned state. The four retrospective sensitivity classes are S0 (no final change), S1 (a certificate edit with no Complete-MSS change for any locked method at $k \in \{5, 8, 16\}$), S2 (a change to at least one such outcome), and S3 (replacement of a state whose decision boundary is not independently scorable). A 200-state retrospective audit applies these classes. It is deliberately stratified rather than drawn at random, so its class counts describe that audited sample and are not release-wide rates, and it gives 55/26/82/37 cases for S0/S1/S2/S3; a single flip at any locked cutoff assigns S2. Method outcomes were unavailable during review.

C.3 Final Certificate Accounting

Table 8 summarizes the released certificates. Across the complete human-audit closure, the ledger records 232 certificate-repair actions and 93 replacement decisions or mappings. These are governance events rather than mutually exclusive state counts: intermediate replacement queues and raw review flags are not final counts, while the final release contains 500 unique scorable states.

Table 8: Final source-reviewed Test500 certificates. Update counts are governance events and do not partition the 500-state denominator.

Property	Count
Scorable states	500
Independent model reviews	1,000
Third-model arbitration rows	455
Final human-audited states	500
Certificate-repair actions (cumulative)	232
Replacement decisions/mappings (cumulative)	93
States with enumerated complete branches	454
States with multiple complete branches	124
Enumerated complete branches	955

C.4 Independent Validity Audit of the Frozen Release

This audit sampled 80 states uniformly at random from the 500-state release and assigned them to two experts outside certificate construction and arbitration. They reviewed independently, without access to each other's verdicts, and completed all 155 certificate groups in the sample. Each state received an accept, revise, or rebuild verdict; each group received eight field-level judgments covering requirement validity, source grounding, joint sufficiency, necessity, the minimum-required threshold, acceptable-evidence completeness, observed-evidence contamination, and alternative-branch structure.

Table 9: Independent validity audit of 80 states drawn uniformly at random from the Test500 release, by two experts outside the construction and arbitration chain.

Property	Value
States audited	80 of 500
Certificate groups audited	155
Expert A verdicts (accept / revise / rebuild)	78 / 1 / 1
Expert B verdicts (accept / revise / rebuild)	79 / 1 / 0
States accepted by both	77 (96.25%)
States flagged by at least one	3 (3.75%)
States flagged by both	0 (0.00%)
Group-level agreement on all eight fields	151 of 155 (97.42%)
Per-field agreement range	97.42–100%
Raw agreement (state / group)	96.25% / 97.42%
$2p_o - 1$ agreement (state / group)	0.925 / 0.948

Table 9 reports the audit in full. The auditors jointly accepted 77 of 80 states and gave identical judgments on all eight fields for 151 of 155 groups. No state was flagged by both, and both recorded confidence four or five on every state. Cohen's κ is uninformative at these margins: with the two auditors accepting 97.5% and 98.75% of states, expected agreement nearly equals observed agreement and κ falls toward zero despite agreement above 96%. We therefore report raw agreement alongside the prevalence-adjusted, bias-adjusted statistic $2p_o - 1$.

All 155 groups received identical judgments on requirement validity, the minimum-required threshold, observed-evidence contamination, and alternative-branch structure. The remaining disagreements were on source grounding, joint sufficiency, and acceptable-evidence completeness, with every group-level disagreement belonging to one state. This judgment concerns the certificate's acceptable evidence, not whether the public candidate pool contains any observed units.

These are reported as agreement statistics for the audit, not as a further revision pass: the released labels were unchanged.

After the freeze, deterministic checks validate 500 state/label joins, evidence IDs, recorded observed-evidence IDs, source quotations, branch membership, and public-pool provenance. A separate offline audit confirms that across all 500 frozen certificates no recorded observed ID appears in an acceptable set or in a complete alternative branch, and checks candidate/observed intersections without changing the frozen inputs or predictions. Initial reviews were blinded to method outcomes; targeted checks and replacement provenance remain in the audit record. Predictions are rescored against the release, with new retrieval outputs required only when public inputs change. All adjudicated repairs were applied, inserted replacements were human-reviewed, and the release passed structural, provenance, and scorer checks. The archive retains blinded reviews, source-grounded decisions, replacements, per-state provenance, and the retrospective severity analysis.

D Metrics and Baseline Implementations

For state t, let $(e_1, \ldots, e_k)$ denote the ranked prediction. Group g_j is covered when at least its declared threshold $q_j = \texttt{minimum_required}$ acceptable IDs appear. Thus $q_j = 1$ is an OR over acceptable evidence IDs; a group with $q_j > 1$ requires the stated count. Complete-MSS accepts the conjunction of all required groups or one fully contained enumerated alternative branch. Each alternative branch is itself a complete AND solution, and the scorer never flattens distinct branches into a union. Group recall averages group coverage within each state before macro-averaging across

states. Necessity-weighted recall replaces the uniform group average with normalized annotation weights. Grouped nDCG adds gain only when a newly covered group first appears.

The public scorer accepts JSONL rows containing a state ID, method ID, and ordered evidence IDs. The scorer joins each frozen ranking to its state certificate inside the organizer evaluator and evaluates the original first k positions. An observed ID is not removed or replaced by a later item. The release audit verifies all 500 state/certificate joins, nonempty certificates, valid public evidence IDs, and separation between predictions and private labels.

D.1 Common Evaluation Protocol

Every method is evaluated on the same public state card and frozen candidate pool; Test500 certificates remain in the organizer evaluator until predictions are complete. The primary MSS-Complement configuration is selected on Cal500 and applied to Test500, the gold-blind full-repository diagnostic, AMA-208, Prospective44, and Action52 without retuning. Ablations and controller-family comparisons vary only their declared factors.

SERBench intervals use 20,000 paired repository-level bootstrap resamples clustered by repository. AMA intervals use 20,000 episode-level resamples with seed 20260901, and Action52 intervals use 10,000 task-level resamples with seed 20260904. The observed-overlap diagnostic uses seed 20260915 and 20,000 paired repository-cluster resamples within each stratum, weighting states equally; the reported percentile intervals are descriptive. Semantic calls use temperature zero and structured outputs. DeepSeek calls use DeepSeek's official API; Claude, Gemini, and Grok calls used for executor comparison or adjudication use OpenRouter; open-weight models, including retrievers and the Qwen3.6-35B-A3B judge, run locally.

