TRUST IS A PROCESS, NOT JUST A SCORE: AGENTIC VERIFICATION FOR TEST-TIME SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

While test-time scaling improves large language models (LLMs) by generating multiple candidates, verification is often treated as a terminal scoring problem, separate from the acquisition and maintenance of evidence. In this paper, we explore verification as an interleaved process of gathering observations and updating a validity-aware state: execution probes, knowledge-base traversals and model adjudication distinguish candidates, while the state tracks dependence, supersession, disclosure and repair. We instantiate this approach, named GEVA (Governed Evidence Verification and Audit), as a deterministic state–action–observation loop over a typed claim–evidence graph, with certificate-based audits governing evidence producers. Across seven regimes spanning code, text-to-SQL, knowledge-graph question answering and multiple-choice reasoning, GEVA improves over greedy decoding throughout. On MetaQA-3hop, it reaches 42.70%, 11.10 points above common-pool Fixed V; matched-evidence controls examine selector choice separately from access to executed path evidence. On a disjoint held-out split, producer audit restores 4.38 accuracy points by withdrawing a harmful judge. Event replay further reproduces full recomputation with nearly $8\times$ fewer claim assessments. These results demonstrate the benefits of jointly acquiring, maintaining and governing verification evidence.

1 INTRODUCTION

Large language models (LLMs) scale inference by sampling and selecting candidates (Snell et al., 2025); selection also filters self-improvement data and verifiable rewards (DeepSeek-AI, 2025; Setlur et al., 2025; Liao et al., 2026). More candidates create opportunities, but exploiting them depends on what the verifier observes. A larger pool may contain a correct answer without supplying a reliable way to find it. Additional verification computation must therefore address two questions: which observation could distinguish the remaining candidates, and how should that observation change the current decision? We study these questions in frozen-policy inference, where verification can evolve while the generator's parameters remain fixed.

Verification = evidence acquisition + evidence management. An execution probe, knowledge-base (KB) traversal or evaluator can supply information missing from the current verifier signal. Acquisition expands the available evidence; management tracks its applicability, dependence and permitted reuse, informing both selection and acceptance. Shared-source checks require dependence accounting, a traversal may supersede a guess, and repair calls for fresh validation beyond the checks already disclosed to the solver.

Consider a candidate program that passes a public test and receives support from agreement with other candidates. A private probe then reveals an exception, prompting a revision. The public test must be checked on the revised program; the old agreement was measured on a different implementation; and the disclosed probe now serves as a regression check rather than fresh independent support. A single accumulated score leaves these distinctions implicit. A verification state makes the dependencies and version boundaries explicit, so that subsequent checks can be interpreted relative to what the solver has already seen and which candidate is currently being assessed.

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.

Evidence management also concerns the source of an observation. Repeating one producer's verdict is different from obtaining a check with another failure mode, and a producer useful on one task may be misleading on another. Where an external resource supplies a premise-qualified certificate of wrongness, its evidence can test a producer's preferences without reading benchmark outcomes. This links within-instance verification to cross-instance producer audit: the former updates candidate support, while the latter determines which sources remain available to the controller.

Governed Evidence Verification and Audit (GEVA) implements this loop with a deterministic controller and a typed claim–evidence state (Figure 1). Acceptance gates guide probes, paths, adjudication or repair; observations update support, risk and obligation coverage. Provenance connects *state consistency*, *selective decisions* and *producer governance*. The state contracts are representation-independent, with a graph providing our concrete implementation.

Different tasks expose complementary roles: code probes support repair, Spider database checks inform stopping, MetaQA paths replace relation guesses, and KQA Pro introduces model-based adjudication. Matched controls separate acquisition from selection, while an instance-disjoint certificate audit restores 43.50% from 39.13% by withdrawing a harmful judge. Together, these studies examine how evidence acquisition, decision rules and producer availability shape the verification process.

Our contributions are:

- A verification rollout combining active evidence acquisition with a provenance-aware state, instantiated as a typed claim–evidence graph.
- Evidence-use contracts and an event-sourced extension for replayable revisions, explicit validity propagation and incremental assessment, evaluated separately from the legacy policy.
- A deterministic acquisition controller and a label-free producer audit based on premise-qualified certificates rather than the judge's own confidence.
- Seven-regime, two-policy evaluation with matched-evidence and repair controls, lifecycle diagnostics, instance-disjoint audit, flat selectors and nested-pool scaling, characterising the roles of acquisition, evidence management and producer availability.

2 Related Work

Verifiers for rollout scaling. Self-consistency, best-of-N, and search rank candidates with outcome or process reward models (Lightman et al., 2024; Luo et al., 2024; Wang et al., 2024; Setlur et al., 2025), while compute-optimal inference adapts search to difficulty (Snell et al., 2025). Reinforcement learning with verifiable rewards uses verifier outputs for optimisation (DeepSeek-AI, 2025; Liao et al., 2026). We study inference-time verification: how to represent, acquire and audit evidence about completed rollouts.

Tool feedback and adaptive verification. CRITIC uses external tools to critique and revise model outputs (Gou et al., 2024); LEVER learns to rank generated programs from their executions (Ni et al., 2023). Knowledge graph (KG) agents traverse structured resources during generation (Sun et al., 2024; Ma et al., 2025; Lin et al., 2025), while Search-R1 and ToolRL train models to invoke search engines or general tools through reinforcement learning (Jin et al., 2025; Qian et al., 2025). Adaptive Generate-Rank-Verify allocates costly verifier queries under assumptions on score-conditioned success (Dughmi et al., 2026), while utility-guided orchestration trades estimated gains against cost and redundancy (Liu et al., 2026). Truth-maintenance systems already track assumptions and retract dependent beliefs (Doyle, 1979; de Kleer, 1986). Our extension applies explicit validity contracts to verifier observations, separately from correlation ancestry, with replayable updates and cost controls; dependency propagation or graph storage alone is not claimed as new.

Selective prediction. Reject-option classification (Chow, 1970; El-Yaniv & Wiener, 2010), selective classifiers (Geifman & El-Yaniv, 2017) and conformal prediction (Angelopoulos & Bates, 2023) motivate risk–coverage evaluation and the area under the risk–coverage curve (AURC). We use a Wilson-derived score (Wilson, 1927) with dependence-adjusted counts (Kish, 1965) to order and gate claims. Its empirical ranking quality is evaluated with complete risk–coverage curves; we do not interpret it as a calibrated error probability.

3 METHOD

3.1 VERIFICATION SETUP

A frozen policy produces an initial pool C_x from public input x, combining a greedy decode with N samples. Verification queries benchmark probes, knowledge bases and tools through $\mathcal{R}_V = (G_0, a_0, e_1, G_1, \ldots, G_T)$ (Figure 1). It emits a claim and separately decides whether to accept it. Challenges may add repaired claim versions; policy parameters remain frozen.

Evidence has three visibility tiers. Public T0 evidence may be shown to the solver and never counts as independent support. Verifier-private T1 evidence can support claims and identify counterexamples. Disclosure writes a consumes edge, removing independent support while preserving contradictions. Sealed T2 benchmark outcomes never enter the graph: only the meta-evaluator reads them after decisions are frozen, and a runtime guard rejects sealed keys passed to producers.

Because code and KG benchmarks score an abstention as zero, the policy always emits its best-supported claim. Figure 1's "Unaccepted" branch withholds acceptance, while an accepted output includes evidence and provenance and satisfies the gate under the available checks.

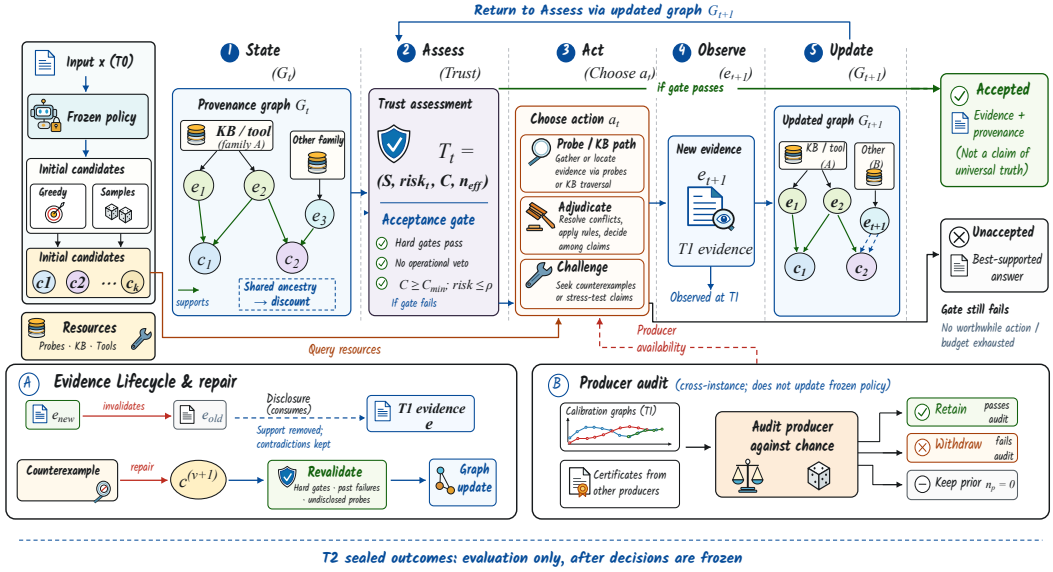


Figure 1: GEVA verification rollout over provenance graphs with a frozen generation policy. **(A)** Evidence lifecycle and versioned repair. **(B)** Cross-instance producer audit. Decisions use T0/T1 evidence; sealed T2 outcomes are reserved for evaluation.

3.2 EVIDENCE MANAGEMENT: A PROVENANCE-AWARE STATE

Each instance holds versioned claims, observations, artifacts and tool runs linked by support, contradiction, provenance and lifecycle relations. Evidence records its producer, family, tool version, seed, ancestors, visibility, strength, polarity and invocation-batch ID. Diagnostic rows retain veto and provenance roles without adding statistical trials. Five source families are distinguished: TRUSTED_RESOURCE, BENCH_INPUTS, DETERMINISTIC, SOLVER_FAMILY and CROSS_FAMILY (Appendices A and B). Coverage $C(c)$ counts supported obligations: specification, robustness, edge cases, consensus and purity. Shared ancestors discount statistical independence (Eq. 1).

