CERTCOMPILE: PRESERVING EVIDENCE UNDER A DELIVERY BUDGET

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval agents often read more evidence than they can pass to a downstream reader. Selecting records by relevance alone can omit intermediate facts needed to support an answer. We present CERTCOMPILE, which records source spans and execution dependencies, prioritizes supporting records within a delivery budget, and verifies the evidence before returning an answer. Rejected executions fall back to the original agent. On 120 development questions, a 512-token execution budget and a 256-token delivery budget yield 12 valid accepted answers with dependency-first packing, compared with five for rank-order packing and one for forwarding all admitted evidence. Exact-match accuracy including fallback is 39.17%, 38.33%, and 35.83%, respectively. The policies agree when the budgets are equal. On 240 MuSiQue questions with two models, prioritizing model-generated citation dependencies does not improve answer quality over the same producer's ranking. These results support preserving execution dependencies when execution and delivery budgets differ, while showing that better evidence retention need not translate into comparable gains in answer accuracy or lower inference cost.

1 INTRODUCTION

A retrieval agent may inspect more records than a downstream reader or interface can receive. Consider a question asking when the composer of a song died. Answering it requires one record linking the song to its composer and another giving that person's death date. Retaining only the date loses the connection to the song.

Evidence selection must therefore account for dependencies between facts. Ranking records independently by relevance can discard an intermediate fact used during execution. Forwarding every retrieved record avoids this loss only when the records fit the delivery budget. We study this mismatch by placing separate limits on the evidence available during execution and the evidence delivered to the reader.

CERTCOMPILE uses the execution record to select evidence. A typed retrieval program links each step to the source spans it reads and the outputs of previous steps. The packer then prioritizes the records needed to support the final value. A verifier checks source membership, dependencies, versions, and the delivery budget. If the supporting records do not fit or a check fails, the original agent answers the request.

On 120 natural-language development questions, dependency-first packing retains more valid supporting evidence than relevance ranking or forwarding all admitted records when the delivery budget is smaller than the execution budget. The gain in overall answer accuracy is smaller because the fallback reader already answers some rejected questions. With equal budgets, the three policies produce identical outcomes. This comparison isolates the contribution of packing by holding execution and fallback fixed.

We also study program reuse and model-generated dependencies. A program library can provide a complete program or supply steps for a new one. Combining steps improves coverage in controlled recurrence experiments, but a constructor using the same primitives without a library matches that coverage on the natural questions. On 240 MuSiQue questions with two models, prioritizing generated citation dependencies does not improve answers over the same producer's sentence ranking. These controls distinguish evidence packing from the effects of program reuse and model-based selection.

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.

Our contribution is a retrieval execution and delivery protocol that preserves supporting records and their dependencies under a token budget, together with an evaluation of evidence retention, answer quality, and cost. We address three questions: **RQ1**, when does evidence packing improve retention and final answers? **RQ2**, what does a program library add to shared execution? **RQ3**, which delivery faults does verification detect, and what does the complete system cost?

2 Related work

Program reuse. Agentic Plan Caching extracts plan templates, while Compiled AI converts repeated workflows into validated code (Zhang et al., 2025; Trooskens et al., 2026). PreAct uses guarded state machines, verifies them before storage, and checks the live UI during replay; failed checks return control to the agent (Li, 2026). Query-conditioned reuse supplies procedures with bindings, applicability conditions, and verification requirements (Li et al., 2026). CertCompile focuses on preserving the evidence used by a retrieval program under a delivery budget. Its optional library supports both whole-program reuse and assembly from historical primitives.

Verified execution and termination. No Certificate, No Execution separates trace proposal, certification, and execution through evidence and conformance obligations (Liu Yanglet et al., 2026). Proof-Carrying Certificates specifies checks around probabilistic components and their trusted assumptions (Koomullil, 2026). Evidence-Carrying Termination binds required claims to a trusted contract and receipt ledger, checks scope, and replays closed transforms before completion (Liu, 2026). CertCompile checks both the retrieval execution and whether packing preserves its supporting records and their dependencies. Semantic errors in the predicted task contract are evaluated separately from runtime consistency.

Evidence under a context budget. Database provenance describes derivations and their supporting records (Buneman et al., 2001; Green et al., 2007). Sufficient Context distinguishes missing evidence from failure to use it (Joren et al., 2025); RECOMP and BRIEF study context compression (Xu et al., 2024; Li et al., 2025). Our packer retains the original records required by the selected execution, including their dependencies, within the delivery budget. Alternative derivations are evaluated offline.

Selective acceptance and calibration. Selective prediction studies coverage and error among accepted predictions (Geifman & El-Yaniv, 2019), including reliability under domain shift in QA (Kamath et al., 2020). Learn then Test accounts for selection among candidate risk-control rules (Angelopoulos et al., 2021). Guarantees under distribution drift require additional assumptions or coverage penalties (Barber et al., 2023). Our secondary calibration analysis tests binder thresholds on held-out route combinations and reports both conditional error and marginal false acceptance relative to the fallback agent.

3 CertCompile

CertCompile executes typed retrieval programs under an execution budget and packs their supporting records under a delivery budget (Figure 1). Verification either accepts the recorded final value—the fast path—or sends the original request to the unchanged agent.

3.1 Candidate programs and task binding

A retrieval program is $P = (V, E, \Theta, \Phi, \tau)$, where V contains typed retrieval operations, or primitives; E is their dependency DAG; Θ specifies parameters; Φ specifies preconditions; and τ gives the terminal type and operator. The natural-language implementation uses ordered two-hop chains with an identity terminal operator.

Before candidate selection, a binder predicts the route from the question. Deterministic seed binding then produces the task contract $C_x = (H(x), f_x, s_x, o_x, v_x)$: question hash, predicted route, typed seed, answer operator and version, and binder policy and version. Seed binding uses only the question and immutable corpus titles. Eligible programs must match f_x. The binder has no access to candidate identity, retrieved answers, support labels, or reference routes.

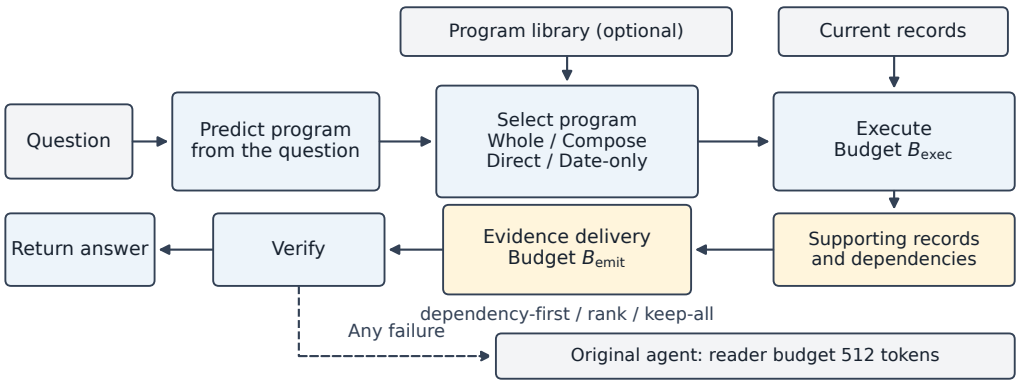


Figure 1: Retrieval execution and evidence delivery under separate budgets. The program library restricts which predicted programs are admitted. Delivery through authenticated storage is evaluated in a separate experiment.

Resolver v7 binds the unique longest context title named in the question and follows fixed relation patterns over admitted records. It abstains on missing or ambiguous matches and renders accepted terminal values canonically. Comparison questions fall back because they require a decision operator absent from the registry.

Optional program reuse. A program library supplies previously verified programs and primitives. WHOLE retrieves a compiled program; COMPOSE can also assemble primitives from different traces along routes allowed by the grammar. Both use the same execution, packing, verification, and fallback. DIRECT-TYPED constructs the predicted program without requiring library membership and otherwise uses the same components. This control isolates the library's admission rule. A second control admits only predicted Date terminals without consulting the library; it selects the same candidates as COMPOSE in the natural-language evaluation. Appendix A.7 gives exact solvers for allocating library storage.

3.2 EXECUTION RECORDS

Retrieval admits evidence records $A_x(P)$ within the execution budget B_{exec}. Primitive j emits $\rho_j = (i_j, y_j, L_j, D_j, v_j)$: typed inputs and outputs i_j, y_j, source spans and record hashes L_j, parent IDs and output hashes D_j, and source and executor versions v_j. Each non-initial input must equal its parent's typed output. All methods read current source records, including when they reuse a historical program.

The execution record, called `ProofObject` in the implementation, contains the contract hash, route fingerprint, ordered primitive records and edges, terminal primitive and value, packed-record IDs, tokenizer version, and measured token total.

3.3 DEPENDENCY-FIRST PACKING AND VERIFICATION

Let $L(P, x)$ contain the records supporting the terminal value and all its transitive dependencies. Execution requires $L(P, x) \subseteq A_x(P)$. Dependency-first packing places these records first, rejects if they exceed B_{emit}, and uses any remaining space for other admitted context. The implementation calls this policy CRITICAL-FIRST:

$$\widehat{M}_t^{P,x} = \text{Pack}_{B_{\text{emit}}}^{\text{critical}}\left(A_x(P), \text{Resolve}_t(P, x), L(P, x)\right).$$

Both budgets count serialized evidence lines with the serving tokenizer; execution-record metadata is measured separately. With equal budgets, $B_{\text{exec}} = B_{\text{emit}} = B$, forwarding the admitted sequence preserves $L(P, x)$. We rerun admission and execution at each B_{exec} before comparing delivery policies. The delivery budget applies to accepted executions; fallback uses its own 512-token input budget (Appendix E).

Table 1: Threat model. Trust refers to faithful execution of frozen components; source truth is assessed separately.

Component/status	Failure	Check or assumption
Candidate, library, text / untrusted	Wrong route, stale primitive, adversarial content	Current bytes, spans, hashes, and provenance are checked.
Binder, grammar / integrity-trusted	Incorrect binding	Frozen prediction/version checked; adequacy scored offline.
Registry, resolver, executor, packer, verifier, tokenizer, operator / trusted	Compromise or implementation error	Outside the correct-TCB assumption.
Source truth, reference labels / offline	False source or incomplete reference	Assessed separately from runtime checks.

The Full verifier accepts only when all four checks pass:

1. The question, predicted-contract hash, binder policy and version, answer operator, route fingerprint, and source, executor, and tokenizer versions match independently frozen values.
2. Primitive order, types, dependency edges, parent-output hashes, and terminal identity match the declared program.
3. Every cited span matches its offsets and hash in a current record admitted during retrieval and included in the package. All dependency records are present, and the total is at most B_{emit}.
4. The terminal primitive supports the final value, its rendered form, and its type. All three satisfy the frozen answer operator.

Failure invokes the same full agent used by ALWAYS-AGENT.

Trusted components. Table 1 defines the trusted computing base (TCB).

Under correct TCB execution and independently frozen reference values, acceptance guarantees the runtime checks in Proposition 1. Read-only attempts preserve the fallback state. Section 4 states the additional assumptions needed for semantic validity.

Delivery through storage. The main implementation verifies the execution record in memory. In a storage extension, a separate receiver loads the evidence and record from an authenticated envelope bound to the request and cache generation. The receiver obtains current records, contract, versions, and budget independently. We compare authentication and budget checks with Full verification and resolver replay under lifecycle faults and validly signed malformed records (Appendix F).

Replay and statistical admission. REPLAY-MATCHED adds a fresh run of the frozen resolver from the same seed and records, then checks every typed output. A separate controller calibrates a binder-margin threshold across route groups (Section 4.2). These development analyses leave the sealed evaluation unchanged.

Natural-language citation adapter. In the external study, a shared model predicts a citation DAG and sentence ranking from the question and admitted sources. Dependency-first packing prioritizes citations to the terminal and its ancestors; citation-first prioritizes all citations; LLM-rank uses only the ranking. Readers receive original sentences with titles and IDs. The adapter checks source membership and internal consistency of the predicted DAG and uses no program library.

4 ANALYSIS

4.1 RUNTIME CHECKS AND SEMANTIC VALIDITY

For a realization $\omega = (x, t, P, \xi, M, \widetilde{a})$, offline scoring evaluates route validity $T_x(P)$, dependency and current-source validity $L_t(P, \xi, M)$, reference support $E_x(M)$, and terminal binding $Q_x(P, \xi, \widetilde{a})$. With runtime type check K^{rt}, define

$$Z_x^{\mathrm{cv}} = T_x L_t E_x Q_x K^{\mathrm{rt}}. \tag{1}$$

The runtime verifier checks predicted task binding, recorded dependencies, packed current evidence, and terminal agreement:

$$G^{\mathrm{ans}} = G^{\mathrm{task}} G^{\mathrm{lineage}} G^{\mathrm{support}} G^{\mathrm{term}}. \tag{2}$$

Let I^{rt} denote the executable predicates in Section 3.

Proposition 1 (Acceptance under the trusted computing base)**.** *With correct trusted components and independently frozen reference values, acceptance implies $I^{\mathrm{rt}} = 1$ and emission of the recorded terminal value.*

Semantic soundness further requires $G^{\mathrm{ans}} \le Z_x^{\mathrm{cv}}$ and

$$Z_x^{\mathrm{cv}} \le Y_x(\widetilde{a}), \tag{3}$$

where Y_x denotes external answer correctness. We report acceptance G^{ans}, contract-valid acceptance $C_x^{\mathrm{cv}} = G^{\mathrm{ans}} Z_x^{\mathrm{cv}}$, correct acceptance $W_x^{\mathrm{ans}} = G^{\mathrm{ans}} Y_x$, and their conjunction

$$J_x^{\mathrm{ext}} = G^{\mathrm{ans}} Z_x^{\mathrm{cv}} Y_x \tag{4}$$

separately. Replay uses the same resolver and semantic assumptions.

If valid and invalid realizations share observation O, every universally sound O-measurable rule must reject both. Our paired tests construct such cases. More informative observations can increase sound coverage, whereas adding checks to a fixed gate can only preserve or reduce its acceptance set (Appendix A.1).

4.2 Fallback, coverage, and cost

Let σ_A be state readable by the original agent and r its randomness.

Proposition 2 (Rejection noninterference)**.** *A read-only attempt that rejects to the same agent with the original request, σ_A, and r preserves that agent's output pointwise.*

For a binary endpoint, let V be the attempted execution's validity, A the coupled agent result, and G acceptance. The hybrid is

$$H = GV + (1 - G)A. \tag{5}$$

Proposition 3 (Fallback accounting)**.** *If $G \le V$, then $H \ge A$. In general, without independence,*

$$\begin{aligned} \Pr(H = 1) - \Pr(A = 1) &= \Pr(G, V, 1 - A) - \Pr(G, 1 - V, A) \\ &\ge -\Pr(G = 1, V = 0). \end{aligned} \tag{6}$$

The marginal false-acceptance probability $\phi = \Pr(G = 1, V = 0)$ bounds the loss relative to fallback. Conditional error among accepted executions is $\rho = \phi / \Pr(G = 1)$ when acceptance is positive.

Appendix A.5 decomposes valid accepted coverage. The citation adapter measures reader quality for every packet. Fallback proofs appear in Appendix A.3.

Grouped admission. Binder-margin calibration assumes iid route groups from an unchanged population and targets route-prediction risk before subsequent verification (Appendix A.6).

Cost. Let C_{try} be attempt cost, C_A the coupled agent's cost, and C_C compilation cost amortized over M requests. The expected total cost is $\mathbb{E}[C_{\mathrm{try}} + (1 - G)C_A] + C_C/M$. For fixed C_A, mean attempt cost C_F, and acceptance $h = \Pr(G = 1)$,

$$\mathbb{E}[\bar{C}_H] = C_F + (1 - h)C_A + C_C/M. \tag{7}$$

A conservative condition for deployment is $\underline{h}^{+} C_A > C_F + C_C/M$ and $\overline{\phi} \le \eta$. Here $\underline{h}^{+}$ and $\overline{\phi}$ are simultaneous confidence bounds on valid coverage and marginal false acceptance, and η is the allowed loss budget (Appendix A.9). When fallback costs vary, the paired expectation must be used.

5 EXPERIMENTS

RQ1: evidence retention under separate budgets. The natural-language development set contains 120 questions, a fixed Qwen binder, resolver v7, and a nine-entry program library. For each question and method, we execute at $B_{\text{exec}} \in \{256, 512, 1024\}$, then compare dependency-first packing, rank-order packing, and keep-all delivery at $B_{\text{emit}} \in \{128, 256, 512\}$, with $B_{\text{emit}} \leq B_{\text{exec}}$. Keep-all forwards the admitted sequence unchanged or rejects if it exceeds the delivery budget. Eight budget pairs, three methods, and three policies give 72 conditions and 8,640 attempts. Each condition includes all 120 questions, including missing candidates and failed executions.

Acceptance is the runtime verifier's decision. A *valid accepted answer* also matches the reference route and exact support coordinates, has a compatible terminal type, and is correct under exact match (EM). This is the natural study's strict joint-validity measure. Hybrid EM includes both accepted answers and fallback answers.

All policies use Full verification and the same recorded 512-token fallback reader responses, pairing final answers without new inference. Runtime attempts are committed before offline scoring. The questions and configuration were available during development. Appendix D isolates packing on 36 completed executions; Appendix E reports all-question results. Live latency is measured separately below.

External evidence delivery. After development on 30 training questions, we freeze 240 MuSiQue-Ans development questions, with 80 each at two, three, and four hops (Trivedi et al., 2022). A fixed hash ordering of public IDs determines selection. Runtime receives only questions and original paragraphs; answers, decompositions, and support annotations are used for scoring after execution is committed. Qwen3-4B-Instruct-2507 (bfloat16) and Mistral-7B-Instruct-v0.3 (4-bit) use the same frozen policies. Evaluation uses the public development split rather than the hidden test set.

BM25 admits a common 4096-token sentence view. At delivery budgets of 256 and 512 tokens, we compare six policies: BM25, BGE, an adapted RECOMP extractive checkpoint, LLM-rank, citation-first, and dependency-first packing (Xiao et al., 2024; Xu et al., 2024). The last three share one producer output. Producer-direct, full-view reading, and question-only reading provide three references. Every packet is read, with no acceptance gate or fallback. Identical reader prompts share one recorded response to remove inference variation between identical packets.

The primary contrast is dependency-first packing versus LLM-rank at 256 tokens within each model. We report official answer EM/F1, paired hop-stratified bootstrap intervals, exact McNemar tests, and Holm adjustment across the two EM tests. Intervals clustered by shared-constituent components provide a prespecified sensitivity check. Each policy is charged its full producer, reader, and CPU ranking costs; component times reflect shared prefix and token-count caches. Actual shared-call usage is reported separately. Appendix G gives the protocol and complete results.

RQ2: exploratory program reuse. The controlled study crosses six topologies, three priors, three drift regimes, and eight seeds, giving 432 recurrence conditions with 16 episodes each. Episodes 1–4 form a common warm-up prefix; episodes 5–16 form the paired evaluation suffix. WHOLE and COMPOSE differ only in cross-trace assembly and share routing, execution, packing, verification, and fallback. Their completion difference is the prespecified primary endpoint; valid-fast-path coverage and cost are secondary. Intervals use 10,000 paired bootstrap resamples within topology/prior/drift strata. ALWAYS-AGENT measures the full-system contrast.

A separate sealed evaluation, committed before label access, pools 210 questions from 30 blocks and 13 route families. The 120-question development configuration fixes one model, warm size, and budget. It compares reuse with DIRECT-TYPED, a Date-terminal rule without a library, terminal- and primitive-support rules, and margin thresholds. A family-held-out sensitivity analysis selects thresholds on the other 12 families while keeping the program library fixed. We also report the sealed results and an eight-target blind evaluation.

RQ3: delivery verification and synchronous cost. A producer writes authenticated execution records to persistent storage; a separate receiver checks them against independently supplied trusted context. Authenticated delivery, Authenticated+Full, and Authenticated+Replay share envelope, context, and evidence-budget checks. Thirteen historical records from four families undergo clean

Table 2: Evidence delivery for Compose: $B_{\text{exec}} = 512$, fixed nine-entry program library. Each cell is valid accepted answers/all accepts out of 120 questions, including all upstream failures.