D.2 Complete Test500 Baseline Matrix

All retrievers rank the complete released candidate pool. General retrievers choose between issue-only and structured state-plus-need queries on a fixed 75-state subset of Cal500. Agent-aware retrievers choose between their supported reasoning or information-need representations on the same subset. These choices are fixed before Test500 evaluation.

BM25 (Robertson & Zaragoza, 2009) uses the standard probabilistic score over the frozen query and source units. BGE-large (Xiao et al., 2023) ranks by dense cosine similarity. SPLADE++ (Formal et al., 2022) uses learned sparse vectors, and ColBERTv2 (Santhanam et al., 2022) uses late interaction. ReasonIR-8B, SweRankEmbed-Large, and Qwen3-Embedding-8B use their released similarity functions (Shao et al., 2025; Reddy et al., 2026; Zhang et al., 2025b). The Qwen3 reranking row retrieves 100 units with Qwen3-Embedding-8B and reranks them with Qwen3-Reranker-8B. AgentIR-4B (Chen et al., 2026) receives the visible reasoning state and current need. Agent-ModernColBERT (Chaffin, 2026; Chen et al., 2026) uses its agent-aware late-interaction representation. The reported Test500 rankings are evaluated without an observed-evidence filter.

Released checkpoints. Table 10 gives the exact model repository behind each evaluated baseline.

Table 10: Evaluated retrieval checkpoints. Each is pinned to a fixed revision in the released configuration contracts.

Baseline	Checkpoint
BGE-large	`BAAI/bge-large-en-v1.5`
SPLADE++	`prithivida/Splade_PP_en_v1`, `Qdrant` ONNX conversion
ColBERTv2	`colbert-ir/colbertv2.0`
ReasonIR-8B	`reasonir/ReasonIR-8B`
SweRankEmbed-Large	`Salesforce/SweRankEmbed-Large`
Qwen3-Embedding-8B	`Qwen/Qwen3-Embedding-8B`
Qwen3-Reranker-8B	`Qwen/Qwen3-Reranker-8B`
AgentIR-4B	`Tevatron/AgentIR-4B`
Agent-ModernColBERT	`lightonai/Agent-ModernColBERT`

The MMR row starts from the Qwen3 reranker and freezes its relevance–diversity coefficient on calibration data. The selected coefficient is 1.0, recovering the original ranking. The independent-similarity control of Table 4 begins from the same nine-view fused ranking as MSS-COMPLEMENT. Its per-call geometry is as follows. The Pro call independently selects up to eight units from the first 80. A Flash call independently selects up to 48 from the first 120 without seeing or excluding that first selection. A final Flash call reranks their union against the frozen fused order and returns exactly seven units.

Displayed differences are calculated from unrounded values and then rounded, so subtracting rounded table entries can differ in the last decimal place.

State-set reranker (SSR). SSR is our one-pass semantic set reranker. It reads the visible state and a retrieved candidate union, then selects five evidence IDs with structured-output validation and deduplication. On Prospective44, the frozen report uses DeepSeek-v4-flash, one call, the leading ten candidates from each of two BGE-refined lexical routes, 1,600-character evidence excerpts, and a 2,400-token response limit. These limits describe controller input excerpts and response generation, not a universal returned-source-token budget. The AMA adapter instead uses trajectory-memory candidates, a 32-card semantic shortlist drawn from a 160-candidate route pool, one DeepSeek-v4-flash call, and a 4,096-token output-context ceiling. Cohort manifests retain these adapter settings and the corresponding predictions.

D.3 COMPLETE BUDGET CURVES

Table 11: Complete-MSS across prefix caps on frozen Test500 predictions. A row remains constant after its available output ends.

Method	$k = 1$	$k = 3$	$k = 5$	$k = 8$	$k = 10$	$k = 12$	$k = 16$
Qwen3 + Reranker	27.8	48.4	61.4	72.4	77.0	80.0	85.8
Independent similarity (fixed 7)	28.4	53.4	66.6	74.4	74.4	74.4	74.4
Single-stage MSS proposal	24.8	53.4	70.0	79.4	79.4	79.4	79.4
MSS-COMPLEMENT	27.0	58.8	**73.0**	**80.6**	80.6	80.6	80.6
Fixed top-8 finalization	27.0	58.8	73.0	81.4	81.4	81.4	81.4

Tables 11 and 12 report the full prefix curves and matched budgets. At sixteen items, Qwen reaches 85.80% Complete-MSS with 8,643 mean source tokens. The matched item and token budgets isolate recovery within a common delivery constraint.

Table 12: Fair source-budget comparisons against Qwen3 embedding plus reranking. Values are Complete-MSS percentages; context is mean source tokens actually admitted, which falls below the returned mean in token-capped rows because individual states can exceed the cap.

Condition	Ours	Qwen	Δ	Ours/Qwen context
5 items	73.00	61.40	+11.60	2,543 / 2,721
8 items	80.60	72.40	+8.20	3,663 / 4,323
4,096 tokens	80.00	71.20	+8.80	3,307 / 4,056
6,144 tokens	80.60	80.00	+0.60	3,655 / 6,100
Per-state matched	80.60	69.20	+11.40	3,663 / 3,628

D.4 PERFORMANCE BY OBSERVED-CANDIDATE OVERLAP

Released candidate pools retain source units the agent had already observed in some states and not in others. This appendix partitions Test500 by that property and reports the main retrieval comparison within each stratum. The diagnostic was conducted after the primary analysis and aggregates the same frozen predictions without reranking, rescoring, or retuning.

A state belongs to the overlap stratum when its released candidate pool contains at least one evidence ID recorded in its observed set. The partition is read from the released per-candidate observation flag and independently reproduced by an exact evidence-ID join; the two agree on all 500 states and on

all 953 flagged candidate entries. The strata are mutually exclusive and exhaust the 500 states: 314 without recorded overlap and 186 with. They cover 45 and 31 repositories, which do not sum to the cohort total because a repository can contribute states to both. Overlap here means recorded exact evidence IDs, not semantic or line-range disjointness.

Table 13: State composition of the two strata by decision boundary.