Event-sourced extension. An opt-in journal records a baseline and ordered before/after revisions, preserving observations otherwise lost to overwrite. Explicit validity requirements are distinct from correlation ancestry. Replacement or revocation invalidates their dependent closure; only affected owner claims need their full trust statistics recomputed. A same-semantics flat implementation remains possible. Proposition 4 states the conditions for equivalence to full recomputation. This

new extension is evaluated separately in Appendix N.1; it does not alter the saved policies of Table 1. Recomputing a score is not re-executing a tool.

3.3 CERTIFICATES: WHEN AN OBSERVATION PROVES WRONGNESS

Certificates distinguish a lack of support from evidence of wrongness. The following propositions state two sufficient conditions for certifying a failure.

Proposition 1 (Execution certificate)**.** *Let the benchmark metric evaluate a candidate by executing it on every element of an instance table I owned by the harness, scoring an exception as failure. If the candidate raises an exception on some $i \in I$, it cannot attain a perfect metric value. The check requires no expected output.*

Proposition 2 (Deductive-KB certificate)**.** *Let the gold answer set of x be generated by instantiating a relation chain of exactly k edges in a knowledge base K. Then an answer unreachable by any exact-k-hop walk from the topic entity in K is not a gold answer.*

We reserve *sound certificate* for evidence whose declared source and semantic premises satisfy such conditions. It is represented separately from the *operational veto* used by ranking, acceptance and repair. The producer audit consumes only the former; it never infers soundness from post-hoc precision. Code probe failures remain operational vetoes but are ineligible for audit supervision, because their generated inputs are not the evaluator's exact invocations. Proposition 2 is applied to MetaQA's generating KB only after complete, non-truncated exact-hop walk reachability; KQA Pro's incomplete subgraph cannot certify absence, although paths already found can still supply positive evidence.

3.4 ASSESS: INDEPENDENCE-AWARE TRUST

Let E contain only active, undisclosed T1 evidence after canonical aggregation by producer-invocation batch. Public gates and diagnostic detail rows add no trials. Each batch pair is charged the larger of its family-correlation prior and ancestor Jaccard overlap; the mean $\bar{\rho}$ gives a design effect (Kish, 1965):

$$\mathrm{IndepScore}(E) = \frac{|E|}{1 + (|E| - 1)\,\bar{\rho}}, \qquad \bar{\rho} = \operatorname*{mean}_{i<j} \max\big(\mathrm{FamCorr}(f_i, f_j),\, \mathrm{Jac}(\mathrm{anc}_i, \mathrm{anc}_j)\big). \tag{1}$$

Intra-family residual correlations are fixed semantic priors, never tuned against sealed labels: TRUSTED_RESOURCE .25, DETERMINISTIC .35, CROSS_FAMILY .40, BENCH_INPUTS .60, SOLVER_FAMILY .85. The ranking score accumulates support *per family* under concave saturation, while contradictions do not saturate:

$$S(c) = \sum_f w_f \Big(1 - e^{-\mathrm{mass}_f/\tau}\Big) - \sum_f w_f \,\mathrm{contra}_f, \tag{2}$$

where positive mass is the sum of clipped evidence strengths, contradiction mass counts negative observations, and $\tau = 2$. All positive families saturate; a bounded pass-rate observation and repeated same-family observations contribute at different scales. Appendix A specifies the family weights and cross-family correlations.

Let $(\hat{n}_{\text{pass}}, \hat{n}_{\text{tot}})$ denote the family-weighted atomic check counts scaled by the design-effect ratio $\mathrm{IndepScore}(E)/|E|$. The *evidence-conditioned risk score* is then

$$\mathrm{risk}_t(c) \equiv \mathrm{RiskScore}_\alpha(c \mid G_t) = 1 - \mathrm{WilsonLower}\big(\hat{n}_{\text{pass}}, \hat{n}_{\text{tot}}, \alpha\big). \tag{3}$$

For each claim, the trust state is $T_t(c) = (S(c), \mathrm{risk}_t(c), C(c), n_{\text{eff}}(c))$, where $\mathrm{risk}_t(c)$ is Eq. 3 and n_{eff} denotes the independence-adjusted evidence count. Figure 1 presents these quantities as components of the trust state: this is an evidence summary, not a calibrated probability of correctness. Equation 3 uses the Wilson interval (Wilson, 1927) extended to fractional counts, with $\hat{n}_{\text{tot}} = 0$ mapped to maximal risk ("no evidence" is not "no risk"). A claim is *acceptable* iff every mandatory hard gate passed, *no operational veto* remains unresolved, $C(c) \geq C_{\min}$, and Eq. 3 $\leq \rho$. Only flagged negatives veto, since vetoing on any negative signal fires on most correct Spider queries: empty result sets and ungrounded column references are common on valid statements in Structured

Query Language (SQL). Claims are ranked lexicographically by (hard gates, no operational veto, S, $-\mathrm{RiskScore}$, coverage, draw order); no fitted parameter and no label enters it.

The design effect exactly matches the variance of the mean for exchangeable Bernoulli checks (Proposition 3). Applied to heterogeneous, weighted evidence, Eq. 3 is an operational risk score, not a bound on benchmark error probability. The controlled simulation in Appendix I yields mean interval coverage .803 without adjustment and .983 with the correct correlation, against nominal .90. This simulation does not calibrate task correctness. We use $\rho = .35$ as a fixed operating threshold and report its empirical acceptance and conditional risk (Table 8).

3.5 Evidence Acquisition, State Updates, and Producer Audit

If the incumbent is not acceptable, the controller chooses among affordable actions from available producers. The action groups in Figure 1 are probe or KB-path acquisition, adjudication, and challenge. The next action maximises an estimated verification benefit per unit cost,

$$a_t = \arg\max_a \frac{\widehat{\Delta}(G_t, a) + \beta\big(1 - \mathrm{GraphCut}(G_t)\big) - \gamma\,\mathrm{CorrPenalty}(G_t, a)}{\mathrm{Cost}(a)}, \tag{4}$$

where $\mathrm{GraphCut}$ combines support spread and behavioural partitioning. $\mathrm{CorrPenalty}$ is the incumbent's normalised family-concentration index, except that cross-family acquisition receives zero penalty. The benefit $\widehat{\Delta}$ uses interval-width contraction and missing-obligation credits; it is a deterministic action score rather than an expectation under a learned outcome model. Appendix C gives the formulas and constants. Withdrawn producers are removed from the available actions.

Action-utility costs are 1 unit for a probe or property batch, 2 for KB traversal, 4 for an LLM probe, 6 for adjudication and 12 for challenge. Reported replay units accumulate acquisition charges only: terminal challenge requests, repair execution, initial generation, preprocessing and audit are excluded. These counters are not end-to-end costs; a constant batch charge can hide increasing physical work.

Observe and update. An action attaches new T1 evidence e_{t+1} to its producer, ancestors and claims, forming G_{t+1}. ACQUIRE-KB-PATH enumerates simple exact-k-hop paths for alignment; a separate exact-hop walk search, allowing repeated nodes, checks absence. An executed observation replaces the relation guess and invalidates support derived from that guess rather than being counted alongside it. Each path observation has $n_{\mathrm{checks}} = 1$; truncated searches cannot certify absence. Adjudication asks another model to select an existing candidate or NONE using public task input. It adds cross-family evidence, not a new answer. The updated graph is reassessed until acceptance, exhaustion of useful actions or the step limit; the final incumbent is always emitted.

Self-audit: scoring a producer against the graph's own certificates. Family weights express source priors; a cross-instance audit tests individual producers against certificates from other producers. Panel B of Figure 1 uses T1 calibration graphs to determine action availability without updating the generation policy.

Definition 1 (Unforced error)**.** *A producer p commits an unforced error on an instance when (i) at least one claim carries a sound certificate issued by a producer other than p, (ii) at least one claim carries none, and (iii) p's maximal positive support is on a certified-wrong claim. Condition (ii) is what makes it unforced: when every claim is certified wrong, no choice is better than another.*

The audit computes this comparison only from evidence whose certificate source and premises are explicitly recorded. Operational vetoes without this qualification never enter W.

For n_p eligible instances with a positive producer preference, let k_p count unforced errors and $q_i = |W_i|/|C_i|$. We compare with independent uniform claim-ID choices on these fixed instances. The revised retention rule uses the exact lower tail of their Poisson–binomial error count:

$$p_p = \Pr\left[\sum_{i=1}^{n_p} B_i \le k_p\right] \le \alpha, \qquad B_i \overset{\mathrm{ind}}{\sim} \mathrm{Bernoulli}(q_i). \tag{5}$$

For $n_p > 0$, failure to pass triggers withdrawal by policy, not a finding that the producer is significantly worse or useless. With $n_p = 0$, it keeps its prior. The test concerns the declared random-

picker reference, not calibrated task accuracy. Replacing the former Wilson/mean-chance rule leaves both MetaQA judge decisions unchanged; Appendix H reports the exact tests and their scope.

Counterexample-driven challenge with versioned re-validation. A challenge supplies the solver with a counterexample against the incumbent. It can be selected by action utility when the incumbent has an operational veto; when all claims have such flags, it is also the controller's fallback. These conditions read graph state only. The all-claims condition is analysed separately from the deployed action counts in Appendix G. Panel A of Figure 1 separates supersession from disclosure: e_{new} `invalidates` superseded e_{old}, whereas `consumes` marks disclosed evidence, removing its independent support while preserving its contradictions. A repair creates $c^{(v+1)}$ with the incumbent as its `parent`. It must re-pass every hard gate and historical counterexample, and is checked on undisclosed probes held out from the start before graph update and reassessment (Appendix G).

4 Experiments

Table 1: **Single-decode and common-pool accuracy (%).** Greedy uses one decode; Majority and Fixed V select from greedy plus eight samples. GEVA uses that initial pool with its reported tools and code repairs. Δ and paired 95% CI compare GEVA with common-pool Fixed V; "–" denotes unavailable greedy verifier scores.