Delivery cap (tokens)	128	256	512
Keep all admitted evidence	0/0	1/2	12/13
Rank-order packing	1/1	5/6	12/13
Dependency-first packing	7/8	12/13	12/13

and benign delivery, seven source/cache/transport interventions, and two signed-producer tests. Additional diagnostics include 234 single-fault and 715 paired-fault tests (Appendix C.5).

A separate live run retrains the original binder on residual data, executes 120 warm-up requests to build a program library, and interleaves five methods on all 120 evaluation questions. Every rejected request makes fresh local Qwen3-4B planner and reader calls. Request timing covers binder inference, selection, retrieval, execution, packing, in-process Full verification, and synchronous fallback. Authenticated file delivery is timed only in the separate 13-record experiment. Acquisition, training, and session loading are recorded separately. The live study measures one pass over the development workload on an initialized service. Its acquired library has ten entries, compared with nine in the fixed replay. Appendix F reports costs and assumptions for projected amortization.

6 RESULTS

RQ1: retaining dependencies helps when budgets differ. At $B_{\text{exec}} = 512$ and $B_{\text{emit}} = 256$, dependency-first packing yields 12 valid accepted answers out of 120 questions, compared with five for rank-order packing and one for keep-all (Table 2). The seven additional valid acceptances over rank order span four families. An exploratory paired family-bootstrap interval for this 5.83-point coverage gain is [1.85, 9.09]. Hybrid EM is 39.17%, 38.33%, and 35.83%, respectively, against the same 35.00% baseline reader. The smaller accuracy gain reflects fallback answers that are already correct on some rejected questions.

All three policies agree question by question in every equal-budget condition. Increasing the execution budget from 512 to 1024 adds no acceptances. Direct-Typed matches Compose's strict valid-acceptance coverage under dependency-first packing without a program library. Appendix E reports the full budget grid, failures, and family effects.

Predicted citation dependencies give no additional gain on MuSiQue. At the primary budget (Table 3), dependency-first packing changes F1 relative to LLM-rank by -0.35 points for Qwen and -0.13 for Mistral. The question-bootstrap 95% intervals are $[-1.25, 0.15]$ and $[-1.00, 0.62]$; shared-component intervals are $[-1.32, 0.16]$ and $[-1.00, 0.64]$. Paired EM wins/losses are 0/1 and 0/0, with Holm-adjusted $p = 1$ for both. Dependency-first and citation-first packing produce identical packets on 222/240 Qwen and 236/240 Mistral questions. Qwen's gains over BM25 and BGE are also achieved by producer ranking alone; Mistral does not improve over BGE.

RECOMP matches dependency-first packing's Qwen EM and achieves higher Mistral F1, while dependency-first packing uses 12.38 and 13.15 times as many generator tokens. Including producer inference, the 256-token policy uses 31.26% and 29.76% more generator tokens than the respective full-view readers. Smaller delivery packets therefore cost more total inference in this study. Appendix G reports CPU encoding and shared-cache component times. Neither compact-producer training candidate meets its prespecified quality criterion.

RQ2: composition gains depend on the setting. At $B = 256$, Compose improves completion by 1.29 points and reduces suffix model tokens by 25.79% relative to Always-Agent (Table 4). The increment over Whole is 0.14 completion points, 2.55 valid-fast-path points, and 2.01% fewer model tokens. Shared-hub and shared-prefix structures gain 8.22 and 7.06 coverage points; the other four topologies tie (Figure 9).

On the natural-language development set, the Date-only control matches Compose on all 120 questions without a program library: 63 candidates, 13 acceptances, and 12 valid accepted answers.

Table 3: Answer EM/F1 (percent) on all 240 MuSiQue questions per model. The six packing policies have a 256-token delivery budget. Reference methods appear below. †Producer-direct and Full-exec-reader use the admitted 4096-token evidence view. Mistral uses 4-bit loading.

Policy	Cap	Qwen EM	Qwen F1	Mistral EM	Mistral F1
BM25	256	10.42	16.95	6.67	10.63
BGE	256	14.17	20.58	7.50	12.45
RECOMP adaptation	256	19.17	27.70	7.08	14.36
LLM-rank	256	19.58	28.82	6.67	11.64
Citation-first	256	19.58	28.82	6.67	11.52
Dependency-first	256	19.17	28.47	6.67	11.52
Producer-direct	4096†	15.00	21.80	2.50	8.35
Full-exec-reader	4096	22.50	32.61	8.75	14.88
Question-only	0	0.00	0.00	2.08	5.50

Table 4: Controlled paired suffix at $B = 256$: 432 recurrence conditions. Completion and valid-fast-path effects are percentage points; model-token changes are relative percentages. Brackets are 95% stratified-bootstrap intervals. The first row evaluates the full stack; the second isolates composition.

Contrast	Completion	Valid fast path	Model tokens
Compose–Agent	+1.29 [1.00, 1.60]	+44.02 [43.02, 45.06]	-25.79 [-26.40, -25.19]
Compose–Whole	+0.14 [0.06, 0.23]	+2.55 [2.18, 2.93]	-2.01 [-2.31, -1.72]

Direct-Typed yields 79, 14, and 12, respectively (Table 35). Terminal- and primitive-support rules also match Compose. Retaining the 13 highest-margin Direct acceptances leaves one wrong answer, which the Date-only rule removes. In a separate construction test that records fragment sources, Compose produces 12/120 jointly valid acceptances versus eight for Whole. A constructor using the same primitives without a library also produces 12 (Appendix H).

Resolution is the main bottleneck after candidate selection: 153 of 189 method–question attempts with a candidate fail, including 138 missing transition patterns. All three verification policies agree on the 360 natural-language attempts. In the sealed study, Compose adds four jointly valid questions over Whole, all in one family. Neither Whole nor Compose produces a valid accepted answer on the eight-target blind evaluation; every method, including the fresh baseline, has zero final-answer EM.

RQ3: delivery checks and measured cost. Across 13 stored execution records, all three authenticated-delivery policies preserve clean and benign cases and reject every tested source-update, stale-cache, cross-request, and transport fault. Full and Replay detect no additional lifecycle faults beyond authentication. For validly signed records with malformed dependencies, authentication accepts all 13, whereas Full and Replay reject all 13. This test relaxes the correct-producer assumption. Mean clean-receiver times are 1.76, 2.45, and 5.60 ms, including file reading and checks but excluding process startup and trusted-context loading.

Table 5 includes all online model calls in the live run. Compose reduces mean request time from 1.135 to 1.012 s and model tokens by 11.52% relative to Always-Agent. Direct costs slightly less and reaches the same valid coverage, with one additional accepted wrong answer. Acquiring the program library costs 138.67 s and 123,174 model tokens; fitting and calibrating the binder costs another 177.51 s. These costs exceed the savings over the measured 120 requests. With the same request mix, acceptance rate, and unchanged sources, the projected break-even points are 2,566 requests for time and 1,010 for model tokens.

7 Discussion

The results distinguish evidence retention from answer quality and total cost. In the typed development study, preserving execution dependencies helps when delivery has a smaller budget than execution. The fallback reader narrows the resulting accuracy gain. With equal budgets, simply forwarding the admitted evidence retains the same dependencies.

Table 5: Live run: 120 requests per method, a ten-entry program library, execution/delivery budgets of 512/256 tokens, and a 512-token planner and reader fallback. Time is observed mean online latency on an initialized service; tokens include all online model calls.

Method	Joint	EM (%)	Model tokens	Time (s)
Always-Agent	–	36.67	127,067	1.135
Whole, dependency	8	40.00	117,011	1.048
Compose, rank	5	40.00	121,009	1.082
Compose, dependency	12	40.83	112,428	1.012
Direct, dependency	12	40.00	111,035	1.002

The source of dependency information also matters. Typed execution records identify the records used to compute a result. On MuSiQue, generating a citation graph costs additional inference but does not improve answers over the same producer's ranking. The cost of generating that graph exceeds the reader tokens saved by compression. An existing execution trace avoids this additional generation step.

Program reuse and verification address separate concerns. Combining stored steps expands coverage relative to whole-program reuse, but a constructor using the same public primitives matches the valid answers on the natural questions. Recurring shared structures help in the controlled study; the natural implementation is concentrated in Date-valued two-hop chains. Verification detects the tested delivery faults and malformed signed records, while answer correctness still depends on source reliability and the resolver's interpretation. Additional structural checks leave the natural-task answers unchanged.

The experiments use public data, a restricted typed resolver, and specific model configurations. The eight-target blind evaluation produces no correct answers for any method, including the fresh baseline. MuSiQue's paragraph-level support labels do not identify every necessary sentence, and the live cost study covers one pass on an initialized service. Amortization projections assume stable request costs and acceptance rates. Broader answer types and full deployment measurements are needed to assess where the observed delivery benefit remains useful.

REPRODUCIBILITY STATEMENT

The artifacts include configurations, seeds, source snapshots, model and tokenizer versions, request logs, scoring code, and SHA-256 manifests. Replay checks source membership, complete-record packing, and token budgets. Binder validation requires exact discrete outputs and compares floating-point outputs within the stated tolerances. Appendix B.7 lists the runs and environments. All evaluated language models are hosted locally.

For the sealed natural evaluations, the data exclusions, execution code, models, tokenizers, and scoring rules were fixed before inference. Predictions are committed before label access. The Qwen3-8B transfer audit and subsequent development experiments are identified separately. The eight-target study excludes target identities from binder fitting and records both model outputs before a joint scoring step. Its diagnostic interventions use reference information after scoring; their inputs and outputs are retained.

The controlled study provides the paired summaries for all 432 conditions, bootstrap contrasts, figures, and table macros. The budget study records 8,640 attempts across 72 conditions, including failures. The live cost study records 120 acquisition requests, 600 online requests, and 1,347 model calls, including one warm-up call. Its logs support the reported acquisition, training, online inference, and verification costs.

MuSiQue artifacts contain the panel selection, policy and model settings, sentence packages, predictions, official EM/F1 scorer, paired statistical analyses, and producer and reader usage. Per-policy cost allocations are reported separately from actual shared-call usage. Development variants retain their selection criteria and unsuccessful outcomes. The program construction and sentence-selection diagnostics include the selected source records and the hashes needed to compare outputs.

The source archive includes the local LaTeX dependencies needed to rebuild the PDF. Appendix protocols identify the experiment artifacts and specify which information is available during selection, execution, and scoring.

ETHICS STATEMENT

The synthetic benchmark contains procedurally generated records. The external experiments use public HotpotQA, 2WikiMultiHopQA, and MuSiQue data (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022). We collect no private user data.

The verifier checks execution records and delivered evidence. Consequential applications also require source validation, access controls, domain review, and assessment of downstream use.

GENERATIVE AI USE STATEMENT

The authors used OpenAI Codex to assist with literature search; construction and checking of synthetic benchmark data; and language editing. The authors checked all AI-assisted outputs, including retrieved sources, synthetic benchmark data, and the edited text and citations. Codex did not annotate evaluation targets or act as an evaluated reader. Qwen3-4B-Instruct-2507, Qwen3-8B-AWQ, Mistral-7B-Instruct-v0.3, and Phi-4-mini-instruct are reported experimental systems. The authors take responsibility for the paper and its artifacts.

REFERENCES

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control, 2021. URL https://arxiv.org/abs/2110.01052. Revised September 2022.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. doi: 10.1214/23-AOS2276. URL https://doi.org/10.1214/23-AOS2276.

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *Database Theory — ICDT 2001*, volume 1973 of *Lecture Notes in Computer Science*, pp. 316–330. Springer, 2001. doi: 10.1007/3-540-44503-X_20. URL https://doi.org/10.1007/3-540-44503-X_20.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2151–2159, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 31–40. Association for Computing Machinery, 2007. doi: 10.1145/1265530.1265535. URL https://doi.org/10.1145/1265530.1265535.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps, 2020. URL https://arxiv.org/abs/2011.01060.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Jjr2Odj8DJ.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, 2020. doi: 10.18653/v1/2020.acl-main.503. URL https://aclanthology.org/2020.acl-main.503/.

George Koomullil. Proof-carrying certificates for LLM pipelines: A trust-boundary architecture, 2026. URL https://arxiv.org/abs/2605.16407.

Bojie Li. PreAct: Computer-using agents that get faster on repeated tasks, 2026. URL https://arxiv.org/abs/2606.17929.

Yifei Li, Heng Wang, Lingling Zhang, Muye Huang, Xinyu Zhang, Jiashuai Liu, Hang Yan, and Rongman Xu. Beyond retrieval: Query-conditioned reuse of long-horizon agent trajectories, 2026. URL https://arxiv.org/abs/2608.12847.

Yuankai Li, Jia-Chen Gu, Di Wu, Kai-Wei Chang, and Nanyun Peng. BRIEF: Bridging retrieval and inference for multi-hop reasoning via compression. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5464–5485. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.301. URL https://aclanthology.org/2025.findings-naacl.301/.

Jason Liu. When may an agent stop? evidence-carrying termination for tool-using LLMs, 2026. URL https://arxiv.org/abs/2608.23623.

Xiao-Yang Liu Yanglet, Xiaodong Wang, and Agostino Capponi. No certificate, no execution: Certified traces as a foundation for trustworthy AI agents, 2026. URL https://arxiv.org/abs/2605.24462.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Geert Trooskens, Aaron Karlsberg, Anmol Sharma, Lamara De Brouwer, Max Van Puyvelde, Matthew Young, John Thickstun, Gil Alterovitz, and Walter A. De Brouwer. Compiled AI: Deterministic code generation for LLM-based workflow automation, 2026. URL https://arxiv.org/abs/2604.05150.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649. Association for Computing Machinery, 2024. doi: 10.1145/3626772.3657878. URL https://doi.org/10.1145/3626772.3657878.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mlJLVigNHp.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Qizheng Zhang, Michael Wornow, Gerry Wan, and Kunle Olukotun. Agentic plan caching: Test-time memory for fast and cost-efficient LLM agents, 2025. URL https://arxiv.org/abs/2506.14852.

A Proofs and extended analysis

A.1 Observable refinement

Theorem 1 (Observable refinement)**.** *For a finite realization set Ω, validity predicate Z, and gate observation O, define*

$$g_O^\star(o) = \mathbf{1}\{\{\omega : O(\omega) = o\} \subseteq Z^{-1}(1)\}.$$

Then $g_O^\star \circ O$ pointwise dominates every universally sound gate measurable through O. Its coverage is

$$h^\star(O; Z) = \Pr\{\Omega_O(O(\omega)) \subseteq Z^{-1}(1)\}. \tag{8}$$

If O' refines O, then $h^\star(O; Z) \le h^\star(O'; Z)$.

Group realizations with the same observation into fibers. For ideal, trusted task, lineage, support, and terminal signals, the chain

$$O_0 \preceq (O_0, T) \preceq (O_0, T, L, E) \preceq (O_0, T, L, E, Q) \tag{9}$$

gives nondecreasing upper bounds on universally sound coverage. These are oracle bounds unless the deployed binder has access to the corresponding trusted signals.

Full proof of Theorem 1. If $g_O^\star(O(\omega)) = 1$, the fiber containing ω is a subset of $Z^{-1}(1)$. Hence $Z(\omega) = 1$, which proves universal soundness.

Let $g \circ O$ be universally sound. If $g(o) = 1$ while the fiber of o contains ω_0 with $Z(\omega_0) = 0$, the gate accepts ω_0, a contradiction. Thus $g(o) \le g_O^\star(o)$ for every o. Taking expectations gives Equation 8.

If $O = f \circ O'$, then

$$\Omega_{O'}(O'(\omega)) \subseteq \Omega_O(O(\omega)).$$

Every all-valid coarse fiber has all-valid refined subfibers, so $g_O^\star(O(\omega)) \le g_{O'}^\star(O'(\omega))$ pointwise. Expectations give monotonicity, with strict inequality exactly when the indicators differ on a set of positive μ-mass: the coarse fiber has mixed validity and the refined fiber is all-valid. □

This result establishes pointwise soundness over the fixed set Ω. It does not provide an average-risk optimum or a finite-sample estimate of $h^\star$.

A.2 ANSWER-DEPENDENT INFORMATION

Proposition 4 (Answer-dependent information is necessary)**.** *Fix x, P, ξ, M. Suppose answers a_0, a_1 have the same answer-independent observation O_0, while $Z^{\mathrm{cv}}(x, P, \xi, M, a_0) = 1$ and $Z^{\mathrm{cv}}(x, P, \xi, M, a_1) = 0$. Every universally contract-sound gate using only O_0 rejects both. Accepting the first and rejecting the second requires an observation that distinguishes their answer support.*

Proof. The two realizations share an O_0-fiber of mixed validity, which the maximal sound gate in Theorem 1 rejects. Separating them requires observing the answer or trusted evidence that binds it to the terminal computation. □

The same argument applies to task binding, dataflow, and evidence support. A candidate can satisfy one task contract but violate another, and the same records can appear in executions with different dependencies. Even valid dataflow may lack the required evidence, which the support signal S identifies. Distinguishing these cases requires task binding independent of the candidate, recorded dependencies, and current supporting evidence.

A.3 FALLBACK PROOFS

Proof of Proposition 3. Fix a model stack θ and endpoint ℓ. Suppressing x and θ, Equation 5 gives

$$H^\ell - A^\ell = G^\ell(V^\ell - A^\ell).$$

Under soundness, acceptance implies $V^\ell = 1$, so the right-hand side is $G^\ell(1 - A^\ell) \geq 0$. This proves pointwise dominance.

Without soundness, partition accepted realizations by (V^ℓ, A^ℓ). Terms with equal outcomes cancel, leaving

$$\begin{aligned}\Pr(H^\ell = 1) - \Pr(A^\ell = 1) = \Pr(G^\ell = 1, V^\ell = 1, A^\ell = 0) \\ - \Pr(G^\ell = 1, V^\ell = 0, A^\ell = 1).\end{aligned}$$

The negative event is contained in $\{G^\ell = 1, V^\ell = 0\}$, which proves Equation 6. □

The proposition compares binary outcomes for a model and its own fallback; it does not cover cross-model comparisons or token F1.

A.4 AVAILABILITY UNDER OPTIONAL PROGRAM REUSE

Let route j occur with probability π_j during warm-up and q_j at test time. Warm requests are iid and independent of the test route. A matching request installs its representative with probability a_j; given presence, it is executable at test with probability r_j, independently of the warm history. Then

$$h_n^{\mathrm{route}} = \sum_j q_j r_j \left[1 - (1 - \pi_j a_j)^n\right]. \tag{10}$$

One warm-up request misses route j with probability $1 - \pi_j a_j$. After n iid requests, its presence probability is $1 - (1 - \pi_j a_j)^n$. Multiplying by independent test executability r_j and route mass q_j gives Equation 10. Multiple independently surviving copies would require the probability that at least one survives.

For composition, let $L_1, \ldots, L_n$ be iid sets of learned verified primitives, $S_n = \cup_s L_s$, and $\rho(A) = \Pr(L_s \cap A \neq \varnothing)$. For test-route mass w_E,

$$c_n^{\mathrm{comp}} = \sum_E w_E \sum_{A \subseteq E} (-1)^{|A|} [1 - \rho(A)]^n. \tag{11}$$

Condition on route E. Inclusion-exclusion over the event that every primitive in E occurs in S_n gives

$$\Pr(E \subseteq S_n) = \sum_{A \subseteq E} (-1)^{|A|} \Pr(S_n \cap A = \varnothing).$$

Since the L_s are iid and each misses A with probability $1 - \rho(A)$, independence across warm steps gives Equation 11.

Let $d_i = \Pr(i \in E)$ and $\rho_i = \Pr(i \in L_s)$. Missing any required primitive prevents coverage, so

$$\max_i d_i(1-\rho_i)^n \le 1 - c_n^{\text{comp}} \le \sum_i d_i(1-\rho_i)^n. \tag{12}$$

If $\rho_{\min} > 0$ over required primitives and $D = \sum_i d_i = \mathbb{E}|E|$, then $1 - c_n^{\text{comp}} \le D\exp(-n\rho_{\min})$.