Stratum	Before search	After search	After file inspection	Before edit	Total
No recorded overlap	200	6	63	45	314
Recorded overlap	0	118	41	27	186

The strata differ systematically in trajectory stage: every state with recorded overlap follows a search, while most states without it precede one.

Table 14: Test500 main retrieval methods within each overlap stratum. Columns match Table 1; best value per column in bold.

Method	Complete@5	Complete@8	Group@5	Group@8	Necessity@5
No recorded overlap, $n = 314$					
BM25	10.51	15.92	15.50	22.49	15.61
BGE-large	35.35	44.90	46.99	57.31	47.13
SPLADE++	29.62	40.76	40.38	51.34	40.86
ColBERTv2	18.15	26.11	25.16	34.34	25.32
ReasonIR-8B	34.08	43.95	44.08	55.94	44.21
SweRankEmbed-Large	55.10	64.97	68.82	77.37	69.33
Qwen3-Embedding-8B	54.46	66.24	67.46	77.69	67.52
Qwen3-Embed + Reranker-8B	65.92	77.39	79.09	86.94	79.83
Qwen3-Embed + Reranker-8B + MMR	65.92	77.39	79.09	86.94	79.83
AgentIR-4B	31.21	41.40	41.07	51.67	41.18
Agent-ModernColBERT	33.76	43.31	45.12	55.47	45.68
MSS-COMPLEMENT	**74.20**	**81.85**	**83.99**	**89.74**	**84.48**
Recorded overlap, $n = 186$					
BM25	6.45	8.06	9.09	11.56	9.02
BGE-large	23.66	33.87	30.33	42.56	30.21
SPLADE++	13.44	20.97	17.61	27.02	17.53
ColBERTv2	14.52	20.43	19.13	26.61	19.27
ReasonIR-8B	17.74	26.88	23.97	34.50	24.07
SweRankEmbed-Large	47.85	58.06	56.05	65.99	56.34
Qwen3-Embedding-8B	49.46	60.75	57.93	69.09	58.02
Qwen3-Embed + Reranker-8B	53.76	63.98	61.25	72.09	61.34
Qwen3-Embed + Reranker-8B + MMR	53.76	63.98	61.25	72.09	61.34
AgentIR-4B	51.08	57.53	58.38	66.04	58.73
Agent-ModernColBERT	34.41	42.47	42.11	51.70	42.37
MSS-COMPLEMENT	**70.97**	**78.49**	**78.72**	**84.77**	**79.06**

MSS-COMPLEMENT ranks first on all five metrics in both strata, and Qwen3 embedding with reranking is the strongest baseline in both.

Table 15: Paired difference between MSS-COMPLEMENT and Qwen3 embedding with reranking within each stratum.

Stratum	Complete@5 difference (pp)	Complete@8 difference (pp)
No recorded overlap, $n = 314$	$+8.28$ $[+3.97, +13.31]$	$+4.46$ $[+1.84, +9.14]$
Recorded overlap, $n = 186$	$+17.20$ $[+8.02, +27.75]$	$+14.52$ $[+8.26, +23.76]$

Relative to Qwen3 embedding with reranking, the Complete-MSS gains at five and eight items have paired 95% confidence intervals entirely above zero in both strata. The main retrieval advantage is therefore present in states both with and without recorded observed-ID overlap in their candidate pools.

Intervals use 20,000 paired resamples clustered by repository within each stratum, with seed 20260915; states are weighted equally and both methods use the same repository multiplicities in each resample. Because the strata differ systematically in trajectory stage, the difference between their effect sizes is descriptive rather than an estimate of the causal effect of observed candidates. Weighting the two strata by state count reproduces the full-cohort means in Table 1 exactly.

D.5 Results by Required Evidence-Group Count

Table 16 separates the frozen Test500 states into mutually exclusive strata with one, two, and at least three required evidence groups. The strata count decision-specific requirement groups rather than distinct evidence items, since one source unit may satisfy several groups. Combining the two multi-group strata gives 284 states: the Complete-MSS gain over Qwen3 embedding plus reranking is 14.79 points at five items, $[+9.42, +22.61]$, and 10.56 points at eight, $[+6.76, +18.12]$. Both repository-clustered intervals exclude zero.

Table 16: Test500 by the exact number of required evidence groups in the frozen certificate. MSS-Complement against Qwen3-Embedding-8B plus Qwen3-Reranker-8B, with 20,000 paired repository-clustered bootstrap draws per row. Bold marks a difference whose 95% interval excludes zero. On single-group states with a unit threshold, Complete-MSS reduces to hit@k.

Required		Complete-MSS@5			Complete-MSS@8		
groups	States	Qwen	MSS-Complement	Δ	Qwen	MSS-Complement	Δ
1	216	78.70	86.11	+7.41	85.19	90.28	+5.09
2	135	61.48	77.04	**+15.56**	75.56	85.93	**+10.37**
≥ 3	149	36.24	50.34	**+14.09**	51.01	61.74	**+10.74**
All	500	61.40	73.00	**+11.60**	72.40	80.60	**+8.20**

At five items the gains are 7.41 points for single-group states, $[-0.55, +14.39]$, 15.56 for two-group states, $[+7.14, +31.94]$, and 14.09 for states requiring at least three groups, $[+6.25, +21.67]$, where completion rises from 36.24% to 50.34%. At eight items, the gains are 5.09 points for single-group states, $[-0.81, +11.69]$, 10.37 for two-group states, $[+4.49, +23.00]$, and 10.74 for states requiring at least three groups, $[+5.60, +16.84]$. The two multi-group strata retain positive intervals at this larger prefix.

D.6 Gold-Blind Full-Repository Diagnostic

Table 17: Gold-blind full-repository diagnostic on the fixed discovery cohort. Coverable denotes states whose complete certificate occurs in the BM25 top 1,000. Values are percentages, and the last column is mean source tokens in the five-item prefix.