Policy	Regime	Greedy	Majority	Fixed V	GEVA	Δ	95% CI
9B	MBPP+	67.72	68.25	76.46	79.37	+2.91	[+1.06, +4.76]
9B	HumanEval+	80.49	81.10	87.20	87.80	+0.61	[0.00, +1.83]
9B	Spider	67.60	67.70	68.09	69.83	+1.74	[+0.48, +3.00]
9B	MetaQA-1hop	24.70	25.30	33.80	33.70	−0.10	[−0.40, +0.20]
9B	MetaQA-3hop	33.10	32.90	31.60	42.70	+11.10	[+9.00, +13.30]
9B	KQA Pro	55.10	54.10	–	61.90	–	–
9B	MathQA-Geo	71.40	79.38	87.80	86.92	−0.89	[−3.10, +1.33]
27B	MBPP+	78.04	78.31	81.48	82.28	+0.79	[0.00, +1.85]
27B	HumanEval+	89.02	89.02	90.85	90.85	0.00	[0.00, 0.00]
27B	Spider	72.05	72.24	72.24	72.92	+0.68	[−0.19, +1.64]
27B	MetaQA-1hop	33.50	32.90	40.90	40.80	−0.10	[−0.50, +0.20]
27B	MetaQA-3hop	36.00	35.40	36.80	44.10	+7.30	[+5.60, +9.00]
27B	KQA Pro	62.80	62.10	–	67.10	–	–

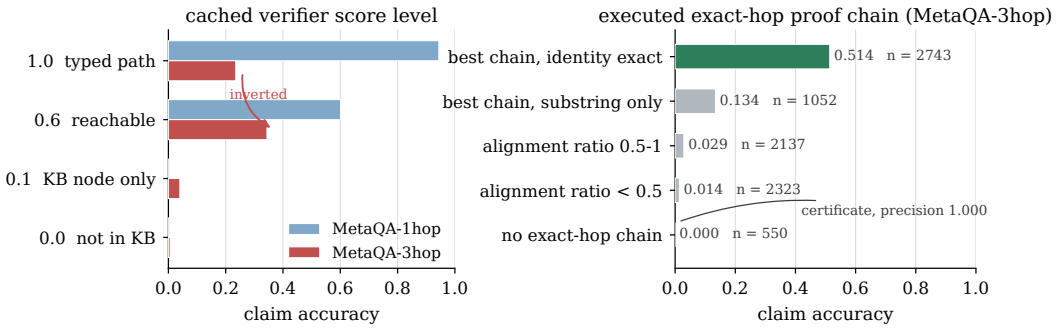


Figure 2: Post-hoc calibration of cached verifier levels (left) and executed exact-hop proof-chain evidence (right) on MetaQA. The cached levels invert on 3-hop questions, whereas executed alignment adds information unavailable to re-weighting; the bottom proof-chain bucket is the premise-qualified absence certificate. No pipeline threshold reads these outcome labels.

4.1 Experimental Setup

We evaluate seven regimes using frozen Qwen3.5-9B pools (Qwen Team, 2026a): one greedy decode plus eight temperature-0.8 samples. MBPP+ (378) and HumanEval+ (164) use visible base tests

and sealed full-plus correctness (Chen et al., 2021; Austin et al., 2021; Liu et al., 2023); Spider dev (1,034) uses test-suite execution accuracy (Yu et al., 2018; Zhong et al., 2020); MetaQA-1hop/3hop use 1,000 questions each and the official KB (Zhang et al., 2018); KQA Pro uses 1,000 validation questions (Cao et al., 2022); MathQA-Geo contains 451 geometry problems (Amini et al., 2019). MetaQA's archived scorer accepts normalised exact or either-direction substring matches; strict entity matching lowers 1-hop/3-hop accuracy by $2.6/1.0$ points (Appendix M). KQA Pro scores the first normalised option letter. Table 1 aligns initial candidates, preserving multiplicity and greedy-first ties; tools and code repairs remain distinct resources.

We report emitted-answer accuracy, accepted-and-wrong frequency (joint false-trust), AURC over risk-ordered outputs, and acquisition charges (§3.5). Conditional accepted error divides false-trust by acceptance and is undefined at zero acceptance. Changing the controller threshold can change outputs and hence AURC.

Defaults are $\alpha = .10, C_{\min} = .60, \beta = .25, \gamma = .30, \rho = .35$; tools and step limits are in Appendix J. Paired percentile-bootstrap intervals use 10,000 replicates; two-sided p uses the exact paired sign test. Both condition on saved pools, excluding generation-seed variance. Audit calibrates on the first 20% of sorted instances using T1 evidence, then freezes producer availability on the remaining 80% (Appendix H).

4.2 Selection Accuracy on Seven Regimes

Table 1 compares single-decode Greedy, common-pool Majority and Fixed V, and GEVA. KQA Pro includes a judge and MathQA-Geo does not; missing verifier scores are marked in the table. Code results include repair: MBPP+ rises from 77.78 to 79.37 ($+1.59$, 95% CI $[+0.53, +2.91]$, $p = .031$; six improved, none harmed), while HumanEval+ is unchanged. Without repairs, MBPP+'s margin over common-pool Fixed V is $+1.32$ ($p = .125$). The matched repair control is detailed in Appendix M.

Certificate eligibility in MathQA-Geo. GEVA reaches 86.92 versus Fixed V's 87.80 ($p = .541$). The negative derivation signal has precision .958, but faulty derivations can accompany correct letters; the contract therefore retains graded evidence without an operational veto or audit certificate (Appendix F).

A second base policy. Qwen3.6-27B (Qwen Team, 2026b) provides another saved-pool evaluation within the Qwen model family. Its common-pool MetaQA-3hop margin is $+7.30$ points; MetaQA-1hop is -0.10, HumanEval+ matches Fixed V, and KQA Pro still lacks greedy verifier scores (Table 1). A separate Mistral-7B-labelled saved-pool check reuses the frozen MetaQA adapter and controller. Strict entity accuracy is 24.20 versus 19.70 for common-pool Fixed V ($+4.50$, 95% CI $[+3.20, +5.90]$). This serves as a supporting one-domain archived-pool compatibility check, separate from the matched lifecycle controls (Appendix K).

4.3 Separating Evidence Acquisition from Evidence Management

Evidence acquisition and selection. Executed KB traversal adds path and alignment information to MetaQA's cached relation signal (Figure 2). The fixed selector achieves 41.30% versus GEVA's 42.70% (GEVA minus selector: $+1.40$ points, 95% CI $[+0.53, +2.27]$, $p = .002577$). This comparison examines selector choice given the same final path evidence, rather than graph storage alone. Complete acquisition, repair, lifecycle, gate and audit controls appear in Appendix M, Table 20.

State-update consistency and local recomputation. Across 100 live KB traversals, 3,108 events include 1,056 invalidations. Incremental replay reproduces every full-recomputation state with 3,708 rather than 28,872 claim assessments. The evaluation distinguishes per-event updates from action-boundary batching and includes a same-semantics flat implementation. Controlled SQLite reruns measure how the affected dependency set determines revalidation work. Table 22 reports all schedules and timings; these experiments evaluate state updates and computation, separately from task accuracy.

Decision gates and producer availability. On Spider, the full policy gives 24.47% joint false-trust and AURC .2086, compared with 32.30% and .3383 under relaxed gates; acquisition and acceptance both participate. Table 2 isolates producer availability on 800 instances disjoint from audit calibration: withdrawing the judge recovers 4.38 points (95% CI $[+2.50, +6.25]$), matching the no-judge reference rather than surpassing it. Accepted conditional error is unchanged. Certificate eligibility and keep-prior cases are detailed in Appendix H.

Table 2: **Producer governance on held-out instances.** MetaQA-3hop: the label-free audit decision from 200 calibration cases is frozen on the remaining 800. Accuracy, acceptance, joint false-trust (FT) and conditional accepted risk are %; units exclude audit, generation and terminal challenges. Audit restores the no-judge result, not a new accuracy ceiling or lower accepted risk.

Policy	Acc.	Accept	FT	Accepted risk	AURC	Units
No judge	43.50	19.63	7.50	38.22	.4447	6.97
Unaudited judge	39.13	19.63	7.50	38.22	.4572	11.76
Audited judge	43.50	19.63	7.50	38.22	.4447	6.97

Across the full task suites, MetaQA-3hop accepts 19.2% with 39.06% conditional error, whereas KQA Pro and MathQA-Geo accept none at $\rho = .35$ (Table 8). Conditional error is undefined at zero acceptance.

4.4 Flat Selectors under Limited Supervision

We compare GEVA with scalar, logistic regression (LR) and gradient-boosted decision tree (GBDT) selectors fitted to a flat ledger. They receive the fully revealed harness view and target-regime labels; GEVA instead acquires selected evidence through its graph and uses no target-domain fitting. At the largest Figure 3 budget, fitted means remain below the matched GEVA reference on MBPP+ (77.48 vs 78.28) and Spider (69.57 vs 69.86), while MetaQA-3hop LR matches 42.85. HumanEval+ and KQA Pro already have fitted selectors above the matched reference at eight labels; MetaQA-1hop's scalar matches at eight and its multi-feature selectors later cross. These learning curves quantify the target-label budgets at which fitted selectors match or exceed the GEVA reference, complementing the evaluation of label-free routing and state updates.

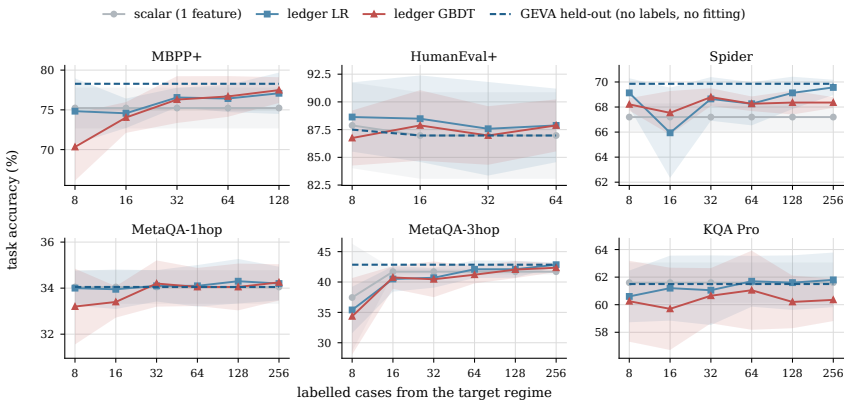


Figure 3: Learning to match GEVA with target-regime labels: mean ±1 SD on seed-specific fixed 40% held-out splits. HumanEval+ at n=8 has four valid draws (one single-class draw omitted); all other points have five. Dashed matched-draw references are 87.50% then 86.97% on HumanEval+, not Table 1's full-suite values. KQA Pro includes the judge; MBPP+ ends at 128.