The plug-in curve is stable in total variation. Let q, ν denote the true test-route and learned-set distributions, and let $\widehat{q}, \widehat{\nu}$ be estimates. A maximal coupling and union bound give

$$|c_n(q,\nu) - c_n(\widehat{q},\widehat{\nu})| \le \min\{1, \mathrm{TV}(q,\widehat{q}) + n\,\mathrm{TV}(\nu,\widehat{\nu})\}. \tag{13}$$

A.5 Coverage of valid accepted executions

For family E with mass w_E, let $p_{m,E}$ be candidate availability and $s_{m,E}$ conditional joint-valid acceptance (zero when $p_{m,E} = 0$).

Proposition 5 (Coverage decomposition)**.**

$$h_m^+ = \sum_E w_E p_{m,E} s_{m,E}, \tag{14}$$

$$h_c^+ - h_w^+ = \sum_E w_E\big[(p_{c,E} - p_{w,E})s_{w,E} + p_{c,E}(s_{c,E} - s_{w,E})\big]. \tag{15}$$

The two terms separate changes in candidate availability from changes in conditional valid acceptance, including the effect of candidate selection.

Proof of Proposition 5. Condition on $R = E$. Let C_m mean that the frozen selector returns a candidate allowed by the grammar and execution budget, and U_m its joint-valid acceptance. Then $p_{m,E} = \Pr(C_m \mid R = E)$ and, when $p_{m,E} > 0$, $s_{m,E} = \Pr(U_m \mid C_m, R = E)$. Thus $\Pr(C_m \cap U_m \mid R = E) = p_{m,E}s_{m,E}$. If $p_{m,E} = 0$, both sides are zero under the stated convention. Averaging over E proves Equation 14. □

The shorter notation $h_{n,m}^+ = \sum_E w_E p_{m,E} s_{m,E}$ yields Equation 15 from the identity

$$\begin{aligned} &p_{\text{compose},E}s_{\text{compose},E} - p_{\text{whole},E}s_{\text{whole},E} \\ &\quad = (p_{\text{compose},E} - p_{\text{whole},E})s_{\text{whole},E} \\ &\qquad + p_{\text{compose},E}(s_{\text{compose},E} - s_{\text{whole},E}). \end{aligned}$$

For an imperfect deployed gate G, let $V^\dagger = Z^{\text{cv}}$ for the offline contract endpoint and $V^\dagger = Z^{\text{cv}}Y_x(\widetilde{a})$ for the scored joint endpoint. Including candidate availability C_m, define

$$\begin{aligned} h_{n,m} &= \Pr(C_m, G = 1), \\ \phi_{n,m} &= \Pr(C_m, G = 1, V^\dagger = 0), \\ \rho_{n,m} &= \phi_{n,m}/h_{n,m}. \end{aligned}$$

Then $h_{n,m} = h_{n,m}^+ + \phi_{n,m}$, with $\rho_{n,m}$ defined when $h_{n,m} > 0$. For the contract choice, $h_{n,m}^+ = \Pr(C_m, C_x^{\text{cv}} = 1)$; for the reported external choice, $h_{n,m}^+ = \Pr(C_m, J_x^{\text{ext}} = 1)$. W_x^{ans} remains the separately scored external-answer coverage. Question-level risk counts each unique question once. Paired uncertainty uses the true `route_family_id` as the resampling cluster, combining all scheduling blocks from the same family. Here $\phi_{n,m}$ is a population probability, whereas $\widehat{\phi}_{n,m}$ is its empirical frequency. The fixed-checkpoint bound below gives a finite-sample upper confidence bound.

A.6 GROUPED ADMISSION DEFINITION

A separate controller freezes K binder-margin rules before calibration (Angelopoulos et al., 2021). For route group r, accepted fraction $a_r(t)$ and accepted-error fraction $e_r(t)$ define $R(t) = \mathbb{E}[e_r(t)]/\mathbb{E}[a_r(t)]$ for positive acceptance. Appendix D gives the Hoeffding and KL tests, which assume iid route groups from an unchanged target population. The target is route-prediction risk before answer verification. Guarantees under distribution shift require further assumptions (Barber et al., 2023).

A.7 PROGRAM LIBRARY OPTIMIZATION

For primitive set S, route values $v_E \geq 0$, nonnegative integer token costs $c_i \in \mathbb{Z}_{\geq 0}$, and integer token budget $B_p \in \mathbb{Z}_{\geq 0}$, the offline objective is

$$F(S) = \sum_E v_E \mathbf{1}\{E \subseteq S\}, \qquad \sum_{i \in S} c_i \leq B_p. \tag{16}$$

The objective is monotone supermodular: adding a primitive completes at least as many routes when the current library is larger. The general problem is NP-hard by reduction from Densest-k-Subgraph: graph vertices become unit-cost primitives and graph edges become unit-value two-primitive routes.

An exact integer program uses primitive variables x_i and route variables y_E:

$$\begin{aligned} \max \quad & \sum_E v_E y_E \\ \text{s.t.} \quad & y_E \leq x_i && \forall E, i \in E, \\ & y_E \geq 1 - |E| + \sum_{i \in E} x_i && \forall E, \\ & \sum_i c_i x_i \leq B_p, \qquad x_i, y_E \in \{0,1\}. \end{aligned}$$

For a rooted tree with root-to-terminal routes and ancestor-closed selection, let q_v be the value of routes ending at node v. If $DP_v[b]$ is the best value in the subtree when v is selected, initialize q_v for $b \geq c_v$ and $-\infty$ otherwise. Merge child u by max-convolution with $\max(0, DP_u[s])$. Under the integer token assumptions above, the exact optimum is $\max\{0, DP_{\text{root}}[B_p]\}$ and the runtime is $O(|V|B_p^2)$, pseudo-polynomial in the numeric budget B_p.

A.8 COST AND DRIFT

Let $s = hC_A - C_F$, where h is the acceptance probability while the compiled program remains valid (Equation 7). After compilation, cumulative expected gain over M requests is $Ms - C_C$. If $s > 0$, weak break-even and strict profit occur at

$$M_{\geq} = \left\lceil \frac{C_C}{s} \right\rceil, \qquad M_{>} = \left\lfloor \frac{C_C}{s} \right\rfloor + 1.$$

The indices differ when C_C/s is an integer. A request that pays for a failed attempt before compilation incurs setup cost $C_C + C_F$.

Suppose requests and drift are independent Poisson processes with rates $\mu > 0$ and $\lambda > 0$. A cycle starts just after compilation. Drift ends the program's validity; the first subsequent request pays a failed read-only attempt and recompilation before the next cycle. Relative to an always-fallback baseline, the expected number of requests before drift is μ/λ, each saving $hC_A - C_F$, while the cold request loses $C_F + C_C$. Hence

$$\mathbb{E}[\text{Gain}] = \frac{\mu}{\lambda}(hC_A - C_F) - (C_F + C_C). \tag{17}$$

A.9 Deployment at a fixed checkpoint

For K frozen candidates and n independent shadow samples, let $\widehat{h}_k^+$ be the valid-accept frequency and $\widehat{\phi}_k$ the marginal false-accept frequency. With

$$\epsilon_n = \sqrt{\frac{\log(2K/\delta)}{2n}},$$

Hoeffding's inequality and a union bound give simultaneous bounds $\underline{h}_k = \max(0, \widehat{h}_k^+ - \epsilon_n)$ and $\overline{\phi}_k = \min(1, \widehat{\phi}_k + \epsilon_n)$. A fixed-checkpoint policy may require both

$$\underline{h}_k C_A > C_F + C_C/M, \qquad \overline{\phi}_k \le \eta.$$

Continuous online inspection requires a confidence sequence instead.

B Extended experimental protocol

B.1 Synthetic families

The synthetic study combines six topologies (single route, parallel disjoint, shared prefix, shared hub, diamond, and long chain), three route priors (uniform, head-heavy, and tail-heavy), and three drift levels (absent, moderate, and high). With eight seeds, this gives 432 recurrence conditions, each with a shared 16-episode stream.

The complete replay compares ALWAYS-AGENT, fresh retrieval, result cache, whole trace replay, whole plan cache, WHOLE, COMPOSE, and an oracle route. The Qwen router sees the question and a frozen atomic catalog. Deterministic exact replay uses the model output, resolved program library, and served tokenizer. A row records final evidence, valid and accepted fast paths, fallback, artifact tokens, model tokens, latency, and tool calls. Latency is estimated by the fixed cost model; wall-clock time is measured in a separate experiment.

The composition ablation fixes the first four episodes as warm-up and scores the remaining twelve. Its six methods are ALWAYS-AGENT, WHOLE with gate and fallback, COMPOSE with gate and fallback, composition without dependency and precondition checks, composition without a gate, and composition without a fallback. All methods share the same Qwen route replay.

B.2 Model slice

The reduced evaluation selects 108 of 432 families before model execution. Each topology and prior contributes two of eight seeds, and all three drift regimes remain. Qwen3-4B and Qwen3-8B use their local tokenizers. Mistral-7B uses its own tokenizer and bitsandbytes 4-bit loading. Token comparisons are made within each model.

The matched-family sealed evaluation uses Qwen3-4B-Instruct-2507 and Phi-4-mini-instruct in BF16. Both are self-hosted with their frozen native tokenizers and receive the same split, schedule, binder decisions, program conditions, and scoring procedure. Each model-method pair has 1,530 conditions. Public model serving uses vLLM 0.13.0. Frozen scoring and replay use Python 3.9.20, NumPy 1.22.0, SciPy 1.11.4, and scikit-learn 1.5.2; model revisions and tokenizer hashes are recorded in the public manifests.

After the joint unseal, we reran the unchanged v4 split, contract, condition grid, resolver, and scorer with Qwen3-8B-AWQ. The audit contains 240 planner and 720 reader requests, or 960 logical requests in total. This run tests one same-family model replacement, with model size and AWQ quantization changing together.

B.3 2WIKI task contract

The recurrent construction groups 400 development questions into 50 route families with eight entity-disjoint instances. Runtime prompts exclude answer, supporting facts, family ID, question type, and gold route. The relation catalog is a frozen public registry.

The v4 predictor uses word TF-IDF features with one to three grams and a linear SVM. Its residual training and calibration pools exclude candidate examples, normalized templates, and extended-QID components. Training and calibration also have zero template and extended-QID overlap. Calibration selects the largest margin threshold whose simultaneous Clopper-Pearson error bound on the calibration pool is at most 5%. Test-time binding compares the predicted contract with a candidate program by exact canonical route.

For conditioned composition, the library admits a source program only when its question contract, candidate route and packed evidence structure agree. Whole-route retrieval searches programs with the matching contract. Composition assembles the predicted route when all distinct primitives are available and grammar and type checks pass. The experiment uses development data.

Resolver v7 and the execution-record format were fixed after examining the 560 development questions and the prior 240 questions after label release. The verifier checks the predicted question contract, canonical route, admitted evidence set, primitive lineage, packed token count, and terminal operator against frozen values. Accepted paths use deterministic terminal emission without a free-form reader. Comparison routes fall back because the terminal registry lacks a comparison operator.

The legacy Qwen3-8B stress test repeats the earlier public lexical dependency gate, which reads question, program, selected record IDs, and sentence text. It does not use answers, supporting facts, family IDs, or gold routes at runtime. Annotated support is used only for offline scoring.

B.4 Runtime checks and offline scoring

Table 6: Runtime checks and evaluation criteria. The first row governs runtime acceptance; the remaining rows use labels opened after execution.

Layer	Runtime?	Reported event	Meaning
Predicted contract + ProofObject	Yes	Gate acceptance	Frozen binder, versions, typed lineage, current spans/hashes, post-pack budget, and terminal rendering are internally consistent.
Reference route/support/terminal	No	Offline contract validity	The accepted artifact matches the held-out semantic route and reference evidence contract.
Reference answer	No	External correctness	Deterministic emitted answer matches the task answer metric.
All layers	Scored only	Joint validity	Runtime acceptance, offline contract validity, and external correctness all hold.

The trusted computing base is the frozen binder policy and grammar, source/version registry, hash store, read-only resolver and executor, dependency-first packer, verifier, tokenizer, and terminal operator. Model outputs, retrieved text, candidate programs, and the learned program library are untrusted. The guarantee assumes correct trusted components and read-only rejected attempts that preserve fallback-visible state.

B.5 Blind identity-held-out eight-target panel

The frozen panel has 8 blocks and five episodes per block. Episodes 0 and 1 come from one donor route family, episodes 2 and 3 from a second, and episode 4 from the target family. The full grid uses warm sizes 1, 2, and 4 and budgets 128, 256, and 512. The confirmatory cell is episode 4 at warm size 4 and $B = 512$.

Selection reads identities and route structure, but no answers, answer metrics, model outputs, binder decisions, gate outcomes, or method effects. All 40 example IDs are excluded from the question-only binder fit. Relative to the earlier v1, v4, v2, and v3 panels, the eight targets have zero overlap in

example ID, normalized-template cluster, and extended-QID component. The residual fit retains other rows from the same route families and, where available, the same template clusters. Every target family retains residual supervision. The study therefore holds out target identities while allowing route-family and template overlap.

All eligible families supported by the runtime registry, including the eight selected families, had appeared in earlier oracle evaluations. Target identities are selected without using method outcomes.

The public runner uses the closed v8 pattern resolver and the retained v4 ProofObject checks. Each model makes 160 logical calls, comprising 40 shared planner and 120 shared fallback-reader calls. WHOLE and COMPOSE reuse these calls over 192 conditions each; the equal-budget fresh-agent baseline reuses the fallback cache. Primitive-union and oracle diagnostics run only after unseal and make no model call. Competitive context tokens must not exceed the model-specific budget; prompt overhead is reported separately.

The primary scored joint endpoint is `runtime_end_to_end_valid_accepted`, which requires runtime gate acceptance, offline reference-contract validity, and external answer correctness for the target question. The contrast is COMPOSE minus WHOLE at the route-family level, tested by an exact one-sided sign flip at $\alpha = 0.05$. Qwen3-4B is primary. Qwen3-8B is tested only if the 4B gate passes, and the cross-model claim requires both tests. Each public manifest is committed before one joint marker opens the private rows. The scorer makes no model call and permits no post-unseal tuning.

After unsealing, a deterministic diagnostic replays four nested interventions at the primary configuration without model calls: observed execution; gold route binding with the original resolver and proof checks; an offline resolver that admits only frozen-planner source routes whose strict 512-token pack covers all official support records; and a library of all 32 reference source programs. It exposes the gold terminal answer only after checking path feasibility. A separate identity-terminal check reads the last official evidence object. These post-unseal interventions use reference labels to estimate descriptive upper bounds; the confirmatory runtime decisions remain fixed.

B.6 HOTPOTQA STRICT PACKING

The HotpotQA diagnostic uses a deterministic 100-question held-out sample. Question-conditioned BM25, BGE, and RRF retrieve $K \in \{2, 4, 8\}$ whole sentence records. Retrieval records are packed first and remaining base records fill the same evidence-record budget. The primary diagnostic is Ordinary, $B = 512$, RRF at $K = 8$ against no retrieval. The paired bootstrap resamples questions within frozen question-type and difficulty strata.

The matched random-refill analysis compares content selection at nearly equal input sizes. It replays the primary cell with 20 content-blind fills per question. The selector draws whole records from the same frozen archive using only stable hashes and tokenizer lengths, while matching the RRF cell's kept-retrieval and newly added record counts. Every random context is at or below its per-question RRF target, with a gap of at most 8 tokenizer tokens; the rendered API prompt preserves the same gap. Random seeds are averaged within question before the same stratified paired bootstrap.

B.7 RUN STATUS

Table 7 lists the runs used in this study.

Table 7: Experimental scope and status.

Run	Scope	Use
Synthetic full v2	432 families, 16 episodes, Qwen3-4B	Main result
Composition ablation v1	124,416 paired rows	Main result
Review-repair contrast v1	432 paired families, 10,000 stratified bootstrap resamples, no model calls	Primary comparator and risk audit
Multimodel reduced v1	Three models, 108 families each	Main result
Independent contract	80,511 train, 7,278 calibration, 400 test	Diagnostic
Conditioned composition	400-question development slice	Development
Legacy 8B external	Public structural gate	Stress test
Hotpot strict budget	100 questions, fixed cap	Packer diagnostic
Hotpot matched random refill	100 questions, 20 fills each	Content-selection control
2Wiki opened development	800 examples: original 560 plus prior 240	Resolver v7 freeze
2Wiki matched-family confirmation v4	240 questions, 30 blocks, 13 true families, two models	Prospective confirmation
Post-hoc 8B transfer	240 planner plus 720 reader requests; unchanged v4 grid	Same-family audit
Blind identity-held-out panel	8 targets, five episodes per block, Qwen3-4B then Qwen3-8B	Prospective negative result
Blind post-unseal stage diagnostic	8 targets, 32 donors, four nested interventions, no model calls	Descriptive failure localization only
Authorization stress v1	13 clean ProofObjects, 169 one-factor mutants, seven profiles, no model calls	Post-unseal mechanism audit
Structured binder development v1	13 held-out route combinations, 2,198 questions, no model calls	Opened-development capability diagnostic
Matched guard and resolver replay	13 base proofs, 78 benign, 234 single faults, 715 paired faults, 9 coherent substitutions	Post-unseal robustness and cost audit
Route-shift calibration	Five partitions of 37 fitting, 13 calibration, and 13 test combinations; fixed 12-checkpoint grid	Opened-development statistical admission diagnostic
No-drift 24-seed calibration	Seeds 0 to 23; 4,608 new route rows	Theory check

The final20 dataset contains 6,270 sentence records, 320 reference support groups, and 160 examples from 20 compositional families. Together with the earlier 400-question slice, it forms the opened 560-question development set. The 240 questions from the prior confirmation joined this development pool after their one-shot unseal. Resolver v7 was frozen from these opened data. The training audit removes 3,658 examples that touch final20 extended QIDs, leaving 147,416 training and 16,380 calibration examples. Each whole-route and compositional development method has 1,020 conditions across three budgets and three warm sizes.

The v4 confirmation contains 240 questions in 30 eight-episode recurrent blocks. Episode 0 supplies 30 warm starts; the remaining 210 unique questions form the evaluated set. The blocks represent 13 true route families. Seven families are reused across blocks, the maximum multiplicity is four blocks, and 12 families overlap the prior opened confirmation. Cumulative exclusion gives zero example, normalized-template, and extended-QID overlap. Primary external answer risk, conditional on fast-path acceptance, is collapsed by unique question. The effect bootstrap resamples true route families, combining blocks from the same family.

The split, question-only binder inputs, recurrent schedule, runtime sentence material, resolver v7, verifier, model and tokenizer fingerprints, metrics, and public execution code are committed before model calls. Qwen3-4B and Phi-4-mini use the same frozen split, schedule, binder, and scorer. Each model-method pair produces 1,530 conditions. A joint unseal begins after both public execution manifests are committed; the scorer then opens the 240 gold rows once and makes no model call.

The identity-held-out panel freeze precedes all new model calls. It commits the 40-example panel, exact-ID binder holdout, runtime sources, two model profiles, condition grid, primary endpoint, family-level test, and fixed testing order. Both model manifests must exist before the joint scorer opens private rows.

The review-repair builder reads only the frozen common-suffix family summary and sealed Qwen summary. It recomputes all WHOLE–ALWAYS-AGENT, COMPOSE–WHOLE, and COMPOSE–ALWAYS-AGENT contrasts at $B = 256, 512$, plus the unique-question layer risks. Its manifest binds both input summaries, their source manifests, the builder source, CSV output, JSON summary, and LaTeX macros by SHA-256. Repeated invocations are byte-identical and make no model call.

The blind-stage diagnostic manifest binds the frozen joint-run inputs, builder source, target- and donor-stage CSV files, JSON summary and LaTeX macros. It records that the diagnostic uses reference labels after unsealing and makes zero model calls. Repeated invocations are byte-identical. The family-effect figure manifest binds the sealed condition scores, builder source, anonymized F1–F13 data and PDF/PNG outputs for the frozen analysis.