	All 500				Coverable 333				
	Complete		Group recall		Complete		Group recall		Source
Method	@5	@8	@5	@8	@5	@8	@5	@8	tok.@5
Qwen3 + Reranker	33.80	40.80	50.69	55.69	50.75	61.26	67.88	74.62	4,290
MSS-Complement	**38.80**	**43.20**	**54.24**	**57.87**	**58.26**	**64.86**	**73.31**	**77.77**	**4,014**

The diagnostic retrieves from each frozen repository using only the visible state and observed mask. BM25 supplies the top-1,000 discovery pool. Qwen3-Embedding-8B ranks within that pool, and Qwen3-Reranker-8B reranks its leading 100 candidates. Private certificates enter only after all rankings and controller outputs are frozen. A complete certificate is present for 333 states in the fixed discovery cohort. This cohort carries its own certificate snapshot and shares 434 state IDs with the current Test500, so its scores are reported against that cohort throughout. The all-state score composes discovery with selection, and the coverable score conditions on successful discovery. The full row-by-row accounting is in Table 17.

D.7 Public Calibration Example

Consider a Cal500 state from `arrow-py/arrow` immediately before an edit. The agent has already inspected 53 source units and must decide how to replace `dateutil` timezone behavior while preserving every call site. The public certificate has two required groups: the parsing behavior in `arrow/parser.py` and the locations that construct or consume timezone objects. Qwen3 embed-plus-reranker fills its top five with call-site evidence and covers only the second group. MSS-Complement includes the parser implementation together with representative call sites, completing both groups under the same five-item limit.

E MSS-Complement Operational Details

E.1 Frozen Candidate and Controller Interface

This interface instantiates the proposal, expansion, and finalization calls in Figure 3. Source cards are the controller's views of candidate units, while the final selected units determine the downstream working context.

On the supplied candidate pool, nine label-hidden views rank intact source units: BM25 over the question, the full state, prior actions, and observations, word and character TF–IDF, dense similarity, entity overlap, and recency. Reciprocal-rank fusion admits the leading 64 units from each view and caps the union at 384. The first semantic call sees 80 source cards capped at 760 characters. The second and third calls each see at most 120 cards, capped at 280 and 620 characters. The second-stage expansion region contains 48 units. These are controller-card character budgets; the final working context is separately constrained to 6,144 source tokens.

The first call uses `deepseek-v4-pro` and the two revision calls use `deepseek-v4-flash`. Both aliases are called through DeepSeek's official API at temperature zero, with structured JSON outputs, a 240-second timeout, and at most three attempts, and were accessed during August–September 2026. Because the provider exposed no immutable revision identifiers, the artifact records aliases, access dates, prompts, and response hashes. The interface requires one state-interpretation controller followed by two selected-set revision calls.

Stage one proposes eight evidence IDs that jointly support a precise answer to the current information need. Stage two receives that proposal and deeper candidates, then identifies evidence for answer facts not yet supported. Stage three checks the combined set, revises the set to reduce repeated support, and returns 4–8 ranked source units. Unknown and duplicate IDs are removed. Deterministic validation enforces membership in the public candidate set and a 6,144 source-token ceiling. A deterministic ranking fallback is available if structured output fails; no reported final run used it.

The prompts never include a grouped-MSS certificate, group count, gold evidence ID, answer, or reward. The second and third prompts do see the currently selected set. This selected-set feedback conditions later choices on support already retained, while the candidate pool, scorer, and private labels remain fixed outside the calls. The selected set here is the current proposal, not the trajectory's observed set O_t; the Test500 adapter applies no additional filter based on recorded observed IDs.

E.2 Attribution and Ablation Accounting

The three-call variants in Table 4 use the same candidate generator, controller allocation, stage caps, source ceiling, decoding settings, and direct provider endpoint. The independent-similarity row removes the MSS objective and selected-set feedback. Of 1,500 controller responses, 1,498 have the state-specific requested length and 1,488 require no normalization; 12 responses are deterministically normalized against the frozen candidate relevance order, and 19 shorter second-stage lists are valid because those states have fewer than 48 candidates. Every final trace satisfies the stage cardinalities $\langle 8, \min(48, N), 7\rangle$ and returns seven valid IDs in all 500 states. Excluding the 12 affected states leaves a Complete-MSS advantage of 5.74 points at five items, with a repository-clustered 95% interval of $[+2.40, +10.22]$.

The two no-feedback rows are separate three-stage runs with inter-stage visibility disabled. Their objective, controller allocation, route depth, candidate pool, stage caps, card budgets, fusion, and 6,144-token ceiling are unchanged; stages two and three read static ranked prefixes, and the three

proposals are fused by reciprocal-rank fusion. One fixes the final count at seven; the other adapts it between four and eight, returning four, five, six, seven, and eight units for 17, 31, 70, 85, and 297 states, respectively. Both cover 500 states over 1,500 logical calls with a recorded configuration hash, and no row in the table has private-label access at inference.

The derived rows are reconstructed offline from the corresponding frozen main-method traces. Single-stage uses the stage-one `initial_ids` in recorded order and inherits nothing from the final-stage ordering. Fixed top-8 uses the complete final ranking and preserves its first five units. The $\lambda = 1.0$ MMR row uses the corresponding Qwen ranking. These derivations make no additional API calls. All 16 released methods have 500 unique states and valid candidate IDs, and the consistency audit verifies cached completion, group-recall, and necessity-recall scores against the same current certificate file at every reported cutoff.

E.3 Post-Freeze Parameter Neighborhood

The formal neighborhood changes route depth or candidate-pool width while keeping the prompt, controller allocation, scorer, and Cal120 state set fixed. Seven earlier compatible runs provide a broader neighborhood. They share the same Cal120 states, three-call family, scorer, and label-hidden inference, but may change more than one geometric or prompt factor; the strict rows isolate individual factors. Table 18 reports both cohorts.

Table 18: Cal120 post-freeze robustness. Strict rows change one declared factor; broad rows reuse compatible historical configurations. Ret. is returned completion and Nec.@5 necessity-weighted recall at five items. Values are percentages except context and seconds.