The archived leave-one-regime-out snapshot shows fitted selectors losing up to 10.10 points (Table 7); because it predates the corrected risk accounting, it is used only as a transfer diagnostic, not a current safety comparison.

Table 3: Strict nested prefixes of one saved $N{=}32$ master pool per regime: $S_8 \subset S_{16} \subset S_{32}$, common greedy, $\rho = .35$, no optional judges. Accuracy is in %; replay units exclude generation, preprocessing, audit and terminal challenge requests. Bold marks the best GEVA accuracy within each row, not significance.

	Greedy			Fixed V			GEVA			Oracle			Replay units		
Regime	8	16	32	8	16	32	8	16	32	8	16	32	8	16	32
MetaQA-1hop	24.90	24.90	24.90	34.00	36.90	39.50	34.30	36.90	**39.60**	34.90	37.70	40.40	4.06	4.16	4.25
MetaQA-3hop	33.10	33.10	33.10	28.60	27.20	27.00	41.40	44.10	**48.30**	46.90	51.90	57.00	6.95	6.95	6.96
Spider dev	67.79	67.79	67.79	65.57	65.96	66.05	**70.89**	70.41	70.60	79.98	82.21	83.27	1.52	1.29	1.26
KQA Pro	55.10	55.10	55.10	55.80	57.80	58.50	59.50	61.30	**61.90**	78.30	85.20	88.40	5.00	5.00	5.00
MathQA-Geo	69.40	69.40	69.40	88.03	88.91	89.80	86.70	86.47	**87.80**	93.79	96.01	97.12	4.98	4.99	5.00

4.5 TASK-DEPENDENT TRADE-OFFS AND POOL SCALING

Nested-pool gains from $N = 8$ to 32 are $+5.30/+6.90/+2.40$ points on MetaQA-1hop/3hop/KQA Pro; Spider (-0.29) and MathQA-Geo ($+1.11$) remain unresolved (Table 3; intervals in Appendix L). A new T1-only sweep of 25 settings per regime varies weights, correlations and ρ: accuracy moves by at most .60 points from default, while acceptance is more sensitive (Appendix L.1). These checks condition on saved resources.

Table 4 compares accuracy, recorded acquisition charges, AURC and joint false-trust. Adaptive acquisition lowers those charges in five of six regimes, although MetaQA-3hop decreases by only 0.03 unit after rounding; KQA Pro increases by 0.27. Charges exclude generation, preprocessing, audit, terminal challenges and repairs. Table 14 separately reports Mistral CPU phases, with original generation cost unavailable.

Table 4: **Accuracy–cost–risk trade-offs.** Full revelation (F) versus adaptive acquisition (A), $\rho = .35$. Acc. and joint false-trust (FT) are %. Replay units exclude generation, preprocessing, audit, terminal challenges and repairs.

	Full revelation (F)			Adaptive (A)			
Regime	Acc.	Units	AURC	Acc.	Units	AURC	FT (F→A)
MBPP+	79.10	4.63	.1522	79.37	1.38	.0954	8.99 → 12.17
HumanEval+	88.41	4.37	.0768	87.80	1.29	.0331	3.66 → 4.88
Spider	70.31	4.98	.1742	69.83	1.49	.2086	23.21 → 24.47
MetaQA-1hop	33.70	7.00	.3304	33.70	4.62	.3177	0.50 → 0.50
MetaQA-3hop	42.70	7.00	.4555	42.70	6.97	.4555	7.50 → 7.50
KQA Pro	61.90	8.76	.1931	61.90	9.03	.1931	0.00 → 0.00

Lifecycle validity and selection. Controlled MBPP+ cache revision changes 112 selections across 144 sibling substitutions on 36 problems (Appendix N.2). In a prospective 24-problem sequential-repair pilot with withheld public assertions, stale reuse leaves 16.7% of selected claims with contract-invalid active evidence; all version-aware views eliminate invalid reuse. GEVA changes 11 final selections relative to stale reuse and matches simple version caching at 37.5% accuracy. The pilot establishes validity-sensitive behavior rather than additional predictive utility (Appendix N.3).

5 CONCLUSION

GEVA connects active acquisition with explicit conditions for evidence use. Acquisition, repair, gates and audit show task-dependent benefits; event replay establishes update consistency and local recomputation. Lifecycle and repair controls demonstrate validity-sensitive behavior. These findings distinguish task-level gains from contract validity and motivate further closed-loop reliability and full-pipeline cost evaluation.

Artificial Intelligence (AI) Use Statement

An LLM assisted with manuscript drafting, substantive revision, copy-editing and research support, including literature discovery for the related-work discussion and brainstorming and stress-testing candidate framings of the verification problem. Most relevant to this, discussion with an LLM helped us articulate the analogy between generation and verification rollouts that organises the present framing. All conceptual contributions and experimental design decisions were made by the authors. Reported results are drawn from the experimental artifacts described in Appendix J. The authors are responsible for verifying the references, experimental claims and final wording.

Ethics Statement

The experiments use graphics processing unit (GPU)-backed model calls for candidate generation and model-based verification, alongside central processing unit (CPU) execution and symbolic checks. Verification-action accounting includes the verification actions acquired by the controller. All datasets used are public research benchmarks; no human subjects or private data are involved.

Reproducibility Statement

Appendix J lists the frozen schema, algorithms, per-domain producer tables, commands and expected artifact paths, including the four ablation variants of Appendix C.1. The bundle contains implementation code and key aggregate results, but not the public benchmark files, model weights, saved pools or complete per-instance traces. End-to-end reproduction requires those documented resources; CPU replay additionally requires the corresponding saved artifacts.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2357–2367, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245/.

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023. doi: 10.1561/2200000101. URL https://doi.org/10.1561/2200000101.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Shulin Cao, Jiaxin Shi, Liangming Pan, Lunyiu Nie, Yutong Xiang, Lei Hou, Juanzi Li, Bin He, and Hanwang Zhang. KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 6101–6119, 2022. doi: 10.18653/v1/2022.acl-long.422.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986. doi: 10.1016/0004-3702(86)90080-9.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Jon Doyle. A truth maintenance system. *Artificial Intelligence*, 12(3):231–272, 1979. doi: 10.1016/0004-3702(79)90008-0.

Shaddin Dughmi, Mahdi Haghifam, and Yusuf Hakan Kalayci. Adaptive generate-rank-verify: Inference-time search with costly verification. *arXiv preprint arXiv:2605.17609*, 2026. URL https://arxiv.org/abs/2605.17609.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11:1605–1641, 2010.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, 2017.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing, 2024. URL https://arxiv.org/abs/2305.11738.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O. Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-R1: Training LLMs to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Leslie Kish. *Survey Sampling*. John Wiley and Sons, 1965.

Ruotong Liao, Nikolai Röhrich, Xiaohan Wang, Yuhui Zhang, Yasaman Samadzadeh, Volker Tresp, and Serena Yeung-Levy. Tool verification for test-time reinforcement learning. *arXiv preprint arXiv:2603.02203*, 2026. URL https://arxiv.org/abs/2603.02203.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Junhong Lin, Shicheng Liu, Jinyeop Song, Song Wang, Julian Shun, and Yada Zhu. Efficient and transferable agentic knowledge graph RAG via reinforcement learning. *arXiv preprint arXiv:2509.26383*, 2025. URL https://arxiv.org/abs/2509.26383.

Boyan Liu, Gongming Zhao, and Hongli Xu. Utility-guided agent orchestration for efficient LLM tool use. *arXiv preprint arXiv:2603.19896*, 2026. URL https://arxiv.org/abs/2603.19896.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatGPT really correct? Rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024. URL https://arxiv.org/abs/2406.06592.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *International Conference on Learning Representations*, 2025. URL https://mlanthology.org/iclr/2025/ma2025iclr-thinkongraph/.

Mistral AI Team. Mistral-7B-Instruct-v0.3. Hugging Face model card, 2024. URL https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3.

Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-Tau Yih, Sida Wang, and Xi Victoria Lin. LEVER: Learning to verify language-to-code generation with execution. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26106–26128. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ni23b.html.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. ToolRL: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/97c5b2707228e7e3fb67e4ecc2e0e607-Abstract-Conference.html.

Qwen Team. Qwen3.5-9B. Hugging Face model card, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-9B.

Qwen Team. Qwen3.6-27B. Hugging Face model card, 2026b. URL https://huggingface.co/Qwen/Qwen3.6-27B.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *International Conference on Learning Representations*, 2025. URL https://mlanthology.org/iclr/2025/setlur2025iclr-rewarding/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/1b623663fd9b874366f3ce019fdfdd44-Paper-Conference.pdf.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 9426–9439, 2024. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, 2018. doi: 10.18653/v1/D18-1425.

Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. Variational reasoning for question answering with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.12057.

Ruiqi Zhong, Tao Yu, and Dan Klein. Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 396–411, 2020. doi: 10.18653/v1/2020.emnlp-main.29.

A Frozen Graph Schema

The schema serialises one JavaScript Object Notation (JSON) object per instance, so a graph can be persisted next to existing result artifacts and diffed across runs. Figure 1 shows the instance-level verification loop, its evidence lifecycle and repair operations (panel A), and the cross-instance producer audit (panel B).

Nodes.

- **Claim**: `claim_id`, `case_id`, `generator`, payload (code / SQL / answer string), `version`, `parent` (set by a repair), `origin` $\in$ {`greedy`, `pool`, `challenge_repair`}, and a content hash.
- **Evidence**: `producer`, `family`, `visibility`, `obligation`, `polarity` $\in \{-1, 0, +1\}$, `strength` $\in [0, 1]$, `n_checks`, `tool_version`, `seed`, `ancestors`, `consumed`, `invalidated`, `operational_veto`, `certificate_kind`, `certificate_source`, `certificate_premises_satisfied`, and a `detail` dictionary. Legacy `detail.definite` values are migrated by a fixed producer-contract map; they do not by themselves confer certificate status.
- **Artifact**: immutable probe tables, counterexamples, KB snapshots, each content-hashed so that two evidence items derived from the same artifact share an explicit ancestor key.
- **ToolRun**: a record of an executed verification action, carrying sandbox executions, LLM calls, tokens, and wall-clock.