The no-drift extension targets family seeds 8 through 23. Its resumable route ledger contains all 4,608 expected rows. Exact replay contains 27,648 rows, and the balanced 24-seed calibration contains 41,472 rows.

B.8 POST-UNSEAL AUTHORIZATION STRESS PROTOCOL

The stress builder reads the frozen Qwen v4 Stage-C candidate ledger, Stage-D ProofObjects, unsealed scores, runtime material, and fallback ledger. It selects the primary warm-4, $B = 512$ cell and deduplicates by serialized proof hash. This leaves 13 ProofObjects from four true route families. Only Full-accepted candidates have complete saved execution records, so all clean profiles use the same accepted objects. The panel measures fault detection among these objects, not the effect of checks on natural-task coverage.

Each base receives exactly one mutation of each kind: route fingerprint, task-contract hash, primitive order, deleted ancestor primitive, source-span offset, current record content, tokenizer version, executor version, parent-output hash, parent lineage, a critical record moved outside the pack, measured tokens above budget, and a same-typed terminal value. Mutations are constructed on a copy, and unrelated derived fields are held fixed wherever possible so the earliest intended layer is identifiable. Full delegates to the frozen historical verifier. The counterfactual profiles ignore rejection reasons after their named layer; LIGHT-GUARD implements its separately stated non-nested policy. Every evaluation recomputes a canonical hash of the associated frozen fallback row and verifies equality to the pre-replay hash. This checks that rejection leaves the recorded fallback state unchanged.

B.9 STRUCTURED BINDER DEVELOPMENT PROTOCOL

The structured binder uses the same question-only TF–IDF features as the flat development binder but fits separate linear-SVM heads for the first and second relations of public ordered two-hop routes. Grammar decoding accepts only registered relation pairs and uses the minimum slot margin as route confidence. From opened binder-development data, exact route combinations with at least 50 examples are capped at 200 examples each. A deterministic hash split withholds 13 of 63 combinations while retaining every component relation in each slot of the training set. The remaining combinations supply 6,029 train, 1,295 calibration, and 1,302 seen-route test questions; the held-out-combination test has 2,198 questions.

The operating threshold uses seen-route calibration and the diagnostic builder's simultaneous one-sided risk rule. Held-out labels are reserved for evaluation, including descriptive risk–coverage curves. This development split allows overlap in families, templates, QIDs, entities, and prior oracle evaluations; v4 runtime decisions remain fixed.

B.10 MATCHED GUARD AND ROUTE-CALIBRATION EXTENSIONS

The matched-guard builder binds Stage C and Stage D to the frozen execution manifest. Contract and candidate hashes come from Stage C before the proof; source version comes from the runtime-material digest, executor/tokenizer versions from the execution manifest, and policies from the frozen source. Expected versions are therefore independent of the mutated record. The original 13 mutations are extended with changes to all executor versions, all source versions, candidate policy, candidate version, and the cited record hash. Pairing uses the 11 original mutations that preserve the two-record

shape, giving 55 unordered pairs; order reversal and ancestor deletion are excluded to avoid ill-defined overlapping transformations. Both members are applied in the fixed recorded order.

The statistical extension selects route combinations by identity, without using prediction outcomes. Fitting, shifted calibration and test combinations are disjoint within each partition, and fitting must cover every relation slot. These constraints ensure disjoint route combinations within a partition, but not independent route sampling. Each controller allocates its own familywise error budget over 12 thresholds and five partitions; selecting among controllers requires an additional correction. The route rule measures exact-route error and binder acceptance. A guarantee for the complete runtime policy would require separate samples labeled for that policy's final endpoint, with the program library and selector also fixed.

The five fixed partitions each contain 37 model-fitting, 13 shift-calibration, and 13 test route combinations. Every held-out slot relation occurs in fitting; example IDs are disjoint within each partition. Separate examples from fitting routes provide seen-route calibration. The four controllers are the historical seen-question threshold, a fixed-grid seen-question control, question-pooled Clopper–Pearson calibration on shifted routes, and the route-group rule in Equation 18. The grid contains 12 checkpoints; each controller allocates $\delta = .05$ over its checkpoints and the five partitions. Final test labels never select a threshold. The partitions overlap, so their ranges describe sensitivity. The development data allow entity and template overlap and lack an established iid route-sampling mechanism. The historical sealed question-level risk bounds retain their iid-question interpretation; the route-calibration labels measure exact route prediction rather than final answer correctness.

C Additional results

C.1 Numerical checks of the theory

The numerical validation uses 108,000 calibration streams and 108,000 held-out streams. The maximum whole-route absolute error is 0.01245 and the maximum compositional error is 0.01583. There are no coverage-dominance or gate-bound violations. Across 42 drift-cost cells with 100,000 cycles each, Equation 17 predicts every observed gain sign, while the maximum magnitude error is 0.55029. The predicted signs agree with the simulations; gain magnitudes show finite-sample error. Exhaustive search, the integer program, and tree dynamic programming agree on all 64 small program libraries; the integer program is optimal on all 64 larger instances.

The no-drift warm-up analysis uses 24 family seeds. Table 8 reports family-averaged point MAE; the prespecified tolerance is 0.05.

Table 8: Point MAE for the 24-seed no-drift warm-law calibration.

Budget	Whole	Compose	Result
256	0.03853	0.03273	Both point estimates pass
512	0.05192	0.04620	Compose point estimate passes

Three of four point estimates meet the 0.05 threshold. The upper confidence limit exceeds 0.05 in every cell, so none passes the interval-based criterion.

C.2 AUTHORIZATION COMPONENT REPLAY

Table 9: Nested gate profiles and the non-nested LIGHT-GUARD. Every profile accepts 13/13 clean persisted ProofObjects, of which 12/13 are strict joint-valid. Mutant false acceptance is over the same 169 one-factor mutants.

Profile	Clean accept	Clean joint-valid	Mutant false accept
G0	13/13	12/13	143/169 (84.62%)
G1	13/13	12/13	117/169 (69.23%)
G2	13/13	12/13	65/169 (38.46%)
G3	13/13	12/13	39/169 (23.08%)
G4	13/13	12/13	13/169 (7.69%)
Full	13/13	12/13	0/169 (0.00%)
LIGHT-GUARD	13/13	12/13	117/169 (69.23%)

Table 10: Per-mutation outcome for LIGHT-GUARD. Full rejects 13/13 instances of every row. "Accepted" means the simplified guard misses the deliberate corruption.

One-factor mutation	First Full layer	Light accepted
Route fingerprint	G0	0/13
Task-contract hash	G0	0/13
Primitive order	G1	13/13
Delete ancestor primitive	G1	13/13
Source-span offset	G2	13/13
Current record content	G2	0/13
Tokenizer version	G2	13/13
Executor version	G2	13/13
Parent-output hash	G3	13/13
Parent lineage	G3	13/13
Critical record outside pack	G4	13/13
Measured tokens above budget	G4	0/13
Same-typed terminal value	Full	13/13

The Full verifier's current-host replay is 0.583/0.885 ms p50/p95. Mean/max canonical JSON size is 3117/ 3232 bytes. All 1,274 profile evaluations leave the associated fallback-row hash unchanged. The clean panel contains accepted Date-terminal executions from four families. Reported rates describe these records and their constructed mutations; natural failure frequencies are unmeasured.

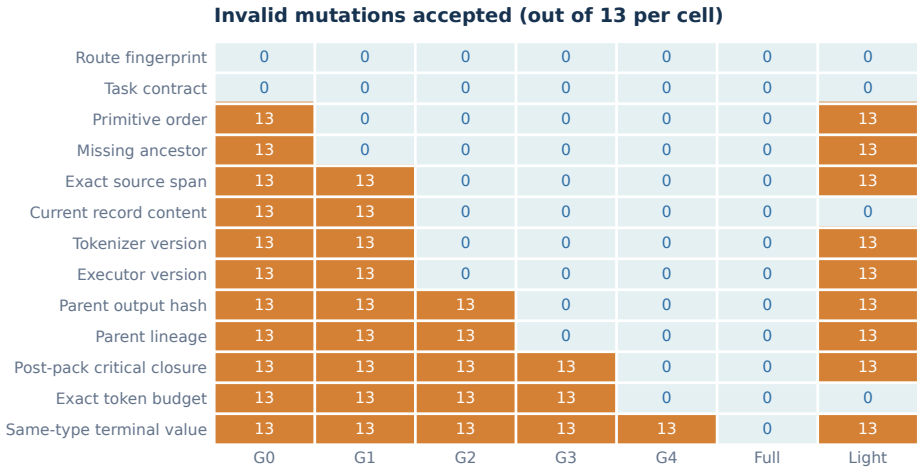


Figure 2: The original mutation suite, resolved by fault class. Each cell is the number accepted among 13 deliberately invalid variants. Dark cells show missed faults; no cell estimates natural prevalence. The nested profiles and historical Light policy are distinct from the stronger direct matched guard.

C.3 Binder accuracy on held-out route combinations

Table 11: Opened-development binder results. The operating thresholds are frozen on seen-route calibration. Flat unseen accuracy is structural because an absent class cannot be emitted.

Split	Binder	Questions	Exact	Selective coverage	Accepted error
Seen route	Flat	1,302	100.00%	100.00%	0.00%
Seen route	Structured	1,302	100.00%	99.85%	0.00%
Held-out combination	Flat	2,198	0.00%	80.30%	100.00%
Held-out combination	Structured	2,198	73.43%	94.45%	23.12%

The structured model's equal-route held-out accuracy is 69.19%; four of 13 route combinations have zero exact accuracy and eight have 100%. Its 480 errors among 2,076 accepted held-out predictions give a one-sided 95% upper bound of 24.70%, far above the 5% seen-route target. Figure 3 shows the frozen operating point and a post-hoc curve computed with held-out labels. The frozen threshold is the runtime operating point; its seen-route risk guarantee does not extend to held-out combinations.

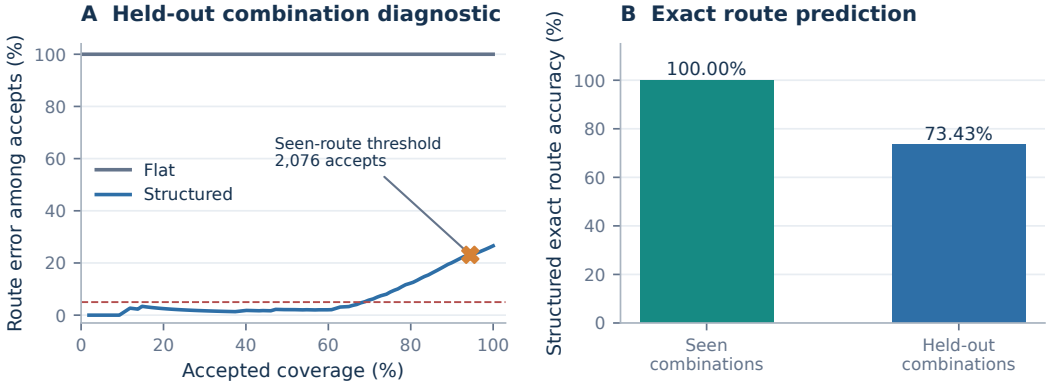


Figure 3: Structured-binder risk–coverage diagnostic. The X marks the seen-calibration threshold; the remaining held-out-label sweep is descriptive and cannot be used for threshold selection.

C.4 WALL-CLOCK MEASUREMENTS AND AN ACCEPTED EXAMPLE

Table 12: Available real timers. Rows were not synchronized and must not be summed into an end-to-end latency.

Stage	p50 (ms)	p95 (ms)	Measurement
Whole candidate lookup	0.008	0.296	Historical in-process Stage C
Composition selection	0.354	0.821	Historical in-process Stage C
Full proof verifier	0.583	0.885	Current-host deterministic replay
Fallback model call	284.791	748.768	Historical frozen reader ledger

For the Mathilde case, the predicted and selected route is `composer → date of death`; the runtime first-hop citation is "The music was composed by Gérard Jouannest," and the terminal citation contains "16 May 2018." The route and answer are valid and the verifier accepts, but strict reference path and terminal checks fail; citation recall and precision are both 0.5. A different sentence in the same article explicitly says that the song's music is by Gérard Jouannest. That sentence supports an alternative derivation, while the confirmatory result retains the preregistered strict score.

C.5 VERIFIER ROBUSTNESS AND RESOLVER REPLAY

The extended panel retains all 13 historical accepted proofs, including the reference-invalid Mathilde case. Direct Full, Light, and replay policies accept all 78 benign transformations. These include JSON-key and catalog order, unobserved metadata, edits or appends outside the admitted record set, and whitespace expansion of an exact span. An unadmitted-catalog edit tests the verifier's proof-relative interface: a production loader must still authenticate any changed global corpus snapshot.

Figure 4 reports single-fault acceptance after selected groups of rejection reasons are masked. The comparison tests whether the remaining Full checks detect each fault. It uses masked rejection reasons rather than separately implemented verifiers, so it does not estimate their cost. The local decision log retains every rejection reason. Paired-fault omissions still admit only 13/715 cases for each of contract, versions, lineage, and packing; all other masks admit none.

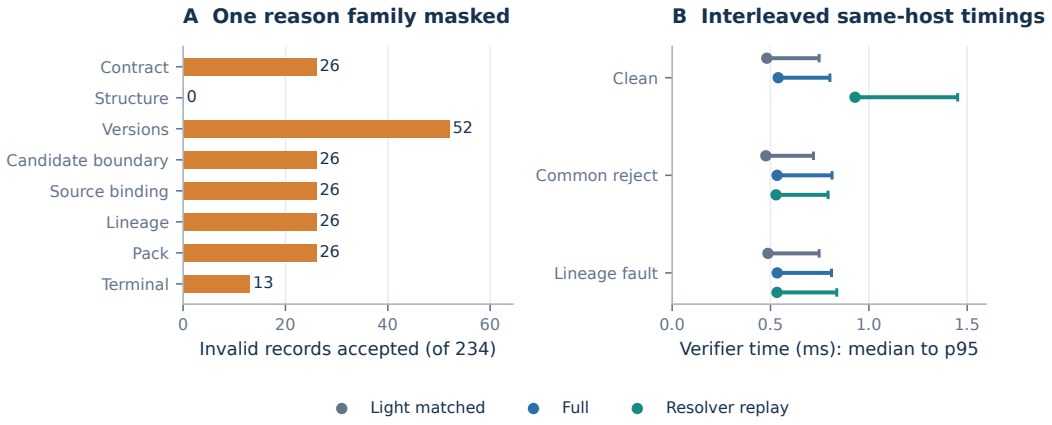


Figure 4: Left: single-fault accepts when one Full rejection-reason family is masked; remaining checks may still contain its target fault. Right: matched CPU replay, where points are medians and line endpoints are p95, not confidence intervals. Each policy/condition has 520 measurements after warm-up. The lineage fault is accepted by Light and rejected by the other two policies.

The 65 matched observation collisions comprise five transformations on each base: primitive order, exact span offset, parent output hash, parent lineage, and same-type terminal value. The observation projection contains all fields used by the direct lightweight decision. The output proof hash records

the input identity but is excluded from the decision features. The failures therefore characterize this lightweight policy.

For the coherent-substitution probe, nine terminal source records contain another recognizable date. The builder selects it by a fixed rule, updates the typed terminal output and exact span, and keeps the source bytes and independent pins fixed. All nine outputs have zero exact match under the original answer scorer. Full and Light accept them, while fresh resolver replay rejects all nine. The test simulates incorrect execution records. Replay detects the changed outputs by recomputation, but still shares the original resolver's relation patterns and any errors in those patterns.

C.6 Five-partition route calibration

Table 13 reports every partition, with seeds 20260906 through 20260910 denoted S1–S5. For each route, we divide accepted errors and acceptances by its total question count. Conditional error is the ratio of these averages across 13 test routes, giving each route equal weight. The fixed-grid seen-question control accepts every test question in all five partitions; it is retained in the machine-readable results together with the uncontrolled grammar reference. The route controller accepts no question in any partition.

Table 13: Equal-route percentages on held-out combinations. Coverage/error use the same test routes in each row. A dash means zero acceptance and undefined conditional error. These are overlapping development partitions.

	Historical seen CP		Shifted-question CP	
Partition	Coverage	Error	Coverage	Error
S1	87.65	13.90	24.78	0.00
S2	99.11	37.44	0.00	–
S3	95.74	22.16	72.07	10.88
S4	97.85	21.50	32.13	0.00
S5	100.00	38.61	0.00	–

For S3 the shifted-question threshold is 0.75. Calibration has 30 errors among 1,627 accepts, giving 1.84% empirical error and a multiplicity-adjusted CP upper bound of 3.14%. The new-route test has 154 errors among 1,759 accepts (8.75% pooled error); the equal-route counterpart is 10.88%. The two error-bearing routes contribute 40/40 and 114/114 wrong accepted predictions. Question counts therefore obscure failures concentrated within route groups. The route rule uses $K = 12$, $S = 5$, $\delta = .05$, and $m = 13$. Its correction is 0.5222. The value 1,419 follows from $m \geq \log(KS/\delta)/(2\alpha^2)$ for zero error and full coverage under this sufficient condition. The value is specific to this sufficient condition.

C.7 SYNTHETIC ABLATIONS

Table 14: Common-suffix composition ablation. Completion, false accepts, and fallback are percentages. The no-fallback arm also stops adding programs during suffix episodes; it is not a fixed-trajectory fallback intervention.

Method	B	Complete	False accept	Fallback
Agent	256	53.67	0.00	100.00
Whole gate and fallback	256	54.82	0.00	58.53
Compose gate and fallback	256	54.96	0.00	55.98
Compose no dependency/precondition checks	256	54.96	0.00	55.98
Compose without gate	256	46.80	12.85	47.84
Compose without fallback	256	31.48	0.00	0.00
Agent	512	60.57	0.00	100.00
Whole gate and fallback	512	61.77	0.00	53.14
Compose gate and fallback	512	61.90	0.00	50.46
Compose no dependency/precondition checks	512	61.90	0.00	50.46
Compose without gate	512	52.43	15.59	40.24
Compose without fallback	512	35.46	0.00	0.00

Removing dependency and precondition checks gives the same results as Full: the frozen selector records zero additional unchecked candidate attempts. Because candidate selection is unchanged, this ablation does not isolate the checks' effect. The execution-record mutations test them directly.

Table 15 reports the primary mechanism contrast and the two system-level contrasts from the same 432 paired families. The prespecified COMPOSE–WHOLE comparison isolates candidate assembly. Comparisons against ALWAYS-AGENT include the complete reuse pipeline.

Table 15: Paired family contrasts. Completion and valid fast-path effects are percentage points; token effects are relative percent changes. Brackets are 95% stratified-bootstrap intervals.

B	Contrast	Completion	Valid fast path	Suffix tokens
256	Whole − Agent	1.16 [0.87, 1.45]	41.47 [40.49, 42.44]	-24.27 [-24.84, -23.70]
256	Compose − Whole	0.14 [0.06, 0.23]	2.55 [2.18, 2.93]	-2.01 [-2.31, -1.72]
256	Compose − Agent	1.29 [1.00, 1.60]	44.02 [43.02, 45.06]	-25.79 [-26.40, -25.19]
512	Whole − Agent	1.20 [0.91, 1.49]	46.86 [45.81, 47.92]	-27.60 [-28.22, -26.98]
512	Compose − Whole	0.14 [0.06, 0.23]	2.68 [2.31, 3.07]	-2.22 [-2.54, -1.92]
512	Compose − Agent	1.33 [1.02, 1.64]	49.54 [48.40, 50.66]	-29.21 [-29.88, -28.53]

C.8 TOPOLOGY AND MODEL SLICES

Table 16: COMPOSE minus WHOLE valid-fast-path coverage over episodes 5–16 at $B = 256$, with a decomposition into candidate availability and conditional validity. Intervals and p values are for the total and use paired family resampling and Holm correction.