Setting	Config.	Comp.@5	Comp.@8	Ret.	Group@5	Nec.@5	Context	Sec.
Frozen center	strict	65.00	75.83	75.83	79.65	79.64	3,178	6.315
Route depth 48	strict	62.50	73.33	73.33	76.39	76.58	3,217	6.470
Candidate pool 320	strict	64.17	74.17	74.17	79.24	79.42	3,171	4.877
Broad A	historical	65.00	74.17	74.17	77.22	77.52	3,362	4.725
Broad B	historical	68.33	75.00	75.00	79.93	80.16	3,217	5.017
Broad C	historical	65.83	72.50	72.50	80.49	80.68	3,214	5.630
Broad D	historical	66.67	74.17	74.17	80.76	81.11	3,185	5.218
Broad E	historical	65.00	74.17	74.17	78.75	78.91	3,262	5.127
Broad F	historical	60.83	68.33	68.33	75.35	75.59	3,234	5.153
Broad G	historical	62.50	71.67	71.67	76.74	77.18	3,233	5.454

The strict neighbors preserve the qualitative operating point: complete-set performance changes by at most 2.5 points at five and eight items, with similar context. The broader neighborhood spans 60.83–68.33% at five items and 68.33–75.00% at eight. Repository-clustered paired intervals and the exact configuration mapping are included in the artifact ledger.

E.4 Identity and Leakage Audit

The frozen Test500 prediction file contains 500 unique state IDs and 1,500 logical controller calls. Every row records `labels_read_at_inference=false`, with controller and parsing fallback counts at zero. The current offline audit recomputes all reported retrieval cutoffs against the frozen labels and checks the source traces used by the derived controls. The contract records configuration, implementation, public-input, private-label, and prediction hashes. A historical network-disabled replay additionally reproduced its corresponding source-run prediction file byte for byte. That replay establishes the source implementation, and the offline audit above covers the final merged release.

F Supporting External and Action Evidence

The following evaluations examine the evidence-acquisition objective beyond the main candidate-pool test. Prospective44 checks the configured policy on new states. AMA-Bench measures answering with the selected working context and accounts for the computation used to construct it. The action evaluation measures next-step repair localization from the supplied evidence, and Fresh23 carries the comparison through to executed repository tests.

F.1 PROSPECTIVE CONFIRMATION

Prospective44 uses the same state schema and grouped-MSS scoring definition, with its own frozen candidate construction. Its 44 states were collected after the freeze and were not used for method selection. At five items, MSS-COMPLEMENT completes 20/44 states (45.45%) versus 18/44 (40.91%) for SSR; its group and necessity-weighted recalls are both 46.21%. The mean returned context is 1,155 source tokens, and all 132 controller calls complete without fallback or private-label access.

F.2 AMA-BENCH CONTROLLED ACCOUNTING

The controlled AMA comparison is a 208-episode real-world evaluation with 12 questions per episode. MSS-COMPLEMENT runs the Cal500 configuration with no AMA-specific tuning, so all 208 episodes are evaluated out of sample for the method. Every row uses DeepSeek-v4-flash for answer generation and a locally deployed Qwen3.6-35B-A3B for correctness judging. The complete token accounting is in Table 3. The AMA paper reports 57.22% for AMA-Agent under its original generation and judging stack (Zhao et al., 2026b); the shared stack used here gives 53.97%, so the two values use different generation and judging stacks.

For the merged MSS-COMPLEMENT run, the answer and online totals include cumulative billed usage from the retained source runs, so the online column is an acquisition cost rather than the cost of one answer. The complete paired result is 359 gains, 1,830 ties, and 307 losses relative to AMA-Agent. The accuracy difference is $+2.08$ points with an episode-clustered 95% interval of $[-0.16, +4.37]$. The answer prompt is 76.2% smaller. Recorded full-run retrieval latency is 4.75 seconds per question for MSS-COMPLEMENT and approximately 6.94 seconds for AMA-Agent, about 31.5% lower in this evaluated deployment. These wall-clock values are deployment measurements; the efficiency statement concerns the answer-visible prompt, while the acquisition tokens that produce it remain in the online-total column.

F.3 CONTROLLER-FAMILY CONFIRMATION

The controller-family comparison holds the 44 states, candidate input, three-stage prompt semantics, evidence limits, and grouped-MSS scorer fixed. Only the controller model family changes. Table 19 reports the complete accounting.

Table 19: Prospective44 controller-family confirmation. Source context and controller tokens are means per state. Latency is provider-dependent.

Controller	Complete@5	Group@5	Context	Ctrl. tok.	Fallback
DeepSeek	45.45	46.21	1,155	43,696	0
Claude	**52.27**	**53.03**	1,168	63,999	0

Each run covers 44/44 states. Neither run reads organizer-private certificates during inference. The Claude run costs $5.78 through OpenRouter. Recorded mean retrieval latency is 6.79 seconds for DeepSeek and 16.07 seconds for Claude, but the provider, serving path, and billing endpoint differ, so this number is not used to rank the controller families.

F.4 ACTION52: RETRIEVAL TO A SCORABLE DOWNSTREAM DECISION

Purpose and task construction. Action52 evaluates whether evidence-set recovery is useful for completing the next agent task. It separates three objects: the evidence returned by a retriever, the production-code targets proposed by an executor, and the reference files used by an offline evaluator. A retrieval score is therefore not reused as the downstream outcome. The executor must interpret the retrieved material and produce its own repair-location decision.

The cohort contains 52 task instances from 52 repositories. Each has an intermediate state, a source-grounded certificate, and a reference patch that passes executable preflight. For 32 cases, the target files come from a constructed minimal residual reference repair. For the other 20, they come from the retained files in the validated historical reference patch. These references define the file set scored for

the next repair. They are retained by the organizer and are not shown as candidate answers. The main analysis retains all 52 cases.

Reference scope. The reference set contains 156 file entries, and the reported endpoint is coverage of that reference set rather than of production code alone. Per-case reference sizes and path composition are preserved in the accompanying audit.

Evidence conditions and budgets. Both learned retrievers receive the materialized residual evidence pools for the independent Action52 states, with observed evidence excluded. They do not perform a new full-repository discovery run. Action52 uses its own certificates and declared returned-set budgets, so its completion percentages are not an additional Test500@5 result. Qwen3-Embedding-8B uses the structured-state query without a reranker, and the reranked condition applies Qwen3-Reranker-8B to that embedding's leading 100 candidates. MSS-COMPLEMENT uses the final retrieval output. The three retrieval conditions contain the same number of evidence units for every case, between five and eight, with 395 units across the 52 cases in each condition. The evidence renderer enforces a 6,144-source-token ceiling per case. Observed evidence is excluded. Identical materialized bundles are used by both downstream executors.