Edge types. `supports`, `contradicts`, `derived_from`, `produced_by`, `consumes`, `invalidates`, `revalidates`, `depends_on`. The tuple is frozen: an unknown edge type raises rather than silently creating a new relation, so schema drift cannot be introduced by a later producer.

Obligations. `spec_conformance` (matches the visible specification examples), `robustness` (does not crash on in-distribution inputs), `edge_case` (survives adversarially generated corner cases), `behaviour_consensus` (agrees with independently checked peers), `purity` (deterministic, does not mutate its arguments). Coverage $C(c)$ is the fraction of obligations with at least one supporting path; `spec_conformance` is mandatory.

Visibility guard. `assert_no_sealed` recursively rejects any object reaching a producer that contains `plus_pass`, `plus_total`, `plus_error`, or `is_correct`. The guard exists because the defect of §G entered through a refactor rather than by design, and a mechanical check is the only defence that survives refactoring.

Aggregation constants. In family order (trusted resource, deterministic, cross-family, benchmark inputs, solver family), the support weights are $(1, .8, .7, .55, .3)$, the diagonal correlations are $(.25, .35, .40, .60, .85)$, and $\tau = 2$. Off-diagonal correlations are $.45$ for benchmark/solver, $.20$ for benchmark/deterministic and solver/deterministic, $.30$ for trusted-resource/benchmark, $.15$ for trusted-resource/solver and $.10$ for all remaining pairs. Ancestor Jaccard overlap can only raise these values. Risk aggregation uses only active private T1 evidence. Rows sharing a stable producer-invocation batch identifier are first reduced to one batch with summed atomic pass/total counts; diagnostic rows already represented by a summary do not count again. If B is the resulting batch set, then $\hat{n}_{\text{tot}} = d \sum_{b \in B} w_{f_b} n_b$ and $\hat{n}_{\text{pass}} = d \sum_{b \in B} w_{f_b} k_b$, where $d = \text{IndepScore}(B)/|B|$. Public T0 evidence remains available to hard gates, coverage and vetoes but contributes neither support nor Wilson trials.

B Per-Domain Producers

The TRUSTED_RESOURCE family groups visible benchmark specifications and external KB evidence. The code enum is `HUMAN_ORACLE`, serialised as `human_oracle`; these legacy names do not denote human annotation or access to sealed labels. "cond." marks a veto issued only where Proposition 2 applies, that is, where the benchmark's answers are generated from the walked KB (`KB_DEDUCTIVELY_COMPLETE`: true for MetaQA, false for KQA Pro). The condition is a declared provenance fact, never an empirical threshold.

Benchmark inputs and determinism. Benchmark-input probes execute harvested inputs without reading expected outputs. The certificate premise additionally requires equivalence to the graded invocation; the MBPP+ false positives show that this cannot be inferred from input provenance alone. The determinism producer repeats a call on copied arguments and flags inconsistent outputs.

Table 5: Evidence producers by domain. "Veto" marks producers that can emit operational veto flags; it does not assert certificate soundness.

Producer	Family	Vis.	Obligation	Veto
Code (MBPP+ / HumanEval+)				
`gate.base_tests`	TRUSTED_RESOURCE	T0	spec_conformance	yes
`probe.bench_random`	BENCH_INPUTS	T1	robustness	**yes**
`probe.bench_holdout`	BENCH_INPUTS	T1	robustness	**yes**
`probe.bench_*.consensus`	BENCH_INPUTS	T1	behaviour_consensus	no
`probe.llm_corner`	SOLVER_FAMILY	T1	edge_case	no
`property.purity`	DETERMINISTIC	T1	purity	yes
Text-to-SQL (Spider)				
`gate.sql_schema`	DETERMINISTIC	T0	spec_conformance	yes
`probe.db_instances`	BENCH_INPUTS	T1	robustness	**yes**
`probe.db_holdout`	BENCH_INPUTS	T1	edge_case	yes
`probe.result_consensus`	BENCH_INPUTS	T1	behaviour_consensus	no
`property.nonempty`	DETERMINISTIC	T1	purity	no
Knowledge graph question answering (KGQA): MetaQA-1hop / -3hop / KQA Pro				
`gate.answer_form`	DETERMINISTIC	T0	spec_conformance	no
`kb.reachability`	TRUSTED_RESOURCE	T1	robustness	cond.
`kb.typed_path`	TRUSTED_RESOURCE	T1	edge_case	no
`kb.path_proof`	TRUSTED_RESOURCE	T1	edge_case	**cond.**
`probe.answer_consensus`	SOLVER_FAMILY	T1	behaviour_consensus	no
`property.specificity`	DETERMINISTIC	T1	purity	no
Multiple-choice geometry (MathQA-Geo)				
`gate.exec_form`	DETERMINISTIC	T0	spec_conformance	no
`property.option_form`	BENCH_INPUTS	T0	spec_conformance	no
`exec.self_contradiction`	DETERMINISTIC	T1	robustness	no
`lattice.derivable`	TRUSTED_RESOURCE	T1	edge_case	no
`probe.answer_consensus`	SOLVER_FAMILY	T1	behaviour_consensus	no
Any domain (§4.3)				
`judge.cross_family`	CROSS_FAMILY	T1	behaviour_consensus	no

Such violations trigger operational vetoes, although the execution-certificate proposition does not establish that determinism is a benchmark requirement. These flags are distinct from failures on graded assertions.

A type-contract producer inferred from the visible assertions was implemented, measured (only 20 violations on MBPP+, slightly negative), and discarded. We report it because "which candidate producers do not work" is part of the result rather than an omission.

C THE KB PATH PROBE

Measured behaviour. The search budget is 20,000 expanded states. Simple paths are retained for semantic ranking, but the negative certificate uses a separate exact-hop reachability pass that allows repeated nodes and deduplicates states only by depth and entity. An offline audit over all 2,000 reported MetaQA cases and 14,731 gold answers finds four 3-hop answers without a simple-path witness but zero without an exact-hop walk; neither certificate search truncates (`code/results/geva/metaqa_certificate_witness_audit.json`). The ranking traversal takes 47 ms on average and expands 3,244 states on 3-hop, versus 0.1 ms on 1-hop. Its mean simple-path entity sets contain 7,569 and 6.6 entities, respectively. Both passes score only existing claims and never add retrieved answers to the pool. If either required search truncates, the implementation withholds the corresponding negative evidence.

The router's action constants. Let $u = \max(0, \text{risk}_{\text{score}} - \text{risk}_{\text{point}})$ and $n = \max(1, \hat{n}_{\text{tot}})$. For probe, property and KB actions, the estimated benefit is $u(1 - \sqrt{n/(n+m)})$ plus the credits below. Cross-family acquisition instead uses uH, where H is the incumbent's family concentration. Challenge uses the incumbent risk score when an operational veto is present.

Algorithm 1 ACQUIRE-KB-PATH: one traversal per instance, serving every claim

Input: topic entity t, question q, hop count k, claims $\{c_i\}$, KB K, state budget B
$P \leftarrow \emptyset$; $\Sigma \leftarrow \emptyset$; `trunc`←**false**
push $(t, \langle\rangle, \{t\})$ onto a first-in, first-out (FIFO) queue
while queue non-empty **do**
  if expanded states $\geq B$ **then**
    `trunc`←**true**; **break**
  end if
  pop (u, π, V); **if** $|\pi| \geq k$ **then continue**
  for all (r, w, d) in outgoing and incoming edges of u with $w \notin V$ **do**
    $\pi' \leftarrow \pi \cdot (r, d)$
    if $|\pi'| = k$ **then**
      $P[w] \leftarrow P[w] \cup \{\pi'\}$; $\Sigma \leftarrow \Sigma \cup \{\pi'\}$
    else
      push $(w, \pi', V \cup \{w\})$
    end if
  end for
end while
$\mathrm{sem}(\pi) \leftarrow$ fraction of relations in π whose alias appears in q, plus .25 for an initial repeated relation traversed in opposite directions in a "share" question, and .25 for matching the mentioned-relation multiset (doubling that shared relation)
for all claim c_i with answer a_i **do**
  $S_i \leftarrow$ lookup(P, a_i) {exact key, else lenient substring match, matching the benchmark metric's own leniency}
  $b_i \leftarrow \max_{\pi \in S_i} \mathrm{sem}(\pi)$, or 0 if $S_i = \emptyset$
end for
$b^\star \leftarrow \max_i b_i$
$(R, \mathtt{trunc}_w) \leftarrow$ exact-k-hop walk reachability from t in K with budget B; allow repeated nodes, deduplicate by (depth, entity)
for all claim c_i **do**
  if no normalised or substring match of a_i in R, and neither search truncated **then**
    emit `kb.path_proof.counterexample`; mark as a sound certificate only if `KB_DEDUCTIVELY_COMPLETE`
    `invalidate` `kb.typed_path` and positive `kb.reachability` of c_i
  else if $S_i \neq \emptyset$ **then**
    $s_i \leftarrow b_i / b^\star$ if $b^\star > 0$, otherwise 0; halve s_i for substring-only identity matches
    emit `kb.path_proof` (edge_case, strength s_i) and `kb.path_proof.reachable` (robustness, strength 1), both with n_{checks}=1
    `invalidate` `kb.typed_path` and positive `kb.reachability` of c_i
  end if
end for
assert the claim count is unchanged {the walk may rank claims, never create them}

Action	Cost	m	Additional credit
Benchmark probe	1	8	None
Property	1	4	.10 missing purity; .15 missing robustness
LLM probe	4	5	.20 missing edge case
KB path	2	3	.15 missing edge case; $.25\min(1, M)$
Cross-family judge	6	–	uH replaces the contraction estimate
Challenge	12	–	Incumbent risk if vetoed

Here M sums positive strengths with an unvalidated-inference ancestor. For positive evidence, set $v_f = \sum_{e:f_e=f} w_f n_e$ and $H = \max(0, (\sum_f (v_f / \sum_j v_j)^2 - 1/5)/(1 - 1/5))$; without positive evidence, $H = 1$. If K is the largest recorded cluster count, L the number of claims and D their support spread, then $\mathrm{GraphCut} = \mathrm{clip}_{[0,1]}[\frac{1}{2}\min(1, D/\sum_f w_f) + \frac{1}{2}(K-1)/(L-1)]$ for $L > 1$ (the cluster term is zero if unavailable); it is 1 for a single claim. These formulas specify the action heuristic and do not assert optimal expected error reduction.