Topology	Δ (pp)	Availability	Certifiability	95% CI	Holm p
Shared hub	8.22	6.82	1.40	[6.71, 9.84]	.0018
Shared prefix	7.06	4.67	2.39	[5.56, 8.68]	.0018
Diamond	0.00	0.00	0.00	[0.00, 0.00]	1.0
Long chain	0.00	0.00	0.00	[0.00, 0.00]	1.0
Parallel disjoint	0.00	0.00	0.00	[0.00, 0.00]	1.0
Single route	0.00	0.00	0.00	[0.00, 0.00]	1.0

Table 17: Controlled system-level directional replication at $B = 256$. Completion is COMPOSE minus ALWAYS-AGENT; token reductions are within model. The COMPOSE–WHOLE final-completion effect is zero for every row.

Model	Route exact	Completion (pp)	95% CI	Token reduction
Qwen3-4B	72.80	1.10	[0.69, 1.50]	23.86
Qwen3-8B-AWQ	71.88	0.93	[0.58, 1.27]	18.38
Mistral-7B 4-bit	32.70	0.69	[0.35, 1.04]	10.55

C.9 INDEPENDENT CONTRACT AND CONDITIONED COMPOSITION

The 400-question results in Tables 18 and 19 are development evidence for contract binding and conditioned route composition.

Table 18: Independent task-contract binding on 400 2Wiki questions.

Metric	Value
Raw route exact	96.00%
Selective coverage	88.75%
Selective route errors	1/355
Selective error, one-sided 95% upper	1.329%
Combined-gate accepts	145/2,550
Route false accepts	0/2,550
Evidence false accepts	12/2,550
Hybrid complete evidence	25.18%
Fresh complete evidence	24.63%

Table 19: Conditioned development composition at $B = 512$.

Warm	Gate accept	Complete	Evidence FA	Unseen whole	Cross trace
1	24.00%	58.57%	2.57%	53	45
2	32.67%	60.33%	4.00%	45	36
4	35.00%	60.50%	4.00%	24	18

C.10 RESOLVER COVERAGE ON DEVELOPMENT DATA

Resolver v7 expands executable coverage using opened title-transition rows. Table 20 compares the resolver versions on 560 opened questions, of which 464 are eligible for transition execution.

Table 20: Executable coverage across resolver versions on development data.

Resolver	Transition resolved	Gate accepted	Wrong accepted
v4	47	6	0
v5	76	7	0
v6	76	39	0
v7	130	44	0
v7 minus v6	+54	+5	0

The opened route factorial in Table 21 separates predicted-route binding from a route oracle and perturbs the oracle resolver.

Table 21: Opened 240-question route factorial. Whole and Compose cells report assemblable / accepted conditions.

Route / resolver	Source promotions	Whole	Compose
Predicted / v7	10	122/ 9	150/ 14
Oracle / v6	13	143/ 11	180/ 18
Oracle / v7	14	150/ 13	180/ 18

Wrong accepts are 0, and strict-budget violations are 0. Resolver v7 was developed to expand executable coverage on previously examined data. The sealed evaluation below estimates coverage and validity.

C.11 SEALED V4 CONFIRMATION

The public schedule contains 240 questions in 30 opaque recurrent blocks. The 30 episode-0 questions populate the program library; the remaining 210 unique questions form the evaluated set for coverage. Answer risk uses the accepted-question counts in Table 22. After unseal, the blocks map to 13 true route families. Of these, 7 appear in more than one block, the largest spans 4 blocks, and 12 overlap the earlier sealed material. The schedule hides family labels during execution while retaining previously seen families.

Table 22: Sealed Qwen3-4B funnel at the unique-question unit. The risk column is the one-sided 95% answer-error upper bound among accepted questions.

Method	Assemblable	Accepted	Joint valid	Answer errors	Risk upper
Whole	71	15	14	0	18.10%
Compose	87	19	18	0	14.59%

Table 23: Sealed Qwen3-4B funnel over 1,530 conditions per method.

Method	Assemblable	Accepted	Joint valid	Fallback
Whole	230	32	31	1,498
Compose	246	36	35	1,494

The 4 composition-only joint-valid questions belong to 1 true family. Table 24 reports the family-balanced estimates.

Table 24: Family-balanced Compose-minus-Whole effects for sealed Qwen3-4B in percentage points. Each effect equally averages the within-family condition-mean contrasts over the 13 true families.

Endpoint	Effect (pp)	95% CI (pp)	Higher	Tie	Lower
Assemblable	0.60	[0.00, 1.81]	1	12	0
Joint valid	0.15	[0.00, 0.45]	1	12	0

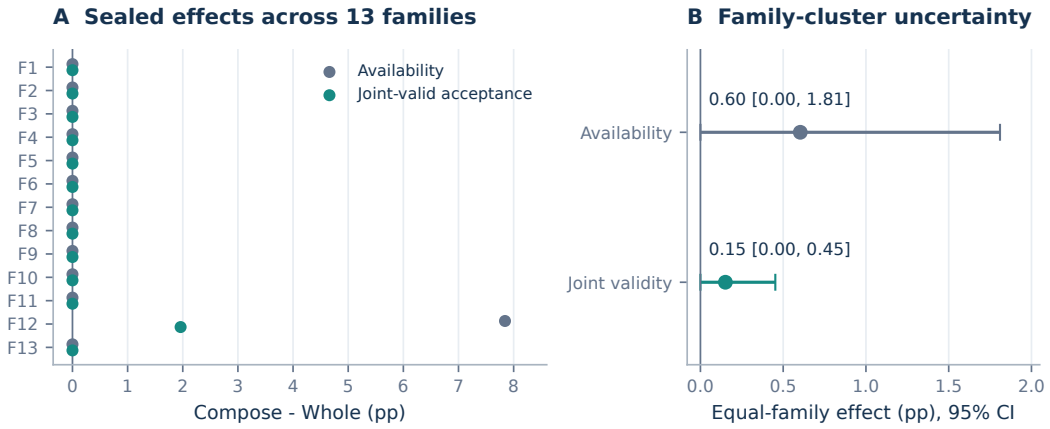


Figure 5: Family-level sealed Qwen3-4B COMPOSE–WHOLE effects. Families are anonymized in a fixed hash-bound order. Points on the left retain every family; the right panel reports equal-family means and their frozen 95% family-bootstrap intervals. Only F12 is positive; the other 12 families tie.

Each method has one runtime-accepted question that is not joint valid. Its external answer is correct and its semantic route is valid, but the offline path and terminal checks fail because the ProofObject cites only half of the reference support chain. Runtime verification checks internal dependencies, current spans and hashes, and predicted-contract consistency. Offline scoring additionally compares the held-out reference support set. This produces 15 accepted versus 14 joint-valid questions for WHOLE and 19 versus 18 joint-valid questions for COMPOSE.

Table 25: Terminal-type support in sealed Qwen3-4B. Counts for each type use unique questions; funnel entries count repeated conditions across the frozen budget–warm grid.

		Whole			Compose		
Type	Questions	Asm.	Accept	Joint	Asm.	Accept	Joint
Date	133	230	32	31	246	36	35
Boolean	42	0	0	0	0	0	0
EntityID	35	0	0	0	0	0	0

Table 26: Sealed Phi-4-mini funnel. Question columns use 210 evaluated questions; fallback is counted over 1,530 conditions per method.

Method	Assemblable	Accepted	Joint valid	Fallback
Whole	0	0	0	1,530
Compose	0	0	0	1,530

The Phi public planner ledger contains 56 valid, nonempty candidate programs among 240 questions. None has an `aligned_valid` binder-topology status in the stage-A rows, so no candidate reaches promotion. The counts come from the frozen public execution records.

C.12 BLIND IDENTITY-HELD-OUT EIGHT-TARGET PANEL

The frozen primary cell contains one target question from each of 8 route families. Results are paired within family and model; absolute token counts are not compared across tokenizers. Table 27 reports the prespecified endpoint. The 4B gate is evaluated first. Because it fails, the 8B row is descriptive rather than a second confirmatory test.

Table 27: Identity-held-out eight-target primary result at warm size 4 and $B = 512$. Valid counts are over 8 target families.

Model	Whole	Compose	Mean Δ	One-sided p	Decision
Qwen3-4B	0/8	0/8	0.000000	1.000000	Primary gate fails
Qwen3-8B-AWQ	0/8	0/8	0.000000	not tested	8 ties; descriptive

All 8 family differences are zero for each model. The 4B and 8B runs have 0 and 0 strict-budget violations. WHOLE, COMPOSE, the equal-budget fresh agent, and primitive union all have target answer EM 0.0%/ 0.0%/ 0.0%/ 0.0% for 4B and 0.0%/ 0.0%/ 0.0%/ 0.0% for 8B. An unconstrained upper bound obtained by copying reference answers without model calls reaches 100.0% and 100.0%. This unconstrained calculation ignores path feasibility; runtime acceptance uses the original evidence checks. Table 29 reports the stricter diagnostic that requires a feasible path.

Table 28 separates the fresh fallback diagnostics from reuse-candidate checks.

Table 28: Target diagnostics for the blind 8-family panel. Program, artifact, answer, and correctness belong to the fresh fallback path; binder match and gate acceptance belong to the reuse-candidate path. "Binder match" means a selected library candidate matches the frozen predicted contract; it is not binder-prediction accuracy against the post-unseal gold route.

	Fresh fallback path				Reuse-candidate path	
Model	Program	Artifact	Answer	Correct	Binder match	Gate accept
Qwen3-4B	7/8	7/8	0/8	0/8	0/8	0/8
Qwen3-8B-AWQ	5/8	5/8	4/8	0/8	0/8	0/8

The 4B fresh fallback reader abstains on all 8 questions. The 4 nonempty 8B answers are all wrong. Neither model selects a program from the library that matches the frozen predicted contract, so no target execution reaches gate acceptance. The stage diagnostic below separates binder errors, source-program availability and candidate selection.

For every model–method pair, 4 of eight targets first fail the binder gate and the remaining 4 first fail route or primitive coverage in the admitted program library. Table 29 then intervenes on progressively more of the pipeline.

Table 29: Post-unseal blind-stage cascade at the frozen primary cell. Donor availability reports aligned/promoted programs under gold binding, the offline library size under packed support, and all donor programs under full union. Candidate/path counts and terminal EM are over eight targets. All rows use private reference information and provide descriptive upper bounds only.

Model	Intervention	Donor availability	Whole cand./path	Compose cand./path	Compose terminal EM
4B	Gold route binding	24/0	0/0	0/0	–
4B	Packed gold support	19	0/0	7/3	3
4B	Full gold donor union	32	0/0	8/3	3
8B	Gold route binding	20/1	0/0	0/0	–
8B	Packed gold support	17	0/0	3/1	1
8B	Full gold donor union	32	0/0	8/3	3

Gold binding alone aligns 24/32 4B and 20/32 8B donors but promotes only 0 and 1, respectively, so every target still has zero candidates. The packed-support intervention creates offline libraries of 19 and 17 programs; Compose then finds candidates for 7/8 and 3/8 targets, but only 3/8 and 1/8 paths are strictly feasible. Supplying all reference donor programs makes every Compose target structurally assemblable while only 3/8 paths remain feasible for each model; Whole remains at zero. The interventions identify separate bottlenecks in binding, donor planning and execution, program admission and availability, and target packing.

C.13 POST-HOC QWEN3-8B TRANSFER AUDIT

This post-hoc audit runs Qwen3-8B-AWQ on the v4 protocol after the joint unseal, with the split, contract, resolver, condition grid and scorer fixed. It tests a model replacement within the Qwen family.

Table 30: Structural funnel under same-family model replacement. Each entry is Whole / Compose over the shared v4 condition grid.

Model	Assemblable	Gate accepted	Joint valid	Joint-valid questions
Qwen3-4B	230/246	32/36	31/35	14/18
Qwen3-8B-AWQ	230/246	32/36	31/35	14/18

The 8B structural funnel matches the 4B funnel. Its family-balanced joint-valid effect is 0.15 percentage points (95% CI [0.00, 0.45]); the 8B-minus-4B interaction is 0.00 percentage points (95% CI [0.00, 0.00]). Only 1 of 13 families improves under composition. Fallback behavior changes with the model: Whole / Compose hybrid F1 is 18.81%/18.99% for 8B, compared with 20.08%/20.14% for 4B.

C.14 LEGACY QWEN3-8B STRESS TEST

Table 31: Qwen3-8B under the legacy public structural gate at $B = 512$. Token changes compare the hybrid with Fresh.

Warm	Reader EM Δ	95% CI	Test token Δ	Session token Δ
1	7.43	[4.29, 10.86]	-14.40%	-4.43%
2	7.67	[4.00, 11.67]	-13.42%	$+9.82\%$
4	10.50	[6.00, 15.50]	-15.14%	$+54.99\%$

The public gate accepts 13.57% of conditions, with a 3.73% marginal evidence false-accept rate and six pointwise dominance violations. The gate can therefore accept invalid evidence despite improvements in reader scores and cost.

C.15 HOTPOTQA STRICT-BUDGET DIAGNOSTIC

The original retrieval grid contains 4,000 evaluated cells, all within the evidence-record cap. The matched analysis adds 2,200 contemporaneous reader calls. The 512-token cap applies to serialized evidence records; API input also includes the fixed instruction and question wrapper. Intervals use 10,000 stratified bootstrap samples at the question unit.

RRF versus no retrieval gives an overall 54.8-point contrast (95% CI [45.6, 63.6]) because the two conditions differ in both context content and length. Content-blind random refill reaches 12.0% F1, versus 1.0% without retrieval. At near-matched prompt length, RRF reaches 55.8%, giving an RRF-minus-random effect of 43.7 points (95% CI [35.1, 52.2]). The random and RRF prompts average 722.85 and 726.04 tokens; their mean gap is 3.19 and the maximum is 8. The corresponding support-recall effect is 66.3 points (95% CI [61.1, 71.5]). In the subgroup analysis, the F1 effect is 43.5 points across 80 bridge questions and 44.5 points across 20 comparison questions.

Table 32: HotpotQA strict-budget diagnostic with a content-blind random-refill control. Evidence tokens enforce $B = 512$; API input includes the fixed wrapper.

Condition	Answer F1	Evidence tokens	API input tokens	Support recall
No retrieval	1.0%	245.96	492.77	1.0%
Random refill, 20 seeds	12.0%	477.42	722.85	8.8%
RRF, $K = 8$	55.8%	480.61	726.04	75.1%

D NATURAL EXECUTION, PACKING, AND CALIBRATION

D.1 FIXED CONFIGURATION AND EVALUATION ENDPOINTS

Configuration and data. This development diagnostic fixes the Qwen warm-4, 512-token configuration, with 120 evaluation questions from 13 families, a nine-entry program library, and resolver v7. The design and source hashes are committed before execution. Runtime uses no answer or support labels and commits 360 attempt records and paired packing records before scoring. All 120 questions remain in the results. The 210-question pooled analysis and the 1,530-configuration analysis use different populations.

Metrics. Candidate availability means that a valid two-hop program is selected; executability means that it produces a complete execution record and evidence package. Acceptance is the verifier's decision. Strict joint validity additionally requires the predicted reference route, exact support-coordinate agreement, terminal compatibility, and answer EM. External correctness alone uses answer EM, without strict support agreement. Conditional error is the fraction of accepted answers that are wrong; it is undefined when no answer is accepted. Hybrid EM/F1 emits the terminal on acceptance and otherwise uses the same recorded baseline answer. For Whole/Compose/Direct, external-correct accepts are 9/13/13; strict-invalid accepts are 1/1/2, and wrong-answer accepts are 0/0/1. The strict-invalid and wrong-answer events overlap. Conditional answer-error rates are 0/9, 0/13, and 1/14, respectively.

D.2 EXECUTION BOTTLENECKS AND SEMANTIC ERRORS

Of 360 attempts, 171 lack a candidate, 153 fail in resolution, and 36 complete and pass all three guards. Whole has 38 resolution failures, Compose 50, and Direct 65. Six failures preserve a completed first hop (one Compose, five Direct); their incomplete records are not promoted into valid proofs. There are no seed, admission, or packing failures in this primary cell. The three verifiers make identical decisions and reject no complete, valid record. Whole and Compose also reproduce the original candidates, decisions, answers, and scores.

The 153 resolution failures account for 81.0% of the 189 method–question attempts with a candidate. The 189 attempts include repeated questions across methods. Table 33 lists the recorded exceptions. Most report missing transition patterns. A missing pattern can reflect either missing retrieved evidence or a failure to match admitted text. Separating these causes requires inspecting each failed hop.

Table 33: Recorded resolution failures among 189 candidate-bearing method–question attempts. Counts are taken from the frozen attempt ledger.

Recorded exception	Attempts
`transition_pattern_absent`	138
`ambiguous_relation_cardinality`	6
`terminal_pattern_absent`	6
`input_title_not_selected`	3
Total	153

A semantic error shared by both checks. The extra Direct acceptance follows the predicted `composer` → `award received` route and emits `National Film Award`, a StringSpan. Its answer scores zero EM. Full and fresh Replay both accept the unchanged natural record. The error persists because Replay uses the same resolver and task interpretation. The 35 other complete method–question attempts have Date terminals. Together, these 36 attempts include repeated questions across methods. The published ledger identifies the failing question for reproduction with the unchanged runtime.

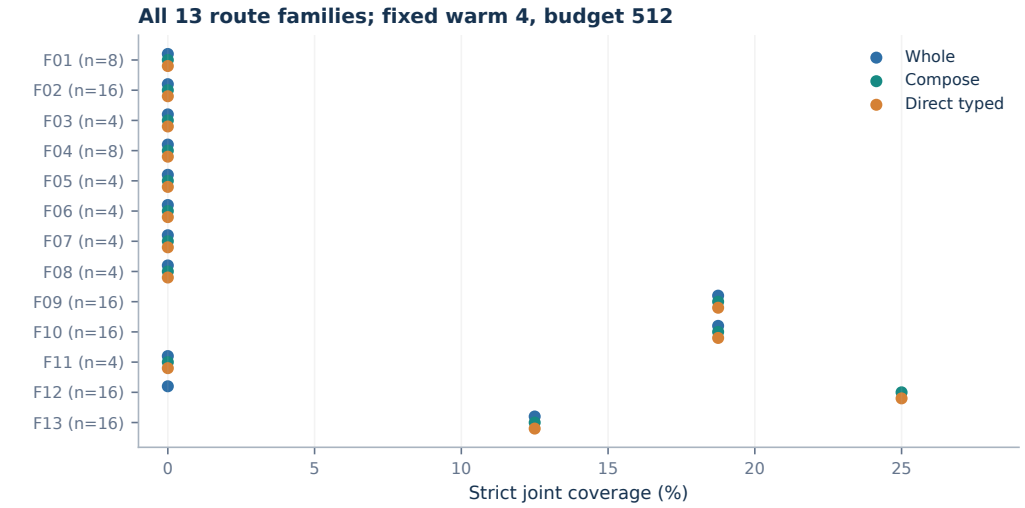


Figure 6: All route families, including zero effects. F01–F13 follow the lexicographic family-ID ordering in the accompanying table. Direct and Compose tie in strict coverage in every family. Compose adds four valid accepts over Whole only in F12 (16 questions). Family IDs are grouping keys, not runtime inputs.

D.3 Evidence retention under delivery budgets

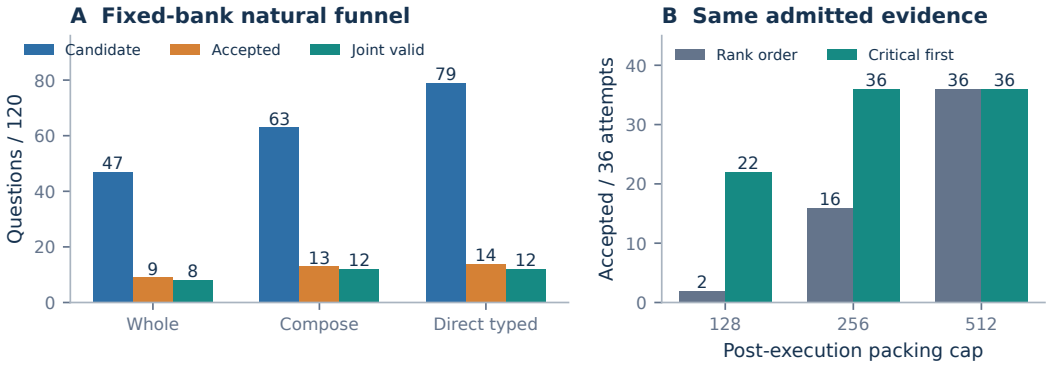


Figure 7: Fixed-configuration funnels and paired packing. The 36 complete method–question attempts share questions and are not independent samples. At cap 128, 14 selected witnesses exceed the cap; no packing policy can retain those same witnesses.