Oracle-MSS is constructed by the organizer from the certificate. It selects an admissible source unit for each required group, deduplicates shared units, and fills the remaining slots from the Qwen ranking. One case needs one more unit than the adaptive method returned to preserve complete coverage. The oracle therefore contains 396 units in total and still obeys the eight-unit ceiling. The executor sees the source material, not group labels, a certificate, or a reference patch.

The Oracle-minus-one-required-group condition removes the group with greatest necessity weight, breaking ties by group identifier. All registered alternatives for that group are excluded, and the vacated slots are filled from the ranking with other source units. It preserves the Oracle's per-case item count and token ceiling and is supplied unchanged to both executor families. Complete-MSS is zero by construction while mean group recall stays at 48.27%, which is what makes this a paired missing-required-group control with substantial evidence still in place.

For a certificate containing g required groups, let h_j indicate whether the supplied evidence satisfies group j. In Action52 each group requires one admissible source unit. Group recall is $g^{-1}\sum_{j=1}^{g} h_j$, while Complete-MSS is $\prod_{j=1}^{g} h_j$. A three-group case with two groups recovered therefore scores $2/3$ on group recall and zero on Complete-MSS. We average each score over tasks rather than pooling groups across tasks. These are certificate-based scores. Because overlapping source windows can carry the same fact under different identifiers, the removal check inspects the materialized source content and the shared state alongside the removed identifiers. For four confirmed duplicate-window cases, the removal also excludes overlapping target implementation spans and matching implementation signatures before filling the vacated slots. The original target group and all other groups are preserved, and both executors use the corrected views. Their reference-file scores are unchanged after this correction. Public state clues, including an issue's existing repair suggestion, remain shared across conditions. The accompanying source audit retains the remaining partial-overlap flags, so the scope of each removal is recoverable case by case.

Executor contract. The executor receives the repository name, public state, evidence text and identifiers, and a target-count cap. It is asked to choose production-code targets to inspect for the next repair step and returns JSON containing relative paths, optional symbols, a short intent, invariants, and evidence citations. It may propose a new production file. The prompt instructs the executor not to choose tests, fixtures, snapshots, CI files, or evaluator files. Paths are generated freely, with no answer menu, source-window lookup, repository search, patch generation, or test feedback in this run.

For a reference repair with r retained files, the common target cap is $\min(6, \max(3, r))$. This organizer-derived scalar is supplied equally to all conditions, while reference paths and patch contents remain hidden. Path normalization and path filtering are deterministic. Extra paths consume the cap without incurring a separate precision penalty. An empty valid selection receives zero recall and zero success.

DeepSeek-v4-flash uses the direct API and Claude Sonnet 5 uses OpenRouter. Both use temperature zero, a 900-token response limit, and the same decision prompt and JSON contract. Up to three

format attempts request a corrected or shorter JSON object without adding task evidence. Every condition has 52 final case records, with no provider substitution. The DeepSeek missing-group condition contains one terminal empty selection, retained as a zero-score case. The experiment changes the downstream executor family while holding the retrieval outputs fixed, which keeps it distinct from the retrieval-controller experiment in Appendix F.3.

Target-level adjudication and equal-budget scoring. The executor may return fewer targets than its cap, so the number of emitted targets is not forced to be equal after inference. Every emitted target is reviewed by a human against the source, state, and approved repair-localization scope. A *precise* target identifies a correct and sufficiently local function, branch, statement, or insertion relation. A *coarse* target is related but does not isolate an actionable location; a *wrong* target is unsupported or irrelevant. Six proposals were explicitly adjudicated as not precise without forcing a finer coarse-versus-wrong label.

Traditional target precision divides precise targets by the number actually emitted, and its denominator ranges from 161 to 169 across conditions. Table 21 retains it for reference. The main table instead reports *fixed-budget target precision*: the number of precise targets divided by the sum of the predeclared per-case caps. Because all conditions use the same 52 cases, this denominator is 171 for every row. An unused slot receives zero credit. This preserves the executor's decision to return fewer targets without deleting, padding, or relabeling any generated proposal.

Deterministic endpoints. For case i, let R_i be the organizer's reference-file set and P_i the executor's valid predicted paths. Reference-file recall is $|P_i \cap R_i|/|R_i|$, averaged equally across all 52 cases. Localization success is the indicator that $R_i \subseteq P_i$. Complete-MSS and group recall are computed separately from the supplied evidence using the certificate. Optional symbols and target rationales enter the target-level adjudication but not the deterministic path-coverage score.

Four cases require more reference files than the six-target cap permits. They remain in every all-case measure, in Table 2 and in Table 20. For completeness, Table 20 also gives all-file success on the 48 cap-eligible cases. The stored field calls this `strict_file_success`, which denotes a different denominator rather than full patch-test success.

Table 20: Additional Action52 accounting. Complete count is out of all 52 cases; cap-eligible success uses the 48 cases whose reference file set fits the target cap. Reference-file recall is the task-macro mean. All ten conditions use the same states.

Executor	Evidence	Complete count	Cap-eligible success	Reference-file recall (%)
DeepSeek	Oracle − 1 group	0/52	32/48	68.53
DeepSeek	Qwen3 Embedding	11/52	33/48	70.32
DeepSeek	Qwen3 + Reranker	17/52	34/48	71.56
DeepSeek	MSS-COMPLEMENT	27/52	36/48	76.01
DeepSeek	Oracle-MSS	52/52	38/48	80.90
Claude	Oracle − 1 group	0/52	37/48	78.84
Claude	Qwen3 Embedding	11/52	36/48	77.96
Claude	Qwen3 + Reranker	17/52	36/48	77.35
Claude	MSS-COMPLEMENT	27/52	38/48	79.60
Claude	Oracle-MSS	52/52	39/48	82.34

Paired effects and interpretation. Within each executor family, the state, decision prompt, cap, and scorer are fixed while the evidence changes. Across the retrieval and Oracle-MSS rows, the fixed-budget target-precision estimates follow the listed group-recall ordering in both families. The missing-group control is the exception: its group recall is below the reranked baseline's while its precision is above it under both executors. The Qwen, MSS-COMPLEMENT, and Oracle rows also increase in Complete-MSS, mean reference-file recall, and all-file success. DeepSeek's full-coverage sequence over all 52 cases is 63.46%, 69.23%, and 73.08% at completeness levels 21.15%, 51.92%, and 100%; Claude's is 69.23%, 73.08%, and 75.00% at the same levels. These percentages use all 52 cases, including the four cases exceeding the six-target cap. The cap-eligible success column instead uses 48 cases with the same numerators.