Ancestry. `kb.path_proof` carries {`producer`, `tool_version`, `family`, `artifact:kb`, `tool:path_enumeration/exact_hop_simple_paths`, `tool:path_semantic_alignment/relation_alias_table`}. It therefore shares three ancestors with `kb.reachability` (Jaccard ≈ 0.43, above the 0.25 intra-family prior, so the pair is discounted towards a single observation) yet does *not* share `tool:relation_inference/keyword_first_match`. The ancestry distinguishes executed path enumeration from the earlier relation guess, allowing the corresponding observations to be superseded explicitly.

C.1 Ablation of Acquire-KB-Path on MetaQA-3hop

Table 6 is retained as a historical implementation diagnostic from before the T1-only, batch-invariant risk correction; it is not mixed with the current headline metrics. It shows why serialising the k edges of one chain as k statistical trials was unsafe and motivated the packing-invariance test now enforced in code. Rows (a)–(e) score chains by alias coverage alone. Charging the k edges of one chain as k independent checks (b, c) shrinks the Wilson interval enough to accept 91% of instances at 44% accepted accuracy, with 0.648 error at 25% coverage against 0.591 overall. Superseding only the relation-guess evidence (d) is insufficient, because the reachability observation draws its strength directly from the verifier score whose top level *is* the superseded guess; that single omission fills the lowest-risk quartile with 35%-accurate claims while the second quartile runs at 54%. With $n_{\text{checks}}{=}1$ and full supersession (e), accuracy and AURC are best among variants (a)–(e), and the risk ordering is monotone ($.392 \to .492 \to .575 \to .588$).

Table 6: Historical pre-correction Acquire-KB-Path implementation ablation on MetaQA-3hop. It diagnoses edge over-counting and supersession but does not use the current T1-only, batch-invariant risk semantics. Bold marks the best value in each column; variant (a)'s lower false-trust accompanies only 16.5% acceptance.

	Variant	Acc.	Accept	False-trust	AURC
(a)	no KB traversal (re-weighting only)	33.30	16.5%	**6.60**	.5303
(b)	$n_{\text{checks}}{=}k$, no supersession	40.00	59.7%	28.10	.4875
(c)	$n_{\text{checks}}{=}k$, with supersession	41.00	91.0%	50.80	.6016
(d)	$n_{\text{checks}}{=}1$, partial supersession	40.90	47.7%	26.80	.5914
(e)	$n_{\text{checks}}{=}1$, full supersession	41.20	26.9%	10.30	.4849
(f)	**(e) + relation multiplicity + walk-safe absence**	**42.70**	26.7%	10.10	**.4609**

D Two Producers That Did Not Work

Spider retains the second-largest gap (69.83 against a 78.14 ceiling), and the natural hypothesis is that its harness checks executability and denotation stability but never whether the query answers the question that was asked. We fixed two check sets from semantics, implemented them, and measured once, deliberately *not* selecting sub-checks against the sealed labels. Neither survived. Question-intent / SQL-structure agreement (aggregation cues versus `COUNT`, `AVG`, `GROUP BY`, `ORDER BY`. . .`LIMIT`) is monotone at claim level ($0.673/0.617/0.549$ for $0/1/2+$ mismatches against a 0.652 base rate) but *lowers* pool reranking ($0.682 \to 0.678$), because it correlates with the question type rather than with candidate quality; candidates within one instance usually share the same mismatch pattern. Schema-linking coverage, i.e. whether the SQL references the schema items the question names, is not even monotone ($0.642/0.668/0.613/0.714$) and costs more ($0.682 \to 0.664$). A per-check inspection also finds an *anti*-predictive member: `cue_group_without_group_by` fires on claims that are 0.790 accurate, above base rate, because "each" in a question often needs no `GROUP BY`. Pruning such members by label would be exactly the overfitting the protocol forbids. We therefore conclude only that these two pre-declared surface question–SQL checks do not close Spider's residual gap; other semantic checks remain open.

E THE FLAT EVIDENCE LEDGER

Figure 3 in the main paper presents all six regimes for the baseline family of §4.4. Each selector receives the flat harness view (every producer on every instance, invalidated and consumed evidence included with flags), every quantity the graph computes as a feature (per-family aggregates, per-obligation coverage, certificate counts, cost, ancestor counts, mean/max pairwise ancestor Jaccard, a Herfindahl index of support concentration, and within-case relative versions of all of these), and sealed outcomes to fit on. `scalar_platt` is the independence-blind weighted pass rate, Platt-scaled; `ledger_lr` is family-aware logistic regression (LR) on the flattened vector; `ledger_gbdt` uses histogram-based gradient-boosted decision trees (GBDT) on the same vector, free to discover any interaction the graph encodes; `ledger_deepsets` is a permutation-invariant sum-pooled multilayer perceptron (MLP) over the raw per-evidence rows plus globals; `ledger_settrans` is a set transformer (Set-Transf. in the table), with one mean-pooled multi-head self-attention block over the evidence set. Three protocols are reported. *In-domain* is 5-fold cross-fitting by instance inside one regime (the ledger's upper bound). *Transfer* is leave-one-regime-out on the domain-invariant feature block, which is the setting the graph policy actually operates in: no labels from the target regime and no per-domain tuning. *Learning curve* (Figure 3) interpolates between them: 40% of the regime is held out once per seed and kept fixed across budgets, and label budgets of 8–256 cases are drawn from the remaining 60%. Up to five seeds are averaged; invalid single-class draws are omitted. Thus HumanEval+ at $n{=}8$ has four valid draws and a matched GEVA reference of 87.50%, whereas its later budgets have five draws and an 86.97% reference. All other plotted points use five draws. Budgets exceeding a regime's training pool are omitted. For Table 7, selector thresholds target GEVA's acceptance rate; finite thresholds and ties can produce small realized-coverage differences, which are retained rather than post-hoc adjusted. Nothing in this module is used by the deployed policy: it is executed only by the meta-evaluator, after every decision is frozen. The three protocols are reproduced by

```
python -m code.eval.codeplus.geva.flat_ledger -config results/geva/ledger_config.json
            -protocol {both|learning_curve} -output results/geva/OUT.json
```

Exact values for Figure 3, including the plotted standard deviations, are supplied in `supplementary_data/learning_curves.csv`, exported from `code/results/geva/ledger_learning_curve_riskfix.json`. KQA Pro uses the 9B judge-enabled configuration described in Appendix J. The one-feature scalar baseline saturates early (75.23 on MBPP+, 67.20 on Spider, 39.05 on MetaQA-3hop), whereas the multi-feature selectors continue to respond to additional supervision. Two regimes deserve a caveat. HumanEval+ has 164 cases, so its held-out split is 66 instances and one instance is 1.5 points; its crossing at eight labels should be read as "the regime is too easy and too small to separate anything". On MetaQA-1hop every entry, GEVA included, lies within 1.05 points of the pool ceiling of 34.50.

The in-domain ledger is competitive on accuracy. Risk ordering is also regime-dependent: on Spider, logistic regression and gradient-boosted trees have AURC $.125$ and $.140$, below GEVA's $.218$. Under transfer, however, logistic regression's MetaQA-1hop AURC rises to $.833$ and false-trust to 26.70%, versus $.324$ and 1.60% for GEVA. These results support a distinction between in-domain fitting and transfer reliability, not a universal advantage of graph storage over flat features.

F PER-REGIME HEADROOM AND RISK DIAGNOSTICS

F.1 RISK ORDERING DIAGNOSTICS

Figure 4 shows every evaluator-ranked selective-risk point using the exact tie rule that produces AURC. The MetaQA-3hop panel uses the disjoint 800-case audited evaluation; the other panels use their final full runs.

The learning-curve comparison in Figure 3 and the archived multi-selector AURCs in Table 7 provide complementary comparisons.

Table 7: Flat-ledger selectors against GEVA, accuracy (%) / AURC. Bold marks the highest accuracy in each row. Upper panel: in-domain sealed labels, 5-fold cross-fitted. Lower panel: leave-one-regime-out transfer. GEVA is the same run in both panels because it fits nothing. This is the archived `flat_ledger_v2.json` snapshot: its MBPP+ row predates corrected grading, and its MetaQA rows predate the exact-hop walk certificate correction. It is not mixed with the corrected headline results in Table 1.

	GEVA	scalar (Platt)	ledger LR	ledger GBDT	DeepSets	Set-Transf.
IN-DOMAIN: 5-FOLD CROSS-FITTED ON THAT REGIME'S SEALED LABELS						
MBPP+	80.42 / .073	78.31 / .153	80.16 / .105	**80.69** / .077	79.63 / .105	79.37 / .098
HumanEval+	87.80 / .029	87.20 / .035	89.02 / .022	89.02 / .024	**89.63** / .027	88.41 / .036
Spider	**70.21** / .218	66.92 / .208	69.54 / .125	67.70 / .140	68.28 / .155	68.57 / .131
MetaQA-1hop	33.70 / .324	33.60 / .322	34.00 / .320	**34.20** / .315	34.00 / .320	34.10 / .319
MetaQA-3hop	42.70 / .460	39.20 / .473	42.80 / .431	**43.00** / .436	42.20 / .469	41.80 / .477
KQA Pro	61.90 / .210	61.60 / .221	**63.00** / .199	61.70 / .229	61.20 / .263	61.50 / .229
TRANSFER: TRAINED LEAVE-ONE-REGIME-OUT, NO LABEL FROM THE TARGET REGIME						
MBPP+	80.42 / .073	78.31 / .125	79.37 / .108	**80.69** / .069	80.16 / .086	79.63 / .090
HumanEval+	87.80 / .029	87.20 / .030	**90.24** / .047	88.41 / .018	87.80 / .027	89.02 / .027
Spider	**70.21** / .218	66.92 / .203	63.93 / .428	68.86 / .244	68.57 / .298	68.18 / .212
MetaQA-1hop	33.70 / .324	33.60 / .318	23.60 / .833	**33.80** / .362	33.30 / .561	33.60 / .452
MetaQA-3hop	**42.70** / .460	39.20 / .476	42.10 / .523	37.80 / .622	37.80 / .573	35.10 / .543
KQA Pro	**61.90** / .210	61.60 / .225	54.70 / .354	57.40 / .352	59.10 / .268	59.90 / .253

Table 8: Selective prediction at $\rho = .35$ with T1-only risk accounting (%). FT is joint; accepted risk is conditional; "–" denotes undefined risk at zero acceptance.