Packing intervention. The primary pipeline uses the same budget B for pre-execution evidence admission and final proof packing. To isolate retention after execution, the paired intervention fixes $B_{\text{exec}} = 512$ and varies only $B_{\text{emit}} \in \{128, 256, 512\}$ on the 36 completed attempts. Both policies receive the same complete program, admitted 512-token evidence, supporting dependencies, and ranking. Only the post-execution cap and packing policy change. At caps 128/256/512, ordinary packing preserves and accepts 2/16/36 of the 36 method–question witnesses; dependency-first packing preserves 22/36/36. Fourteen witnesses require more supporting-evidence tokens than cap 128 permits; none exceed 256 or 512. Per-attempt ratios, discarded records, rejection reasons and packing times are released. The cap constrains the selected supporting records; an alternative derivation may fit. Semantic labels remain fixed in this post-execution comparison. Appendix E reports the rerun on all questions with separate execution and delivery budgets. Appendix F reports fresh synchronous calls separately from replay with recorded fallback answers.

D.4 FALLBACK AND PLANNER–BINDER DIAGNOSTICS

Fixed-trace versus online fallback. We pair 5,184 suffix requests per budget and check $Y_{\text{full}} = GV + \mathbf{1}\{\text{fallback succeeds}\}$ on every row. Holding the full trajectory fixed and suppressing fallback gives GV, whereas the online rerun also stops adding programs after the prefix. There are zero identity failures at either budget.

Table 34: Completion percentages under distinct interventions.

Policy	$B = 256$	$B = 512$
Full trajectory	54.96	61.90
Same trajectory, suppress fallback (GV)	44.02	49.54
Online no-fallback rerun, library also changes	31.48	35.46

At 256 tokens the mean primitive-library sizes are 4.54 versus 3.40, and candidate rates 52.16% versus 42.88% for full and online no-fallback. The two runs therefore differ in both fallback and subsequent program reuse.

Phi planner/binder interface. We recompute the alignment function on all 240 Phi planner records. There are 56 non-null validated candidate programs, but no aligned executable two-hop program. Final reason counts are 172 planner-absent, 13 binder-reject, five unknown relation, 19 planner/binder mismatch, and 31 unsupported proof topology. The counts follow the filter order; a later failure is recorded only after earlier checks pass. Every corresponding frozen Stage-A status matches, confirming that the zero executable-program outcome persists under the interface check.

D.5 COSTS AND AMORTIZATION

Instrumented Full attempt means are 3.16/4.44/5.92 ms for Whole/Compose/Direct, including failed attempts; p95 values are 15.58/15.21/16.85 ms. These single-run CPU measurements include instrumentation but exclude binder inference, program acquisition, data loading, and synchronous fallback. The recorded reader calls cost 37,404 tokens for Always-Agent, versus 34,016/32,365/31,707 remaining fallback tokens. The CPU and recorded-reader measurements are separate components, not a synchronous total-latency measurement. A deployment saves cost only if

$$C_{\text{binder}} + C_{\text{attempt}} + \mathbb{E}[(1-G)C_{\text{agent}}] + C_{\text{acquire}}/N < \mathbb{E}[C_{\text{agent}}].$$

Estimating break-even requires the remaining cost terms and the dependence between acceptance and agent cost.

D.6 SECONDARY CALIBRATION BY ROUTE

Route-level admission rule. For the original rule, define $a_r(t)$ and $e_r(t)$ as the accepted and accepted-error fractions divided by all questions in route group r.

Proposition 6 (Sufficient route-level admission rule)**.** *For a fixed fitted model, m independent identically distributed calibration route groups, target α, and unchanged target route population, put*

$$U(t) = \frac{1}{m}\sum_{r=1}^{m}[e_r(t) - \alpha a_r(t)] + \sqrt{\frac{\log(KS/\delta)}{2m}}. \tag{18}$$

Across S predeclared analyses, with probability at least $1-\delta$, every rule selected with $U(t) \le 0$ has $\mathbb{E}[e_r(t)]/\mathbb{E}[a_r(t)] \le \alpha$ whenever $\mathbb{E}[a_r(t)] > 0$.

Hoeffding's inequality applies because $e_r - \alpha a_r \in [-\alpha, 1-\alpha]$; a union bound covers all KS tests. The analyses need not be independent. We maximize calibration coverage among passing rules or decline reuse.

Bounded-loss KL sensitivity. For each route group, $0 \le e_j \le a_j \le 1$. Set $X_j = e_j + \alpha(1 - a_j) \in [0, 1]$, so $\mathbb{E}[X_j] \le \alpha$ is equivalent to $\mathbb{E}[e_j] \le \alpha \mathbb{E}[a_j]$. For a fixed threshold under independent identically distributed route groups, the bounded-loss Chernoff test gives

$$p = \exp\{-m \, \mathrm{kl}(\min(\bar{X}, \alpha), \alpha)\}.$$

We use the bounded-loss test of Angelopoulos et al. (2021) because group errors can be fractional. We use the same 12 fixed thresholds and five partitions, with cutoff $.05/60$; there are zero passes. At zero error and full coverage, $p = (1 - \alpha)^m$ yields 139 groups; the Hoeffding sufficient condition yields 1,419. Both calculations assume independent route sampling, an assumption not established by these development splits.

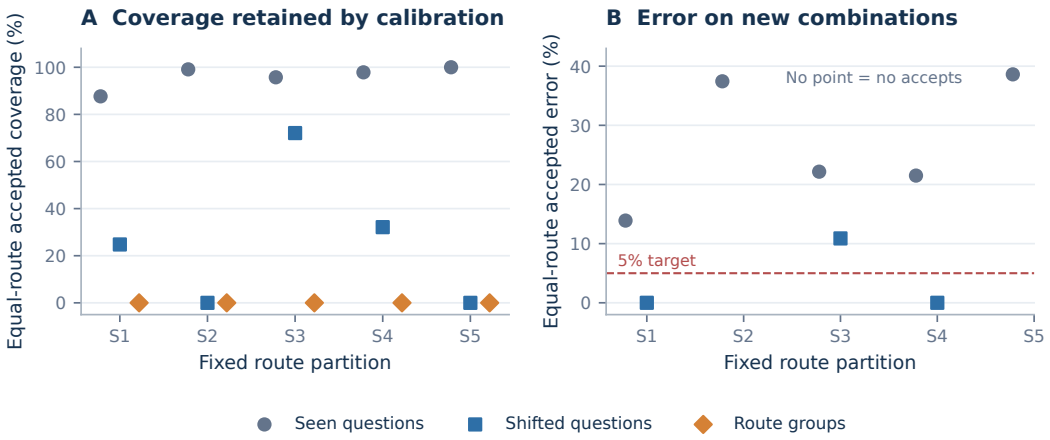


Figure 8: Secondary five-partition calibration diagnostic. Missing error points mean zero acceptance, not zero risk. Partitions overlap.

E Separated budgets and admission controls

These studies use the 120-question, 13-family development set, a fixed binder, and a nine-entry program library. We compare library-based admission rules and rerun local execution with separate evidence budgets, using no new model calls. Coverage is measured over all 120 questions, including missing candidates and failed executions.

E.1 Comparison of program admission rules

We compare the frozen Full decisions with four simple restrictions on the Direct candidate: terminal relations present in the program library; nonterminal relations present in it; membership of all primitives in the library's primitive set; and a rule that uses no library and requires the predicted terminal relation to be `date of birth` or `date of death`. The rules operate on predicted programs without reference terminal types or answers. They share the fixed Direct execution and its scores, isolating admission from execution.

Table 35: Opened-data admission controls. Each row contains all 120 questions. Correct counts external-answer EM; Joint additionally requires strict route, support, and terminal agreement. The last four rows are simple rules.

Rule	Candidates	Accepts	Correct	Joint	Hybrid EM (%)
Direct	79	14	13	12	38.33
Whole	47	9	9	8	38.33
Compose	63	13	13	12	39.17
Terminal relation seen	63	13	13	12	39.17
Nonterminal relation seen	79	14	13	12	38.33
All primitives seen (set membership)	63	13	13	12	39.17
Date-terminal only (no bank)	63	13	13	12	39.17

The nine library entries contain three distinct programs: performer–death, performer–birth, and composer–death. Of the 120 targets, 41 have no Direct candidate, 47 have a whole program present in the library, 16 have all primitives present but an unseen combination, and 16 lack a required primitive. The unseen combination is composer–birth; its 16 targets account for all four extra COMPOSE accepts over WHOLE. Every candidate admitted through the library is identical to the Direct program and has the same execution outcome. The implementation therefore expands coverage by admitting combinations already predicted by Direct.

COMPOSE, terminal support, primitive membership and the Date restriction agree on every question's candidate and acceptance decision. The extra Direct error follows the predicted composer–award route, outside the Date rule's terminal domain. The Date rule reproduces the gain without historical programs. The observed advantage is therefore also explained by a restriction on terminal type.

Margin controls and family-held-out sensitivity. At the same 13 accepts, a Direct margin threshold yields 12 externally correct and 11 joint-valid accepts, versus 13 and 12 for COMPOSE; at nine accepts, margin filtering yields 8 correct and 7 joint-valid, versus 9 and 8 for WHOLE. These descriptive results are also obtained with the Date rule. We also choose a margin threshold using labels from the other 12 families and evaluate the held family, repeating over all 13 families. The fixed threshold grid and selection goals are specified in the released protocol. Only threshold-selection labels exclude the held family; the program library remains fixed. This analysis measures threshold sensitivity with an unchanged library.

Table 36: Other-family threshold selection, evaluated on all held-out questions and pooled over the same 120-question panel. Entries are accepts / external errors / joint-invalid accepts. Date-only agrees with COMPOSE throughout.

Training selection criterion	Direct	WHOLE	COMPOSE / Date-only
Empirical external error $= 0$	7/1/2	9/0/1	13/0/1
Empirical joint error $= 0$	2/0/1	4/0/1	4/0/1
Acceptance coverage $\leq 7.5\%$	9/1/2	9/0/1	8/0/1
Acceptance coverage $\leq 10\%$	12/1/2	9/0/1	11/0/1
CP95 external-risk bound $\leq 5\%$	0/0/0	0/0/0	0/0/0

Conditional risk is undefined when no answer is accepted. The uncorrected single-threshold CP criterion rejects every policy with nonzero acceptance because the accepted samples are small. The bound is uncorrected for threshold selection and family dependence. The Direct external-error criterion and all joint-error criteria show that zero training error can coexist with errors on the held family. Complete risk–coverage curves retain margin ties and include rejection of all questions.

E.2 SEPARATE EXECUTION AND DELIVERY BUDGETS

The interface passes B_{exec} to evidence admission and execution, and B_{emit} to packing and verification. For each question, method, and execution budget, the executor runs once; all delivery policies share that attempt, and evaluation includes failed attempts. The grid uses $(B_{\text{exec}}, B_{\text{emit}}) \in \{(256, 128), (256, 256), (512, 128), (512, 256), (512, 512), (1024, 128), (1024, 256), (1024, 512)\}$, three methods, and three policies: dependency-first packing, rank-order, and keep-all. The 8,640 rows reuse 1,080 local executions on 120 questions. Keep-all preserves the admitted order and falls back when the sequence exceeds the delivery budget.

The delivery budget constrains accepted executions. Rejections use the recorded fallback answer from a separate 512-token reader context. Reported EM/F1 therefore describes this replay policy, with fresh inference and synchronous latency evaluated separately in Appendix F.

Table 37: Full-question budget results at $B_{\text{exec}} = 512$. A/J denotes accepted / joint-valid fast paths, each out of 120; EM is hybrid answer accuracy in percent with the same historical fallback. Packing decisions use no offline score labels.

		Dependency-first packing		Rank-order		Keep-all	
Method	B_{emit}	A/J	EM	A/J	EM	A/J	EM
COMPOSE	128	8/7	38.33	1/1	35.00	0/0	35.00
	256	13/12	39.17	6/5	38.33	2/1	35.83
	512	13/12	39.17	13/12	39.17	13/12	39.17
Direct	128	8/7	38.33	1/1	35.00	0/0	35.00
	256	14/12	38.33	6/5	38.33	2/1	35.83
	512	14/12	38.33	14/12	38.33	14/12	38.33
WHOLE	128	6/5	38.33	0/0	35.00	0/0	35.00
	256	9/8	38.33	4/3	37.50	1/0	35.83
	512	9/8	38.33	9/8	38.33	9/8	38.33

At 512/256, COMPOSE has 57 absent candidates and 50 resolver failures before packing. Of the 13 remaining attempts, dependency-first packing accepts all; rank-order loses dependencies in seven and the verifier rejects them; keep-all exceeds the cap on 11 and accepts two. One of the 13 dependency-first packing accepts remains strictly invalid. At 512/128, dependency-first packing cannot fit five of the selected witnesses and accepts eight. Those selected records exceed the budget under every policy; an alternative derivation may fit. The logs retain these failure stages, fallback decisions, and actual emitted evidence tokens for every condition. At equal 512-token budgets all policies agree, as expected because preserving the admitted sequence is feasible.

Family distribution and uncertainty. At 512/256, dependency-first packing adds seven joint-valid accepts over rank-order for both COMPOSE and Direct: 5.83 percentage points over all 120 questions. The gains are 1, 2, 2, and 2 in families F09, F10, F12, and F13 (the lexicographic ordering of Appendix D); the other nine families have zero difference. WHOLE gains five accepts across F09, F10, and F13, or 4.17 points. An exploratory paired cluster bootstrap draws 13 families with replacement, retains all paired questions within each draw, and divides total differences by the resampled question count. With 20,000 draws and seed 20260918, the 95% percentile intervals are $[1.85, 9.09]$ points for COMPOSE/Direct and $[0.00, 7.58]$ for WHOLE. These exploratory intervals describe the post-hoc comparison across the 13 observed families. Here packing improves retention, while the admission controls match library-based selection.

Reproduction. The released runs are `runs/repair_history_admission_20260918_v1`, `runs/repair_dual_budget_20260918_v1`, and `runs/repair_dual_budget_family_20260918_v1`. Their builders are `experiments/build_history_admission_revision.py`, `experiments/build_dual_budget_revision.py`, and `experiments/analyze_dual_budget_family_revision.py`. The first run includes all question-level rules, thresholds, and family folds; the second contains the complete budget grid and failure records; the third contains paired family differences and the resampling protocol.

F EVIDENCE DELIVERY AND SYNCHRONOUS COST

F.1 DELIVERY BETWEEN SEPARATE PRODUCER AND RECEIVER PROCESSES

The main implementation creates and verifies an execution record in one process. Here the producer writes the record to an envelope, flushes and synchronizes the file to disk, and closes it before a separate receiver opens it. The untrusted component is the envelope cache or transfer path. A separate trusted request registry supplies the question, program, task contract, current source records, version pins, and delivery budget. Expected programs come from the frozen Stage C records created before the proof. The receiver checks the contract and candidate evidence against those records, independently of the received proof.

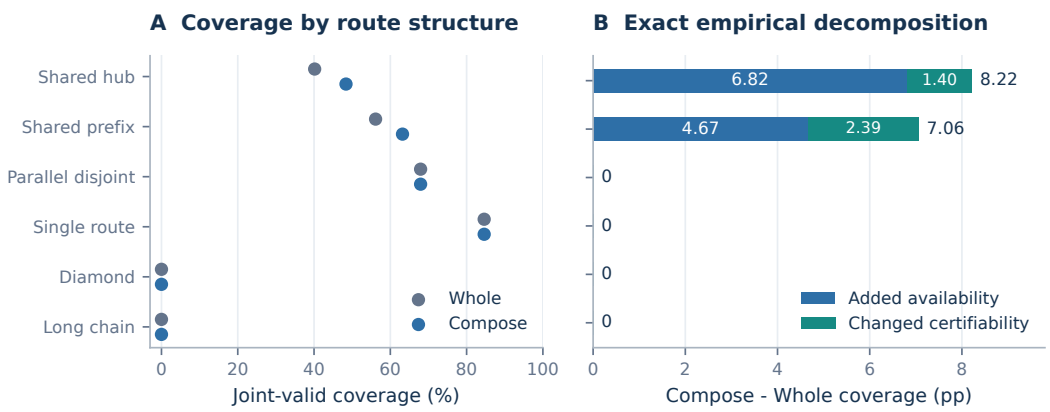


Figure 9: Controlled coverage decomposition at $B = 256$: 864 suffix requests per topology and method. Both positive and zero-effect structures are shown.

The envelope binds its proof to a digest of this context, including request identity and cache generation. HMAC-SHA256 authenticates the canonical JSON body. A fresh key is passed privately to the receiver and is absent from saved artifacts. Catalog order and metadata outside source-record identity do not change the context digest. Changes to current source bytes or version pins do. The registry, version authority, and key distribution remain trusted.

We compare AUTHENTICATED, AUTHENTICATED+FULL, and AUTHENTICATED+REPLAY. All three apply the same authentication, current-context binding, strict wire-schema parsing, and exact evidence-budget check. Full adds the existing execution-record verifier; Replay also reruns resolver v7. The comparison isolates structural checking beyond authentication. Delivery metadata, including the cache-generation counter, is authenticated by the envelope layer. The evidence budget counts the same serialized evidence lines as the original experiment; complete envelope bytes are recorded separately and subject to a file-size limit.

F.2 CONTROLLED LIFECYCLE AND PRODUCER DIAGNOSTICS

The development panel contains 13 distinct execution records from four families, all accepted by Full in a previous run. Each proof receives the same 12 declared scenarios. Three classes retain a valid delivery: unchanged delivery, JSON reserialization, and reordered source catalogs with additional non-source metadata. Seven classes introduce faults: source-version updates, source-content updates, stale cache generations, cross-request cache mixups, truncated files, lineage corruption after signing, and attempts to rebind an old envelope to a new context without a valid signature. The interventions measure detection of specified faults rather than their natural frequency.

Two producer diagnostics are reported separately. In one, an authenticated producer signs a proof with missing dependency edges. In the other, it signs an object that exceeds the current evidence budget. These diagnostics relax the correct-producer assumption, whereas the lifecycle and transport cases alter delivery or receiver state after production.

Table 38: Accepted deliveries in the persisted-witness panel. Every entry is the count out of 13 distinct base proofs *in each scenario class*. All classes within a row have the displayed result. Classes and timing repeats do not increase the number of independent proofs.

Scenario group	Classes	Auth.	+Full	+Replay
Clean and benign delivery	3	13/13	13/13	13/13
Lifecycle or transport fault	7	0/13	0/13	0/13
Producer-signed invalid lineage	1	13/13	0/13	0/13
Producer-signed over-budget object	1	0/13	0/13	0/13

Table 38 shows no additional lifecycle or transport detections from Full or Replay over the authenticated baseline. Full's additional rejection occurs only in the producer-signed lineage diagnostic; Replay adds no further rejection on this panel. The clean answers remain externally correct for 13/13 proofs and strict jointly valid for 12/13. All three policies accept the strict-reference mismatch. Authentication establishes origin and freshness, Full checks internal structure, and offline answer and evidence scoring assesses semantic validity.