Table 21: Action52 target-level adjudication. Raw precision uses the number of targets actually emitted. Fixed-budget target precision uses the common 171-slot denominator and is the equal-budget value reported in Table 2. "Other" denotes an explicit non-precise verdict without a forced coarse-versus-wrong subtype.

Executor	Evidence	Precise	Coarse	Wrong	Other	Raw precision (%)
DeepSeek	Qwen3 Embedding	83	62	22	0	49.70
DeepSeek	Qwen3 + Reranker	88	57	17	0	54.32
DeepSeek	Oracle − 1 group	90	55	18	0	55.21
DeepSeek	MSS-COMPLEMENT	110	52	7	0	65.09
DeepSeek	Oracle-MSS	111	54	2	0	66.47
Claude	Qwen3 Embedding	93	61	8	0	57.41
Claude	Qwen3 + Reranker	93	56	12	0	57.76
Claude	Oracle − 1 group	95	55	10	4	57.93
Claude	MSS-COMPLEMENT	100	54	6	2	61.73
Claude	Oracle-MSS	114	42	5	0	70.81

The Oracle-minus-one condition supplies a paired evidence-removal check. Its Complete-MSS is zero because every task excludes one group's registered alternatives, although it retains partial evidence and 48.27% group recall. Against the same-count Oracle, removing that one group costs 12.28 points of fixed-budget precision with DeepSeek and 11.11 with Claude, and lowers mean reference-file recall for both executor families. The executor still reads the visible state and interprets the target locations, so these retrieval metrics describe the evidence supplied to localization while target decisions remain executor outputs.

Table 22 retains uncertainty at the task level. We resample the same 52 paired task identifiers 10,000 times with seed 20260904 and report percentile intervals for mean reference-file recall. Each repository contributes one task. The DeepSeek oracle-group contrast has a wholly positive interval. Both learned-retrieval contrasts have positive point estimates with intervals spanning zero at this cohort size. Table 23 applies the same procedure to the reranked control and adds fixed-slot target precision. Pairing against the reranked condition is tighter than against plain embedding because the two selections agree on most cases, so the MSS-COMPLEMENT contrast separates there on a smaller point estimate.

Table 22: Paired Action52 differences in mean reference-file recall. Intervals are in percentage points and resample tasks, not condition means.

Executor	Evidence comparison	Difference (pp)	95% interval
DeepSeek	MSS − Qwen	+5.69	$[-2.41, +14.26]$
DeepSeek	Oracle − MSS	+4.89	$[-1.92, +12.58]$
DeepSeek	Oracle − missing group	+12.36	$[+4.18, +21.81]$
Claude	MSS − Qwen	+1.64	$[-3.53, +8.05]$
Claude	Oracle − MSS	+2.74	$[+0.00, +7.40]$
Claude	Oracle − missing group	+3.50	$[-1.44, +9.76]$

Table 23: Paired Action52 differences against the reranked control, matching Table 22; the target-precision block also gives Oracle − missing group. Differences are in percentage points. W/T/L counts cases on which the left condition is better, equal, or worse.

Executor	Comparison	Difference (pp)	95% interval	W/T/L
Reference-file recall				
DeepSeek	Reranker − Embedding	$+1.24$	$[-5.17, +7.69]$	2/47/3
DeepSeek	MSS-COMPLEMENT − Reranker	$+4.44$	$[+0.12, +10.44]$	4/48/0
Claude	Reranker − Embedding	-0.60	$[-6.33, +5.17]$	2/48/2
Claude	MSS-COMPLEMENT − Reranker	$+2.24$	$[-2.88, +8.65]$	2/48/2
Fixed-slot target precision				
DeepSeek	Reranker − Embedding	$+2.92$	$[-5.14, +11.45]$	14/27/11
DeepSeek	MSS-COMPLEMENT − Reranker	$+12.87$	$[+5.39, +20.36]$	18/29/5
DeepSeek	Oracle − missing group	$+12.28$	$[+4.27, +20.34]$	20/26/6
Claude	Reranker − Embedding	0.00	$[-6.83, +6.63]$	11/30/11
Claude	MSS-COMPLEMENT − Reranker	$+4.09$	$[-1.89, +10.29]$	13/32/7
Claude	Oracle − missing group	$+11.11$	$[+5.20, +17.51]$	15/35/2

F.5 END-TO-END REPAIR ON FRESH23

This appendix gives the full accounting behind Table 2. Fresh23 is a frozen 23-task cohort of 21 Windows tasks and two macOS tasks, one task per repository. Each task runs the whole loop. The retrieval method supplies evidence, the executor proposes a patch, the patch is applied to an executable repository snapshot, and the repository's own tests decide the outcome. Every condition covers all 23 tasks, issues two or three iterative retrieval calls per task, returns at most five source units per call, and has no retrieval fallback. The per-task executor assignment is identical across conditions: 20 tasks use DeepSeek-v4-pro and three use DeepSeek-v4-flash. Strict resolution requires the fail-to-pass (F2P) tests to pass and the selected pass-to-pass (P2P) neighborhood to stay green; pass-to-pass is evaluated only where fail-to-pass passes.

The three retrieval conditions read no organizer-private label at inference time. The two Oracle conditions are organizer-constructed diagnostics that do read the private certificate, and they are reported as bounds on the target rather than as deployable methods. Oracle-MSS covers every required group; the minus-one-group condition removes one required group's registered alternatives from that same evidence and refills the vacated slots, so evidence volume and the rest of the pipeline are held fixed while certificate completeness is broken.

Table 24: Fresh23 end-to-end repair over 23 frozen tasks. Resolved columns are counts with percentages. F2P is fail-to-pass and P2P pass-to-pass, and P2P is evaluated only where F2P passes. Candidate target-file hit counts the tasks in which some attempted patch modifies at least one file the reference repair touches, including attempts that were not ultimately applied. Tokens are the mean per-task API total, prompt plus completion, and calls are the mean iterative retrieval calls. Bold marks the best value among the three deployable conditions.