Regime	Acc.	Accept	FT	Accepted risk	AURC
MBPP+	79.37	88.10	12.17	13.81	.0954
HumanEval+	87.80	90.24	4.88	5.41	.0331
Spider	69.83	89.85	24.47	27.23	.2086
MetaQA-1hop	33.70	10.70	0.50	4.67	.3177
MetaQA-3hop	42.70	19.20	7.50	39.06	.4555
KQA Pro	61.90	0.00	0.00	–	.1931
MathQA-Geo	86.92	0.00	0.00	–	.0644

F.2 SELECTION MARGIN UNDER POOL SCALING

MathQA-Geo: certificate eligibility and routing behavior. This expands §4.2. The candidate signal—a program that executes and lands on a different legal option than the one it asserts—has post-hoc precision 0.958 (n=667, base rate 0.303), but is not a sound certificate: a buggy derivation can coexist with a correct letter. It is retained as strong negative evidence without an operational-veto flag. Three mechanisms are therefore separated. (i) *Repair.* The current evaluation records $392/451$ correct outcomes (86.92%), against an initial-pool oracle of $421/451$. Repair is unavailable because this configuration assigns no operational veto, not because repair logically requires a sound certificate. (ii) *Audit.* With Phi-4, accuracy is 85.59 versus 86.92 without it, and acquisition units are 8.71 versus 4.97. With no eligible certificate supervision, the audit follows its keep-prior branch. (iii) *Risk gate.* All 451 outputs are unaccepted at $\rho = .35$. Their conditional accepted risk is undefined; reported accuracy evaluates the emitted ranking, not certified acceptance.

G ROUTER DEFECTS EXPOSED BY CROSS-DOMAIN PORTING

An audit of the evaluation harness exposed an entry-point extraction defect that assigned zero passes to $11/378$ MBPP+ tasks for every method. All comparisons use corrected labels. The leak audit found zero sealed-field occurrences.

Porting the router across domains exposed six defects that a single-domain evaluation could miss. Several surfaced only on Spider or the KG regimes, while cost ordering and evidence-consumption issues also affected code-domain behavior. We list them because "which mistakes this design makes

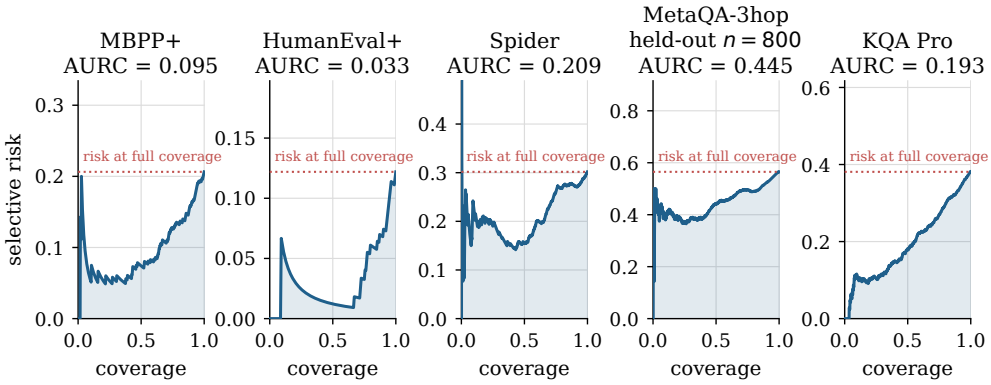


Figure 4: Complete selective-risk curves under the corrected T1-only accounting after ordering by Eq. 3; lower is better. The MetaQA-3hop panel is the audited held-out split ($n = 800$) corresponding to Table 2, not the $n = 1{,}000$ full-suite row in Table 8; other panels use full runs. Dotted lines mark full-coverage error, and all AURCs integrate every ranked output.

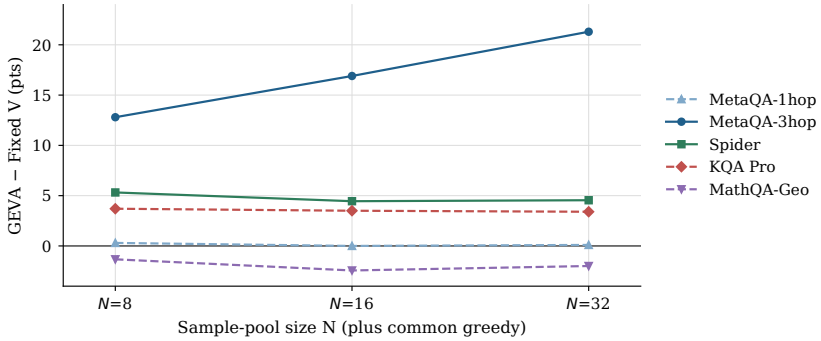


Figure 5: GEVA minus Fixed V accuracy (percentage points) as the pool grows, using the same nested-pool configurations and results as Table 3. Positive values favor GEVA. The scaling pools differ from the headline pools in Table 1; all five scaling regimes are retained.

easy" is part of its evaluation. Two scope statements belong here as well. First, repairs are revalidated on a probe slice held out from the start, so a repair cannot be accepted on the very evidence that was disclosed to produce it. Second, the experiments evaluate selection from saved initial pools and counterexample-driven repair. They do not establish a bound on answer correctness or state-of-the-art generation on any benchmark.

1. **The contradiction veto counted unflagged negatives.** Harmless on code, where almost every negative is a crash-triggered operational veto; generated probe failures are nevertheless ineligible as sound certificates unless they reproduce the benchmark's exact graded invocation. On Spider, "returns an empty result on all instances" and "references an unknown column" are common on *correct* queries, and on KQA Pro "unreachable" carries no deductive force. Restricting the veto to flagged negatives (design-protocol condition "no unresolved *high-confidence* contradiction") fixed it.

2. **The challenge trigger must require veto flags.** Trigger precision under the two candidate conditions: "every claim vetoed" gives $91.7/100/100/100\%$ on MBPP+/HumanEval+/Spider/MetaQA-1hop/-3hop, whereas "every claim has some negative" collapses to 36.8% on Spider.

3. **Cost ordering.** A fall-through let the 12-unit challenge fire on the bare hard-gate view, skipping 1-unit probes; it cost 0.5 accuracy points and dropped the challenge trigger's precision from 92% to 28%. Enforcing strict cost order moved an early MBPP+ snapshot

from 79.89 to 80.42; corrected grading yielded 79.63, and the final T1-only risk-accounting rerun is 79.37.

4. **The flat baseline needs a task-verifier variant.** On the KG regimes the generic "argmax of hard-gate pass rate" is a much weaker baseline than the paper's Fixed V; we added an explicit `verifier_v` baseline so the comparison is against the strongest in-domain verifier rather than a straw man.
5. **`consumes` over-invalidation.** Consuming the incumbent's negative evidence, or the whole probe batch rather than the disclosed probes, made challenged candidates look defect-free again and dropped repair adoption from $32/43$ to $9/43$ (§3.5).
6. **Option letters leak position into a judge's verdict.** Presenting multiple-choice candidates by their benchmark *letter* makes the judge pick 'A' in 8 of 14 adjudications regardless of where 'A' sits in the candidate list. Showing candidates as answers under a deterministically permuted neutral index is worth $+2.3$ points of judge-alone accuracy.

Action outcomes. Table 9 reports action counts and the accuracy and initial-pool oracle of the accepted subset.

Table 9: Per-action decomposition. "Accept acc." is routed accuracy on the accepted subset and "group oracle" the initial-pool oracle of that same subset. The challenge column records terminal challenge requests, not the number of successfully generated repairs. "Unaccepted" denotes an emitted answer without acceptance (stored action `abstain`).

Regime	ACCEPT	CHALLENGE	UNACCEPTED	Accept acc.	Group oracle
MBPP+	333	36	9	86.2	90.4
HumanEval+	148	11	5	94.6	99.3
Spider	929	0	105	72.8	80.9
MetaQA-1hop	107	595	298	95.3	95.3
MetaQA-3hop	192	11	797	60.9	60.9
KQA Pro	0	0	1000	–	–
MathQA-Geo	0	0	451	–	–

H THE CERTIFICATE AUDIT

Algorithm 2 Audit one producer against the graph's certificates (label-free)

Input: T1 calibration graphs $\mathcal{G}$ (with the producer's evidence attached), producer p, level α
$n_p \leftarrow 0$; $\text{err} \leftarrow 0$; $\mathbf{q} \leftarrow [\,]$
for all $G \in \mathcal{G}$ **do**
  $W \leftarrow \{\text{claims of } G \text{ with a premise-backed sound certificate from a producer} \neq p\}$
  if $W = \emptyset$ **or** $W =$ all claims of G **then**
    continue {no clean alternative existed}
  end if
  $c^\star \leftarrow$ claim carrying p's maximal positive, non-invalidated strength
  if $c^\star$ undefined **or** its strength $= 0$ **then**
    continue {p expressed no preference}
  end if
  $n_p \leftarrow n_p + 1$; $\text{err} \leftarrow \text{err} + [\,c^\star \in W\,]$; $\mathbf{q}.\text{append}(|W|/|\text{claims}(G)|)$
end for
if $n_p = 0$ **then**
  return *keep prior*
end if
$p_{\text{lower}} \leftarrow \text{PoissonBinomialCDF}(\text{err}; \mathbf{q})$
return *retain* if $p_{\text{lower}} \leq \alpha$ else *withdraw*

The implementation stores operational vetoes and sound certificates separately. An evidence item enters W only when its certificate kind and source are present and its producer-declared semantic

Table 10: Exact conditional random-picker audit on the same 200 calibration cases. n_p is the eligible count, k_p the errors. Retain requires lower-tail $p \leq .10$; upper-tail p diagnoses unusually many errors but is not the retention rule.

Judge	n_p	k_p	Mean q_i	Lower p	Upper p	Decision
Qwen3.6-27B	30	8	.2593	.6403	.5383	withdraw
Phi-4-14B	29	9	.2605	.8209	.3201	withdraw

premises hold; no sealed outcome can promote a veto into audit supervision. Legacy graphs are migrated by a fixed producer-contract map, reported in `certificate_source_premise_map.json`.