Receiver microbenchmark and reproduction. Each proof–scenario–policy combination is timed three times after tokenizer initialization. Clean-delivery mean/median receiver times are 1.761/1.623 ms for Authenticated, 2.453/2.289 ms with Full, and 5.598/3.522 ms with Replay. Timing includes envelope file reads, parsing, context-digest calculation, and policy checks. It excludes trusted-context loading, process startup, producer work, model inference, and fallback. The experiment records file, source, tokenizer, and input hashes, together with scenario decisions and individual timing repeats. Reproduction uses a fresh ephemeral authentication key; saved signatures cannot be reverified without rerunning the producer–receiver exchange. The experiment makes no model calls.

F.3 Fresh acquisition and synchronous online accounting

We run Qwen3-4B-Instruct-2507 locally through vLLM 0.13.0 on an RTX 4070 Ti SUPER with 16 GB GPU memory. The serving process uses bfloat16, an 8,192-token context cap, one simultaneous sequence, temperature zero, and fixed question-specific seeds. Server-default prefix caching is shared across methods. The planner prompt, relation catalog, reader prompt, and parser match the natural execution study; no reference answer enters a request. Gold scoring starts after the completed runtime ledger has been committed. The workload is the same development set.

The original binder is fitted on 154,005 residual training questions and calibrated on 353 residual questions. No candidate ID enters those pools. The newly fitted model reproduces every original candidate route and accept/reject decision across the 240 public questions; margins have small numeric differences. Loading training inputs, fitting, and calibration take 1.014, 176.451, and 0.044 s. The model is loaded once for the service session, but each typed-policy request performs a new binder prediction.

Dedicated acquisition processes all 120 public warm questions. Each makes a fresh planner call; planner–binder agreement is required before typed execution and Full verification. A successful execution adds its program to the library. Unsuccessful warm attempts still make a reader call against the fresh planner's retrieved evidence, so their model and elapsed costs remain in the acquisition total. Ten entries enter the program library, compared with nine in the fixed-budget replay. Acquisition therefore also changes the available programs. Total acquisition is 138.671 s and 123,174 model tokens; assembling the library objects from those results takes 0.00058 s.

The five arms then run on the same 120 suffix questions, in independently shuffled method order within each question. Whole, Compose, and Direct use execution cap 512, delivery cap 256, and in-process Full verification. Compose additionally has a rank-order arm. Always-Agent calls the planner and reader for every request; rejected typed attempts call them anew with the same question and seed. Online answers use fresh model calls. The fallback reader retains its independent 512-token context cap. The 600 online requests and 120 warm requests make 1,346 model calls, plus one service warm-up call. Table 5 includes both planner and reader tokens. It does not include the separate authenticated-file receiver in the online path.

Observed cost and answer outcomes. The online baseline answers 44/120 questions correctly. Compose with dependency-first packing answers 49, while Whole, Compose with rank order, and Direct answer 48. Compared per question with the baseline, Compose improves five and worsens none; Direct improves five and worsens one. The strict-valid fast-path counts remain 8, 5, 12, and 12 for Whole, Compose with rank-order packing, Compose with dependency-first packing, and Direct with dependency-first packing. These estimates and the recorded p95 latency come from one interleaved pass; reliable tail-latency estimates require repeated runs.

Cost accounting. Each request's wall clock begins before binder inference and ends after deterministic emission or its actual synchronous fallback. It includes candidate selection, retrieval, failed attempts, packing, and verification. All model usage counters reconcile to prompt plus completion to-

kens. The initial 1.657 s session load and 1.859 s model warm-up are recorded separately as common service setup. All arms use an initialized model server; server startup is outside the measured interval. Request ledgers are persisted after each timed request.

Amortization and source-refresh sensitivity. Let d be the measured per-request saving over Always-Agent and C_0 the charged setup. Under an unchanged workload, the projection is $N_{\text{break}} = \lceil C_0/d \rceil$ for $d > 0$. Dedicated acquisition charges the entire warm phase; existing-history deployment treats that phase as sunk and charges reconstruction. We separately report an already trained binder and fitting for this deployment. If a full library refresh costs C_r every U requests, the projected saving is $d - C_r/U$; nonpositive values have no finite break-even point under these assumptions.

For Compose with dependency-first packing, measured online time saving is approximately 0.1233 s per request. Dedicated acquisition plus binder fitting costs 316.181 s, giving 2,566 projected suffix requests to recover it. Token acquisition is recovered after 1,010 projected requests. With existing history but a newly fitted binder, the time projection is 1,441 requests; with dedicated history and an already trained binder, it is 1,126. Refreshing the complete acquired library every 1,000 requests would eliminate the projected time saving. The refresh projection uses the measured acquisition cost each time. Direct execution is slightly cheaper in the observed pass, and the 120 suffix requests fall short of the projected break-even points.

Reproduction and audit. `experiments/run_live_cost_revision.py` provides separate preparation, runtime, and scoring phases. `experiments/analyze_live_cost_revision.py` reconciles all 720 request records, pairs answer changes, and produces existing/dedicated-history and binder-training scenarios. Runtime hashes precede scoring; dependency and tokenizer audits are performed after the run. Reproducing the timing measurements requires live inference with the model weights and original dataset.

G External evidence delivery on MuSiQue

Evidence retention with predicted citations. This experiment tests whether preserving dependencies in a predicted citation graph improves answers under a small delivery budget. MuSiQue-Ans contains connected two- to four-hop questions (Trivedi et al., 2022). The adapter replaces typed execution dependencies with model-predicted citations and uses no program library. Its checks establish source membership and graph consistency rather than entailment or typed-execution validity.

Panel and data access. We select 30 official training questions for development and 240 official development questions for evaluation, balanced across the three hop counts. Within each hop stratum, IDs are sorted by a fixed seeded SHA-256 digest; selection uses neither labels nor model outputs. The selected training and evaluation IDs share neither complete question IDs nor constituent single-hop IDs. Runtime files contain only the question and original paragraphs, with paragraph IDs and titles. Answers, aliases, supporting labels, and decompositions are stored separately for offline scoring. The selected IDs, prompts, runtime source snapshots, model/tokenizer fingerprints, and configuration are committed before evaluation inference; prediction ledgers are committed before scoring. The balanced evaluation panel comes from the public development split; hidden-test performance and pretraining exposure are unknown. We develop the configuration with Qwen and transfer it unchanged to Mistral.

Common view and outputs. A fixed sentence splitter preserves exact source spans. BM25 orders records using the question and title-plus-sentence text. Whole records are admitted greedily to a common $B_{\text{exec}} = 4096$-token view, counted with the respective model tokenizer, including IDs, titles, and separators. All methods for the same question and model see exactly this view. It can differ between models because their tokenizers differ. Instructions and question text are outside the evidence budget but included in actual model-token accounting.

One producer returns a short answer, evidence steps with sentence IDs and parent-step references, a terminal step, and a ranked sentence-ID list. The validator requires known source IDs, unique step IDs, topologically ordered parents, one to eight steps, and the last step as terminal. It derives the terminal's ancestor closure. Invalid graphs and producer abstentions remain in the denominator. The reader receives only the question and original selected sentences, never the producer answer, graph,

or generated reasoning. The producer and readers use temperature zero, fixed question-derived seeds, and completion limits of 1024 and 96 tokens, respectively.

Matched policies and references. At $B_{\text{emit}} \in \{256, 512\}$, we compare BM25, BGE-small-en-v1.5, the published RECOMP HotpotQA extractive checkpoint (Xu et al., 2024), LLM-rank, citation-first, and dependency-first packing. BGE uses its query instruction, CLS pooling, and normalized embeddings; RECOMP uses attention-mask mean pooling and unnormalized dot products. The checkpoint is adapted to the common sentence view and delivery budget. All policies use whole-record greedy packing. LLM-rank uses the shared producer's ranking. Citation-first protects every valid cited record; Dependency-first packing protects only records cited by the terminal and its ancestors. If the protected set exceeds the cap, packing falls back to its fixed greedy order and records that failure. Every packet is read, without a verification gate or agent fallback. Producer-direct, a reader on the entire admitted view, and a question-only reader provide three additional references: 15 outputs per question per model.

Compute configuration. Inference uses one RTX 4070 Ti SUPER (16 GB) with vLLM 0.13.0 under WSL. Qwen3-4B-Instruct-2507 uses bfloat16; Mistral-7B-Instruct-v0.3 uses bitsandbytes 4-bit loading with `dtype=half`. Both services use an 8192-token context limit, one concurrent sequence, and 0.75 GPU memory utilization. The BGE and RECOMP encoders run with four CPU threads on a Ryzen 7 9700X, batch size 32, and a 512-token encoder input limit. The artifact manifest pins model weights, tokenizers, pooling rules, and software versions. The two configurations differ in model identity, size, and precision.

Matched responses and cost. An identical complete reader prompt for the same model and question reuses one recorded response. A training pilot produced different answers to repeated identical prompts and seeds, so we fixed this sharing rule before evaluation. Different packets receive fresh calls; sharing never crosses questions or models. Each policy allocation includes its complete applicable view construction, ranking, producer, packing/checking, and reader costs, even when operations were shared in this controlled experiment. Policy allocations overlap because each includes the calls it would require; the physical request ledger counts each shared call once. Dense encoders run on CPU and the language model on GPU. Both generation services enable prefix caching, and the client memoizes evidence token counts across policies. The allocated times therefore reflect the experiment's shared cache state. We allocate full required model-token usage to each policy and report loading, development, and any failed calls separately. Timings cover initialized services with CPU encoders and shared caches, rather than isolated policies with GPU encoders.

Endpoints and uncertainty. Answer EM and F1 maximize over official aliases and are cross-checked against the unmodified official evaluator. The primary paired contrast is dependency-first packing minus LLM-rank at 256 tokens, separately for each model. We use 10,000 paired bootstrap samples within hop strata and an exact two-sided McNemar test for EM, with Holm adjustment over the two primary model tests. Other contrasts and the 512-token budget are secondary. The citation-first contrast additionally reports exact packet equivalence. The producer is instructed to connect every needed step to the terminal. When all cited records belong to terminal ancestors, dependency-first and citation-first packing protect the same records and emit identical packets. They can differ only when disconnected extra steps cite additional records, which the parser permits. Packet-equivalence diagnostics exclude invalid graphs, while answer-quality metrics retain all questions. Paragraph-support coverage measures overlap with annotated paragraphs, not sentence sufficiency or semantic validity.

The panel's public constituent single-hop IDs induce 130 connected components, including 88 singletons and a largest component of ten questions. A prespecified secondary bootstrap resamples components, retaining every member and averaging the three within-hop effects equally. This checks sensitivity to shared constituent questions, while other sources of dependence may remain. The analysis implementation was separately committed during inference, before scoring; the analysis specification precedes the first evaluation request. Both interval constructions estimate answer-quality differences; neither is a test of population equivalence.

G.1 RESULTS

Both configurations complete all 3,600 prediction cells, with no logged failed model calls. All 15 conditions per model agree with the official evaluator on answer EM, answer F1, and supporting-paragraph F1. The 7,200 predictions use 5,202 physical model calls: 5,243,084 prompt and 273,021 completion tokens, or 5,516,105 generator tokens in total. Tables 39 and 40 retain every condition and failure. The primary differences and both interval constructions appear in Tables 41 and 42. Preserving predicted dependencies adds no observed answer-quality gain in either model. Qwen's gain over BM25 and BGE is already present with the same producer's ranking, while RECOMP matches dependency-first packing's EM. Mistral's RECOMP F1 exceeds dependency-first packing by 2.84 points at 256 tokens.

Table 39: Qwen3-4B: all 15 conditions, each with 240 questions. EM/F1, paragraph support F1 (Supp.), complete supporting-paragraph coverage (Cov.), and predicted closure (Cl.) are percentages. Fail counts parsed-output failures. Paragraph coverage and closure are distinct from semantic validity. †Execution-view budget for producer-direct.

Policy	Cap	EM	F1	Supp.	Cov.	Cl.	Fail
BM25	256	10.42	16.95	39.02	14.17	2.50	0
BGE	256	14.17	20.58	49.28	24.17	6.67	0
RECOMP adaptation	256	19.17	27.70	44.84	18.33	5.83	0
LLM-rank	256	19.58	28.82	56.52	35.42	24.58	1
Citation-first	256	19.58	28.82	56.52	35.42	25.00	1
Dependency-first	256	19.17	28.47	56.37	35.83	28.33	1
BM25	512	12.08	19.69	35.25	32.92	8.75	2
BGE	512	17.92	24.87	42.89	40.42	11.67	0
RECOMP adaptation	512	18.75	29.09	42.08	35.42	10.83	1
LLM-rank	512	20.00	29.72	48.08	45.42	34.58	0
Citation-first	512	20.00	29.72	48.06	45.42	35.42	0
Dependency-first	512	20.00	29.74	48.06	45.42	35.42	0
Producer-direct	4096†	15.00	21.80	25.60	17.92	36.25	32
Full-exec-reader	4096	22.50	32.61	25.78	98.75	36.25	0
Question-only	0	0.00	0.00	0.00	0.00	0.00	0

Table 40: Mistral-7B (4-bit): all 15 conditions, each with 240 questions. EM/F1, paragraph support F1 (Supp.), complete supporting-paragraph coverage (Cov.), and predicted closure (Cl.) are percentages. Fail counts parsed-output failures. Paragraph coverage and closure are distinct from semantic validity. †Execution-view budget for producer-direct.

Policy	Cap	EM	F1	Supp.	Cov.	Cl.	Fail
BM25	256	6.67	10.63	39.92	12.92	7.08	0
BGE	256	7.50	12.45	49.52	19.58	6.67	1
RECOMP adaptation	256	7.08	14.36	44.07	14.58	5.00	1
LLM-rank	256	6.67	11.64	46.32	18.75	20.83	0
Citation-first	256	6.67	11.52	46.17	18.33	23.33	0
Dependency-first	256	6.67	11.52	45.94	18.33	24.17	0
BM25	512	6.67	11.14	36.27	28.75	7.92	0
BGE	512	7.50	12.49	44.05	35.83	12.08	1
RECOMP adaptation	512	7.50	15.10	42.74	31.25	9.17	1
LLM-rank	512	5.42	11.01	41.54	32.50	32.50	0
Citation-first	512	5.00	10.68	41.84	33.33	40.83	0
Dependency-first	512	5.00	10.68	41.82	33.33	40.83	0
Producer-direct	4096†	2.50	8.35	26.19	12.08	44.17	41
Full-exec-reader	4096	8.75	14.88	25.76	98.33	44.17	1
Question-only	0	2.08	5.50	0.00	0.00	0.00	0

Table 41: Prespecified dependency-first minus LLM-rank at 256 tokens: F1 differences and 95% intervals in percentage points. Question bootstrap is primary; shared-constituent component bootstrap is secondary sensitivity.

Model	ΔF1	Question CI	Component CI
Qwen3-4B	-0.35	$[-1.25, +0.15]$	$[-1.32, +0.16]$
Mistral-7B (4-bit)	-0.13	$[-1.00, +0.62]$	$[-1.00, +0.64]$

Table 42: The same primary contrast in EM: paired wins/losses and two-sided exact McNemar p-values. Holm adjustment covers the two model-specific primary EM tests.

Model	Wins	Losses	Raw p	Holm p
Qwen3-4B	0	1	1.0000	1.0000
Mistral-7B (4-bit)	0	0	1.0000	1.0000

Where information is lost. The common execution view contains every annotated supporting paragraph for 237/240 Qwen and 236/240 Mistral questions. Complete sentence coverage is unmeasured. Only 87 and 106 producer outputs have valid citation graphs. At 256 tokens, dependency-first packing's protected set overflows on 19/240 and 48/240 questions, respectively; among valid graphs the rates are 19/87 and 48/106. Citation-first overflows on 31/87 and 50/106 valid graphs. Reducing this structural overflow does not produce a corresponding answer-quality improvement. The results suggest three possible bottlenecks: support selection, graph construction, and reader use of evidence. Graph consistency alone does not establish that the cited statements support the answer.

Scope beyond date-like answers. A descriptive, post-hoc lexical classification of the canonical gold answer marks 46 questions as date-like and 194 as non-date-like. This grouping uses surface form and takes place after panel selection. On the non-date-like group at 256 tokens, Qwen dependency-first packing obtains EM 34/194 and F1 28.63, versus 35/194 and 29.06 for LLM-rank and 32/194 and 26.38 for RECOMP. Mistral obtains 11/194 and 11.10, versus 11/194 and 11.26 for LLM-rank and 13/194 and 13.13 for RECOMP. The non-date-like group likewise shows no answer-quality gain from preserving predicted dependencies. The released CSV includes both lexical groups and all hop groups.

Cost interpretation. Figure 10 charges both producer and reader tokens. Dependency-first packing at 256 tokens uses 12.38 times RECOMP's generator tokens with Qwen and 13.15 times with Mistral. Relative to the full-view reader references, it uses 31.26% and 29.76% more generator tokens despite delivering a smaller packet. The full-view reader requires a larger recipient interface than the 256-token cap and provides a reference for total inference cost. Tables 43 and 44 give full policy allocations. Actual CPU encoder work and physical calls are reported separately in Tables 45–49. The component durations include the effects of caches shared across policies.

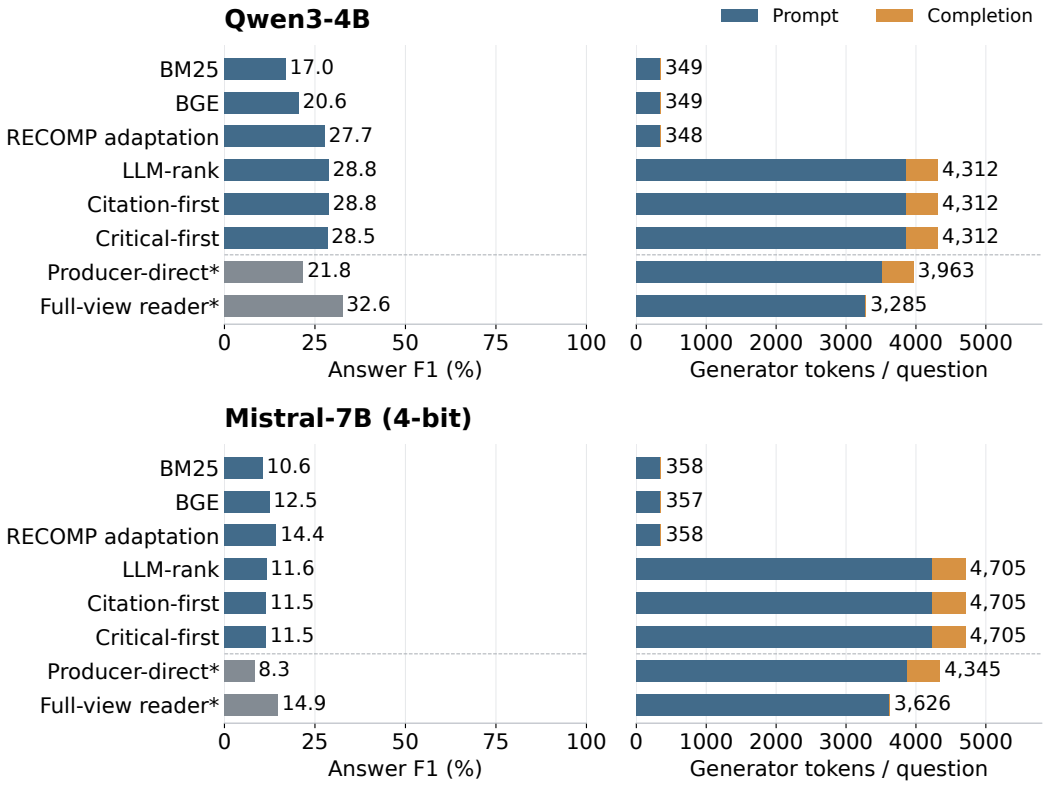


Figure 10: MuSiQue answer quality and allocated generator tokens per question, each over all 240 evaluation questions. The first six policies deliver at most 256 evidence tokens. Asterisks mark references using the 4096-token execution view: producer-direct answers without a reader; full-view reader receives the whole admitted view. Token bars charge every required producer and reader prompt and completion. CPU encoder work and latency are reported separately. The full-view reader requires a recipient budget above 256 tokens.