	Resolved of 23				Cost	
Condition	Strict	F2P	P2P	Candidate target file	Tokens	Calls
Retrieval, no private label at inference time						
BM25	5 (21.74)	8 (34.78)	6	22	133,775	2.52
Qwen3 + Reranker	6 (26.09)	**10 (43.48)**	7	20	126,261	2.43
MSS-COMPLEMENT	**8 (34.78)**	**10 (43.48)**	**9**	20	**118,351**	2.52
Organizer-private certificate diagnostics						
Oracle − 1 group	7 (30.43)	8 (34.78)	8	23	98,121	2.48
Oracle-MSS	8 (34.78)	11 (47.83)	9	23	99,655	2.48

The minus-one-group contrast is the cleanest reading of the target on this cohort, and Table 25 gives every paired contrast with its interval. The two Oracle conditions share an executor, a token ceiling and an evidence volume, and differ only in whether one required group is covered. Removing it costs

Table 25: Paired Fresh23 contrasts on the same 23 tasks, with 20,000 task-clustered bootstrap draws per row and seed 20260912. Win, tie and loss count the tasks on which the first condition resolves and the second does not, both agree, and the second resolves and the first does not.

Contrast	Δ (points)	95% interval	W/T/L
Strict resolution			
MSS-COMPLEMENT vs Qwen3 + Reranker	$+8.70$	$[-13.04, +30.43]$	4/17/2
MSS-COMPLEMENT vs BM25	$+13.04$	$[-4.35, +30.43]$	4/18/1
Oracle-MSS vs Oracle $-$ 1 group	$+4.35$	$[-8.70, +17.39]$	2/20/1
Qwen3 + Reranker vs BM25	$+4.35$	$[+0.00, +13.04]$	1/22/0
Fail-to-pass resolution			
Oracle-MSS vs Oracle $-$ 1 group	$+13.04$	$[+0.00, +26.09]$	3/20/0
Oracle-MSS vs BM25	$+13.04$	$[+0.00, +26.09]$	3/20/0
MSS-COMPLEMENT vs BM25	$+8.70$	$[-8.70, +26.09]$	3/19/1
MSS-COMPLEMENT vs Qwen3 + Reranker	$+0.00$	$[-17.39, +17.39]$	2/19/2

three fail-to-pass resolutions and returns the condition to BM25's eight, and under the fail-to-pass criterion no task is resolved by the incomplete certificate but missed by the complete one.

MSS-COMPLEMENT reaches the Oracle's strict resolution from the lowest mean API token usage of the three deployable conditions. Its fail-to-pass count matches the reranked baseline at ten, while its pass-to-pass count is nine against seven: the two conditions achieve fail-to-pass success on the same number of tasks, and MSS-COMPLEMENT's repairs leave the surrounding tests green more often, which is what the strict criterion records. The cohort tests directly whether the benchmark's target predicts an outcome it does not control.

Candidate target-file hit does not order the conditions the way resolution does. BM25 reaches 22 of 23 while resolving five, MSS-COMPLEMENT and the reranked baseline both reach 20, and the Oracle conditions reach 23. Attempting an edit in a file the repair touches is not the same as covering what the decision requires, which is what the certificate records and what the resolution columns follow. One task, `graphistry__pygraphistry-661`, has an incomplete pass-to-pass environment and is therefore unresolved under the strict criterion in every condition even though its fail-to-pass and pass-to-pass checks pass throughout; that single task is why the pass-to-pass count in Table 24 exceeds the strict count by exactly one in every row. Wall-clock latency is recorded in the released runs but is not compared here: the reranked baseline serves its reranker locally while the remaining components call hosted endpoints, so its durations describe that deployment rather than the method. The token counts in Table 24 are deployment independent and carry the cost comparison.

G ARTIFACT, RESPONSIBLE USE, AND REPRODUCTION

The release separates a public archive, submitted as the anonymous supplementary material accompanying this paper, from an organizer-private evaluator that is not distributed with it. The public archive contains 500 labeled Cal500 states and 500 label-free Test500 states, together with sanitized candidate pools, schemas, scoring code, and reconstruction metadata; sanitization removes private fields, identities, and secrets, and does not filter observed evidence. The archive also carries the state-level observed-overlap assignments, the twelve-method stratified matrix, the aggregation script, and its source hashes and seed. Test500 public files contain no grouped certificate, acceptable alias set, necessity weight, gold role, next-action label, organizer-private filename, API key, gold patch, or Python bytecode. The organizer archive contains exactly 500 Test500 certificates together with reserve and excluded review records, and is marked not for publication.

Clean-environment reproduction covers the public scorer, SSR baseline, and a three-state MSS-COMPLEMENT smoke run. Full semantic evaluation requires a compatible structured-output endpoint. Candidate generation and scoring run locally in the declared Python/PyTorch environment. Frozen result JSON/CSV files, configuration contracts, and the separate reproducibility ledger map every reported result to its source artifact.

The Fresh23 end-to-end cohort ships as a separate frozen package. It holds one run record per method and task carrying the proposed patch, the executed test commands and their outcomes, and the

retrieval trace behind the supplied evidence, together with a per-task comparison table, an aggregate summary, and a SHA-256 digest for every file. Repository snapshots are identified by commit for reconstruction under upstream terms rather than bundled. It also carries the run records of the two organizer-constructed certificate diagnostics; the certificates those diagnostics read are not distributed. No deployable method reads an organizer-private label at inference time.

H Scope and Release Responsibilities

SERBench focuses on repository evidence from multi-language coding trajectories. The visible state schema and grouped-MSS scorer apply across agent frameworks and programming languages. Certificates specify sufficient alternatives for the current decision, allowing multiple minimal solutions and leaving the agent's internal reasoning unconstrained. This target complements long-context models: a larger memory remains available for discovery, while the selected working set is evaluated against the decision's unresolved requirements. The experiments measure this selection and its downstream use.

Model-assisted annotation retains item-level provenance and independent verification. Repository source remains governed by upstream licenses, and the release distributes reconstruction information rather than bundled snapshots, as described in Section G.

The paired action cohorts keep states, repository snapshots, tools, prompts, and evaluators fixed across evidence conditions. They connect state-conditioned evidence recovery to an application endpoint without changing the benchmark target. Private certificates remain held out from training.