Table 43: Qwen3-4B: standalone-policy allocations for 240 questions per row. Generator prompt, completion, and total tokens are panel totals in thousands. Time is mean component-sum seconds per question, including the required producer, reader, and CPU ranking work; it is not separately measured synchronous end-to-end latency. Measured durations inherit shared prefix and token-count caches across experimental policies; they are not isolated-policy latencies. Session loading is separate. †Execution-view budget for producer-direct.

Policy	Cap	Prompt (k)	Compl. (k)	Total (k)	Time (s)
BM25	256	81.1	2.6	83.7	0.416
BGE	256	81.2	2.6	83.8	1.659
RECOMP adaptation	256	81.0	2.6	83.6	4.606
LLM-rank	256	925.5	109.5	1035.0	9.084
Citation-first	256	925.5	109.5	1035.0	9.084
Dependency-first	256	925.5	109.4	1034.9	9.082
BM25	512	142.5	2.8	145.3	0.460
BGE	512	142.6	2.7	145.3	1.715
RECOMP adaptation	512	142.5	2.8	145.3	4.667
LLM-rank	512	986.9	109.4	1096.3	9.115
Citation-first	512	986.9	109.4	1096.3	9.109
Dependency-first	512	986.9	109.4	1096.3	9.112
Producer-direct	4096†	844.4	106.6	951.0	8.834
Full-exec-reader	4096	785.6	2.8	788.5	0.758
Question-only	0	20.9	2.2	23.0	0.181

Table 44: Mistral-7B (4-bit): standalone-policy allocations for 240 questions per row. Generator prompt, completion, and total tokens are panel totals in thousands. Time is mean component-sum seconds per question, including the required producer, reader, and CPU ranking work; it is not separately measured synchronous end-to-end latency. Measured durations inherit shared prefix and token-count caches across experimental policies; they are not isolated-policy latencies. Session loading is separate. †Execution-view budget for producer-direct.

Policy	Cap	Prompt (k)	Compl. (k)	Total (k)	Time (s)
BM25	256	83.1	2.8	85.9	0.330
BGE	256	82.7	3.0	85.8	1.539
RECOMP adaptation	256	82.9	3.0	85.9	4.403
LLM-rank	256	1015.4	113.7	1129.1	6.896
Citation-first	256	1015.4	113.7	1129.1	6.897
Dependency-first	256	1015.5	113.7	1129.1	6.897
BM25	512	144.3	2.8	147.1	0.375
BGE	512	144.2	3.2	147.4	1.637
RECOMP adaptation	512	144.3	3.2	147.5	4.492
LLM-rank	512	1076.8	113.3	1190.1	6.948
Citation-first	512	1076.7	113.4	1190.0	6.946
Dependency-first	512	1076.7	113.4	1190.0	6.948
Producer-direct	4096†	932.3	110.5	1042.8	6.667
Full-exec-reader	4096	867.0	3.1	870.2	1.065
Question-only	0	23.0	2.8	25.8	0.172

Table 45: Source-admission, producer, and actual-run diagnostics. Admission uses the full-execution-view reader's paragraph IDs. Protected-set equality excludes all invalid graphs and abstentions. Actual inference usage counts shared calls once and is distinct from alternative-policy allocations. Unknown failed-call tokens are not zero.

Diagnostic	Qwen3-4B	Mistral-7B
Evaluated questions	240	240
Valid producer graphs	87	106
Producer abstentions	96	40
Invalid producer graphs	25	53
Invalid producer JSON	32	41
Valid protected-set comparisons	87	106
Equal critical/cited sets (valid only)	57	102
Strict-subset critical sets (valid only)	30	4
Admitted complete paragraph coverage (%)	98.75	98.33
Admitted paragraph recall (%)	99.58	99.38
Actual successful model calls	2,619	2,583
Unique reader calls	2,379	2,343
Reader references sharing calls	981	1,017
Actual generator prompt tokens	2,536,119	2,706,965
Actual generator completion tokens	133,304	139,717
Actual generator total tokens	2,669,423	2,846,682
Logged failed calls	0	0
Failed calls with unknown tokens	0	0

Table 46: Dependency-first versus citation-first: counts with equal selected-sentence sets, ordered sentence IDs, and reader answers. Same-packet disagreements count different answers for identical artifact hashes; the shared-response rule requires zero.

Model	Cap	n	Set eq.	Order eq.	Answer eq.	Disagree
Qwen3-4B	256	240	231	222	235	0
Qwen3-4B	512	240	239	221	238	0
Mistral-7B (4-bit)	256	240	236	236	239	0
Mistral-7B (4-bit)	512	240	238	236	240	0

Table 47: Valid-graph opportunities for dependency closure to differ from protecting all citations. Subset counts strict-subset critical evidence; Same/Diff. count identical/different final artifact hashes within those cases. Invalid graphs are excluded. Different protected sets need not change the delivered packet or answer quality.

Model	Cap	Valid	Subset	Same	Diff.
Qwen3-4B	256	87	30	12	18
Qwen3-4B	512	87	30	11	19
Mistral-7B (4-bit)	256	106	4	0	4
Mistral-7B (4-bit)	512	106	4	0	4

Table 48: Protected-set overflow at both delivery caps. All denotes the percentage over every evaluated question; Valid restricts the denominator to structurally valid producer graphs. Invalid graphs and abstentions have no protected set and remain in All's denominator; they are not protection successes. A missing valid-only rate is shown as –. Policies that do not protect a set are inapplicable and omitted here, with explicit status retained in the released CSV.

Model	Policy	Cap	n	Valid	Overflow	All (%)	Valid (%)
Qwen	Citation-first	256	240	87	31	12.92	35.63
Qwen	Dependency-first	256	240	87	19	7.92	21.84
Qwen	Citation-first	512	240	87	2	0.83	2.30
Qwen	Dependency-first	512	240	87	2	0.83	2.30
Mistral	Citation-first	256	240	106	50	20.83	47.17
Mistral	Dependency-first	256	240	106	48	20.00	45.28
Mistral	Citation-first	512	240	106	8	3.33	7.55
Mistral	Dependency-first	512	240	106	8	3.33	7.55

Table 49: Actual experiment durations and encoder input work, counting each question's context once. CPU stages are measured wall durations; complete ranking already includes its listed sub-stages and overhead. The unused RRF computation is included only in actual experiment work, not in any policy's required cost. Encoded texts are input counts, not encoder tokens. The commitment span includes initialization, bookkeeping, and any pauses; it excludes earlier service startup and is not pure inference latency.

Recorded component	Qwen3-4B	Mistral-7B
Actual common-view stage (s)	40.071	27.615
Actual complete ranking stage (s)	1304.869	1262.276
Ranking: BM25 (s)	0.381	0.379
Ranking: BGE (s)	298.400	287.947
Ranking: RECOMP (s)	1006.053	973.916
Ranking: unused RRF fusion (s)	0.013	0.013
Encoded texts per encoder	18,304	17,816
Actual successful model-call duration (s)	2728.138	2294.301
Commitment elapsed span (s)	4156.172	3659.336

Table 50: Qwen3-4B: all-condition hop breakdown, EM / F1 in percent; 2-hop $n = 80$, 3-hop $n = 80$, 4-hop $n = 80$. †Execution-view budget for producer-direct.

Policy	Cap	2-hop EM / F1	3-hop EM / F1	4-hop EM / F1
BM25	256	15.00 / 24.41	8.75 / 10.58	7.50 / 15.87
BGE	256	16.25 / 26.70	21.25 / 25.79	5.00 / 9.24
RECOMP adaptation	256	23.75 / 33.80	21.25 / 29.04	12.50 / 20.27
LLM-rank	256	26.25 / 35.28	22.50 / 28.85	10.00 / 22.33
Citation-first	256	26.25 / 35.28	22.50 / 28.85	10.00 / 22.33
Dependency-first	256	25.00 / 34.24	22.50 / 28.85	10.00 / 22.33
BM25	512	18.75 / 28.43	8.75 / 13.04	8.75 / 17.60
BGE	512	22.50 / 32.31	18.75 / 23.80	12.50 / 18.50
RECOMP adaptation	512	25.00 / 38.25	20.00 / 27.64	11.25 / 21.37
LLM-rank	512	27.50 / 39.06	20.00 / 25.29	12.50 / 24.82
Citation-first	512	27.50 / 39.06	20.00 / 25.29	12.50 / 24.82
Dependency-first	512	27.50 / 39.10	20.00 / 25.29	12.50 / 24.82
Producer-direct	4096†	22.50 / 31.65	12.50 / 17.28	10.00 / 16.47
Full-exec-reader	4096	31.25 / 40.19	23.75 / 32.00	12.50 / 25.63
Question-only	0	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00

Table 51: Mistral-7B (4-bit): all-condition hop breakdown, EM / F1 in percent; 2-hop $n = 80$, 3-hop $n = 80$, 4-hop $n = 80$. †Execution-view budget for producer-direct.

Policy	Cap	2-hop EM / F1	3-hop EM / F1	4-hop EM / F1
BM25	256	11.25 / 17.95	2.50 / 4.33	6.25 / 9.61
BGE	256	15.00 / 21.31	3.75 / 9.50	3.75 / 6.54
RECOMP adaptation	256	15.00 / 22.29	2.50 / 12.30	3.75 / 8.50
LLM-rank	256	11.25 / 19.38	3.75 / 6.14	5.00 / 9.40
Citation-first	256	11.25 / 18.38	3.75 / 6.14	5.00 / 10.03
Dependency-first	256	11.25 / 18.38	3.75 / 6.14	5.00 / 10.03
BM25	512	11.25 / 19.16	3.75 / 5.15	5.00 / 9.09
BGE	512	13.75 / 23.47	6.25 / 9.32	2.50 / 4.68
RECOMP adaptation	512	13.75 / 24.04	5.00 / 14.39	3.75 / 6.86
LLM-rank	512	8.75 / 19.30	5.00 / 7.25	2.50 / 6.48
Citation-first	512	8.75 / 19.34	5.00 / 7.25	1.25 / 5.46
Dependency-first	512	8.75 / 19.34	5.00 / 7.25	1.25 / 5.46
Producer-direct	4096†	6.25 / 15.82	0.00 / 3.28	1.25 / 5.95
Full-exec-reader	4096	16.25 / 24.34	5.00 / 10.00	5.00 / 10.29
Question-only	0	5.00 / 9.84	1.25 / 3.69	0.00 / 2.97

G.2 TRAINING-ONLY PRODUCER SIMPLIFICATION

On the 30-question Qwen training panel, dependency-first packing at 256 tokens answers 10/30 exactly (F1 39.72), versus 11/30 (F1 40.21) for RECOMP. The producer accounts for 95.3% of dependency-first packing's summed component time. We test two shorter producer interfaces on this training panel.

Candidate v1 replaces the generated dependency graph and ranked list with at most eight source-anchored subject–relation–object links and reduces the producer completion cap from 1024 to 512 tokens. The packer protects all selected source records, as in citation-first packing. The reader receives only original source text. V1 rejects an output if a generated subject or object does not occur literally in its cited sentence. This checks literal spans only.

On the same training panel, v1 lowers EM to 8/30 and F1 to 31.18. It produces 14 accepted outputs, eight explicit abstentions, seven lexical-span failures, and one answer without links. Producer completion tokens fall from 12,579 to 3,643 (71.04%), but the full 256-token policy's prompt-plus-completion tokens fall only from 134,174 to 125,265 (6.64%). Mean allocated component time falls from 8.672 to 3.664 seconds across the two sessions. V1 fails the quality and interface criteria, so it is not advanced beyond the training evaluation.

V2 uniformly retains structurally admissible source IDs and treats entity string matching as a diagnostic, replaying v1 producer outputs and costs. The seven lexical failures include phrase variation and misbound annotations. V2 retains the cited source text; readers never receive generated links. Other failure handling is unchanged. All 90 training cells, cache matching, and advancement rules are fixed before new calls. Advancement requires EM/F1 at least as high as the original 256-token baseline, at most two non-abstention source/structure failures, at most 12 abstentions, and at least 25% fewer producer completion tokens. V2 completes 90 cells with 14 new reader calls and otherwise reused outputs. EM is 10/30, but F1 is 38.16, below 39.72. It meets every selection criterion except F1. The complete 256-token policy uses 125,300 tokens and mean component time 3.699 seconds, versus 134,174 tokens and 8.672 seconds originally. These timings come from separate training sessions. Neither candidate advances to evaluation; both primary model configurations retain the original producer.

Recorded development and initialization costs. Table 52 counts new physical calls in every recorded training run, including the interface pilot. Historical producer or reader calls are counted only at their first execution. Across these runs, 494 calls consume 559,270 generator tokens, with no logged failed calls. Their recorded successful model-call durations sum to 491.254 seconds; client initialization totals 3.802 seconds. These quantities exclude ordinary CPU preprocessing, downloads, hashing, human work, and unrecorded probes.

Table 52: Recorded training-development model usage. P/R count new producer and reader calls. Tokens include prompts and completions. V2 replays prior producer outputs and only charges its 14 new reader calls here.

Run	New P/R	Prompt	Completion	Total
Initial interface pilot (6 questions)	6/84	76,144	2,479	78,623
Final training baseline (30 questions)	30/302	329,990	15,850	345,840
Compact producer v1	30/28	124,191	3,901	128,092
Citation retention v2	0/14	6,489	226	6,715
Total	66/428	536,814	22,456	559,270

GPU service startup is shared across runs and distinct from client loading. Launch-to-ready measurements are 137.522 seconds for the Mistral evaluation service and 43.372 seconds for the Qwen service restored for v2. The original Qwen service startup was not measured. Service flags and logs are retained with the run artifacts. Policy timings exclude these one-time startup costs.

H EVIDENCE SELECTION AND CONTROLS FOR PROGRAM REUSE

These development studies examine program construction from historical sources and sentence selection under a reader token limit. Construction methods share the same relation plan; selection methods use current evidence.

H.1 PROGRAM CONSTRUCTION WITH RECORDED SOURCES

The admission experiment checks historical support for the binder's complete predicted program. Here we evaluate program construction separately. Every method receives the same frozen question-only binder plan, public relation registry, current source sentences, tokenizer, resolver and verifier. Construction uses no reference topology, answers, or support annotations. The no-bank arm reconstructs the plan from public primitives; Whole copies a matching complete historical program; Compose copies historical steps with the required roles and assembles an executable intermediate representation. Each candidate then runs retrieval and the V7 resolver independently, without an execution cache.

A candidate counts as cross-source only when its selected steps come from at least two historical examples and no single available program contains every required step with the correct role. Each step records its source example, excluding cases where a single program would suffice. The nine natural source examples retain their archived programs and source records.

Here $B_{\text{exec}} = B_{\text{emit}} = 512$ evidence tokens, with a common 64-operation construction cap. No-bank has no Date filter. The separate execution/delivery budget study uses 512/256. Acceptance requires the Full verifier to pass. Joint validity also requires exact answer, reference-route and terminal-type agreement, exact coverage of supporting-fact coordinates, and nonempty supported terminal coordinates. We reuse natural reference scores only when the rerun program, proof hash and acceptance exactly match the Direct execution; other cases require separate scoring.

Table 53: Actual construction and execution on all 120 previously opened natural questions. Cross-source counts use the strict donor criterion above. All methods receive the same relation plan.

Constructor	Candidates	Accepted	Answer-correct	Joint-valid	Cross-source
No-bank rebuild	79	14	13	12	0
Whole	47	9	9	8	0
Compose	63	13	13	12	16

Four of Compose's 16 cross-source candidates are accepted. Against Whole, Compose has four joint-valid wins, zero losses and 116 ties. Against no-bank, it has zero wins, zero losses and 120 ties. Every one of its 63 constructed programs is also constructed by no-bank, with identical per-question main resolver calls and construction-operation counts. Compose avoids one incorrect no-bank acceptance; the Date rule also excludes this case.

We also evaluate the complete $3 \times 3 \times 2$ synthetic design: composer/performer/director transitions, birth-date/death-date/birth-place terminals, and present/missing terminal evidence. Three warm programs execute before their steps enter the synthetic program library. All 18 targets are retained. No-bank, Whole and Compose construct 18, six and eight candidates and solve nine, three and four targets, respectively. This fixture tests execution sensitivity to missing terminal evidence.

Source interventions and operation accounting. For each panel we evaluate the full and empty program libraries, remove each source example in turn, remove all sources for each route in turn, and use an equally sized library of unrelated synthetic programs. These yield 5,886 paired cells and 2,688 actual target-candidate executions, plus three synthetic warm executions. No model calls are made. Removing the sole composer donor lowers natural Compose from 63 candidates, 13 accepts and 12 joint-valid accepts to 31, five and five. Removing any of the eight individually redundant donors leaves the outcome with the full library unchanged. No-bank gives the same result under every library intervention. The interventions change admission through the library while holding the shared executor fixed.

Construction uses 515/340/483 algorithmic operations and 98/56/77 main-resolver calls for no-bank/Whole/Compose. The construction cap does not bind. Lower totals reflect fewer attempted candidates; work on common candidates is identical. Index construction separately parses nine historical entries and indexes 18 steps. These counts exclude historical acquisition, binder inference, fallback and diagnostic Replay parsing.

Comparison with construction from primitives. Assembly from recorded source programs increases joint-valid coverage from eight to 12 questions relative to whole-route reuse. Construction from the same public primitives also solves those 12. Evaluating reuse of local procedures, bindings, or search strategies requires a replanning control with the same history and total budget.

H.2 Sentence selection on the training panel

The producer often selects an answer sentence without all connecting evidence. V3 selects original sentences from the same admitted evidence, without generating an answer or graph. Its fixed objective combines reciprocal-rank relevance from BM25, BGE and RECOMP, query-term coverage, diminishing relevance gains within a paragraph, and a penalty for repeatedly selecting from one paragraph. A lexical connection term rewards sentences that name other selected paragraphs' titles, including paths from titles mentioned in the question. Matching respects token boundaries and excludes ambiguous titles. The lexical links indicate title mentions rather than verified relations or entity identity.

Both V3 and its diverse-rank ablation use all admitted candidates, the same 12-state beam, at most eight sentences, identical token counting and canonical output order. The ablation disables only the connection term. Neither sees reference supports, answers, decompositions, producer answers or historical programs. Every delivered line must be an exact original source record. All 120 packets and reader prompts (30 training questions, two variants and two budgets), source hashes and selection criteria are committed before scoring. The results are exploratory because the panel was used for development.

Table 54: Training-only paragraph-support diagnostics on all 30 questions. Complete means at least one selected sentence from every annotated support paragraph; it does not establish sentence sufficiency or correct answers. Precision and recall are per-question macro averages in percent.

Budget	Policy	Complete	Precision	Recall	Mean tokens
256	Dependency-first packing	7/30	48.78	62.22	251.9
256	RECOMP	5/30	34.11	48.61	249.7
256	Diverse-rank control	11/30	32.39	71.94	253.2
256	V3 bridge-cover	13/30	34.98	75.56	252.8
512	Dependency-first packing	13/30	37.55	71.94	506.0
512	RECOMP	8/30	24.81	58.06	506.6
512	Diverse-rank control	12/30	26.67	74.44	393.8
512	V3 bridge-cover	18/30	32.20	85.28	401.7

At the primary 256-token budget, V3 increases complete paragraph retention from the original policy's seven questions to 13, but the matched control already reaches 11. Its gain over that control is two questions, below the prespecified threshold of three; the other screens (higher recall, at most five percentage points of precision loss, and no loss of original complete retention) pass. V3 therefore fails the primary selection criterion; the 512-token result is secondary. Compared with dependency-first packing, the higher recall comes with lower precision (34.98 versus 48.78).

V3 and the ablation emit identical packets on 17/30 questions at 256 and 14/30 at 512. Their 120 cells define 89 distinct reader requests, none matching a previously executed training request exactly. This run evaluates evidence selection only; reader EM/F1 has not been measured.

The 120 CPU subset searches take 197.85 seconds. Each policy also uses all three rankers, whose historical time averages 6.085 seconds per question. The timings cover subset search and ranking, excluding common admission and reader inference.