Retention feeds exactly one number into the router: a factor of 1 for a retained producer and 0 for a withdrawn one, applied to the whole numerator of Eq. 4. Producers with $n_p = 0$ are not audited and retain their family prior.

Availability is a policy decision. The factor gates the whole action utility, rather than leaving an exploration credit for a withdrawn producer. Passing keeps the original utility; it is not an estimate of the size of any accuracy gain. An uncertified alternative is not necessarily correct, and failure to pass does not imply absence of useful information.

Calibration and evaluation instances. The audit uses the leading 200 of 1,000 sorted MetaQA-3hop identifiers, reading only T1 evidence. This non-random split can correlate with templates. The frozen decision is evaluated on the remaining 800 cases. The revised exact test retains every eligible q_i instead of replacing them by their mean $\bar{q}$. The reference variance is $\sum_i q_i(1-q_i) = n\bar{q}(1-\bar{q}) - \sum_i (q_i - \bar{q})^2$; therefore mean chance alone does not determine the null distribution. We compute its inclusive tails by finite convolution in $O(n^2)$ time and $O(n)$ memory.

Both decisions match the historical Wilson rule (upper values $.3803/.4281$), so no held-out policy output is changed by this statistical replacement. Neither upper-tail test establishes that the judge is worse than random. A distinct-answer rather than claim-ID-uniform sensitivity gives lower-tail $p = .4202/.6370$ and also withdraws both judges. This changes the reference sampling unit, not just precision. The KB producer has no eligible opportunities and keeps its prior. Complete eligible IDs, errors, q_i, chosen/wrong claim IDs and input/module hashes are recorded in `code/results/geva/audit_exact_final_20260918/`; the permanent offline entry point is `audit_review`. It reproduces the historical calibration aggregates before computing the new test and never calls a model on a missing cache entry.

Statistical scope. The exact distribution concerns independent uniform choices conditional on these fixed pools, certificates and eligibility. It is not an exact test under arbitrary cross-instance dependence or merely an average-error constraint. Repeated templates, shared systematic certificate errors and post-development selection remain limitations. No calibrated benchmark-correctness bound follows. With only 29–30 eligible observations, non-retention may reflect low power. Tests are nominal per producer, not a multiplicity-adjusted discovery claim; adjusted thresholds also withdraw both judges here. Wider utility requires additional producers, random/stratified splits and held-out downstream comparisons. Historical full-suite results remain separately labelled in Table 11.

Table 2 reports the disjoint held-out effect of the calibration decision. The audit gains 4.38 points over the unaudited judge (95% CI $[+2.50, +6.25]$, exact paired $p < 10^{-5}$; 47 improved, 12 harmed). It restores the no-judge policy, rather than exceeding it; acceptance and conditional accepted risk are unchanged. This is instance-disjoint evaluation, not an independent replication of method development or a random-split stability study.

Full-suite adjudication. Table 11 is retained only as a routing diagnostic from the original flag-based audit. It is not evidence for the current sound-certificate audit: generated code probes are now ineligible, and MBPP+ judge routing was not recomputed after corrected grading, so no audited MBPP+ final is reported. On the archived Spider snapshot, judge-alone accuracies are 65.18 and 65.67, and the routed values are 70.21 and 70.12; MetaQA-1hop routing is unchanged. Across all fourteen archived no-J./+J. comparisons, GEVA with the judge exceeds judge-alone accuracy,

Table 11: Archived full-suite adjudication on the pre-certificate-split Qwen3.5-9B snapshot, accuracy (%). These rows include audit-calibration cases and are distinct from Table 2. Qwen3.6-27B shares the policy's vendor but differs in generation; Phi-4-14B differs in vendor and pretraining corpus. Judge uses greedy fallback on unanimous pools or `NONE`; no J./+J. are GEVA without/with the unaudited judge. W/R are the archived flag-based audit statuses, not sound-certificate decisions; — means disabled or not reported. Bold marks the best routed accuracy per row, including ties.

Regime (n)	Judge	Greedy	Judge	no J.	+J.	Audit	Final
MBPP+ (378)	Qwen3.6	67.72	68.78	**80.42**	**80.42**	—	—
	Phi-4	67.72	67.20	**80.42**	**80.42**	—	—
HumanEval+ (164)	Qwen3.6	80.49	78.66	**87.80**	**87.80**	W	**87.80**
	Phi-4	80.49	79.27	**87.80**	87.20	R	87.20
Spider (1,034)	Qwen3.6	67.60	65.18	**70.21**	**70.21**	R	**70.21**
	Phi-4	67.60	65.67	**70.21**	70.12	R	70.12
MetaQA-1hop (1,000)	Qwen3.6	24.70	27.40	**33.70**	**33.70**	R	**33.70**
	Phi-4	24.70	26.50	**33.70**	**33.70**	R	**33.70**
MetaQA-3hop (1,000)	Qwen3.6	33.10	34.10	**42.70**	37.20	W	**42.70**
	Phi-4	33.10	34.80	**42.70**	36.30	W	**42.70**
KQA Pro (1,000)	Qwen3.6	55.10	60.80	59.00	**61.90**	P	**61.90**
	Phi-4	55.10	57.70	59.00	**60.00**	P	**60.00**
MathQA-Geo (451)	Qwen3.6	71.40	63.19	**88.25**	86.92	P	86.92
	Phi-4	71.40	67.63	**88.25**	87.58	P	87.58

but adding a judge improves over no-judge GEVA only on KQA Pro. Current certificate-qualified conclusions rely on the disjoint MetaQA-3hop audit in Table 2 and the no-certificate boundary discussion in Section H, not on the archived W/R statuses here.

Cost of the judge. On KQA Pro, Qwen adjudication costs 6 verification units on the 66.1% of instances where the router requests it. The graph combines the verdict with KB reachability and consensus. On MetaQA-3hop, adding this judge raises false-trust from 10.0 to 22.8. Withdrawal reduces judge purchases to 0.0%. One vLLM server (Qwen3.6-27B, bf16, one H20, $\approx$51 GiB of weights), $\approx$2 tokens of output per verdict with thinking disabled, 24 concurrent requests: 1000 KQA Pro adjudications in 86 s, 1000 MetaQA-3hop in 105 s. Once verdicts and the other per-instance artifacts have been generated, `decide -judge-verdicts` supports CPU-only replay. Those complete artifacts are not bundled here, so the supplied package alone does not reproduce every result end to end.

I Simulation of the Independence-Adjusted Risk Score

The quantity the risk gate thresholds is one minus a Wilson lower bound evaluated on independence-discounted effective counts $n_{\text{eff}} = m/(1 + (m-1)\bar{\rho})$. One part of this is exact.

Proposition 3 (Variance matching is exact)**.** *Let $X_1, \ldots, X_m$ be exchangeable Bernoulli(p) checks with $\mathrm{Corr}(X_i, X_j) = \bar{\rho}$ for $i \neq j$. Then $\mathrm{Var}(\bar{X}) = \frac{p(1-p)}{m}\big(1 + (m-1)\bar{\rho}\big) = \frac{p(1-p)}{n_{\text{eff}}}$ with n_{eff} as in Eq. 1.*

Proof. $\mathrm{Var}(\sum_i X_i) = m\,p(1-p) + m(m-1)\bar{\rho}\,p(1-p)$; divide by m^2. □

Hence n_{eff} matches the variance of an independent sample mean. Variance equivalence does not imply equal information or transfer the result to heterogeneous correctness proxies. Two further issues affect interval coverage: Wilson's coverage is nominal only to the normal approximation (and fractional counts lie outside its derivation), and $\bar{\rho}$ is a family prior rather than an estimate. We therefore do not claim $\Pr[R \leq \mathrm{RiskScore}] \geq 1 - \alpha$ as a theorem.

Table 12 simulates correlated Bernoulli checks (20,000 trials per cell, α=.10, seed 2026) on a grid of $p \in \{.60, .75, .90, .95\}$ and $m \in \{4, 8, 16\}$ at controlled $\bar{\rho} \in \{0, .20, .40, .60, .85\}$, comparing the independence-blind interval against the adjusted one. The failure mode of ignoring correlation is not mild: at $\bar{\rho}$=.85 the naive interval covers 0.677 of the time on average against a nominal 0.90,

Table 12: Simulated coverage of the nominal-0.90 lower bound on correlated Bernoulli checks, by true pairwise correlation. "naive" assumes independence ($n_{\text{eff}}{=}m$); "adjusted" uses the design-effect discount. "min" is the worst cell. Reproduced by `python -m code.eval.codeplus.geva.bound_calibration`.

$\bar{\rho}_{\text{true}}$	naive mean	naive min	adjusted mean	adjusted min
0.00	0.945	0.816	0.945	0.816
0.20	0.862	0.638	0.969	0.771
0.40	0.793	0.505	1.000	1.000
0.60	0.735	0.377	1.000	1.000
0.85	0.677	0.210	1.000	1.000
All cells, by quality of the $\bar{\rho}$ prior				
correct $\bar{\rho}$	0.803	0.210	0.983	0.771
under-estimated by 0.15	0.803	0.210	0.964	0.688
over-estimated by 0.15	0.803	0.210	0.997	0.866

and only 0.210 in its worst cell. Across the entire grid, mean coverage is .803 without adjustment and .983 with the correct correlation; underestimating correlation by .15 yields .964. These values describe the simulated Bernoulli model, not calibration of the benchmark correctness of accepted outputs.

J Reproduction

Evaluation configurations. The reported resource configurations use the result files indexed in `supplementary_data/result_sources.csv`. Code evaluations include repair records; the initial-pool baselines do not generate those repairs. MetaQA uses live KB-path evidence, and KQA Pro uses a Qwen3.6-27B judge (Qwen Team, 2026b); MathQA-Geo uses no judge. The second-policy KQA Pro configuration uses Phi-4-14B (Abdin et al., 2024). The saved configuration assigns the same-vendor Qwen judge the cross-family prior; this is a modelling choice rather than established statistical independence. The second-policy comparison evaluates resource configurations rather than an isolated repair-generator effect. Judge-resource effects are reported in Table 11. The cost comparison uses the `eval_budget` result files with $\rho = .35$, four steps for code/Spider, six for MetaQA-1hop, seven for MetaQA-3hop and five for KQA Pro. The default cost budget is unlimited; step limits and positive action utility terminate acquisition. Initial generation, cached graph preparation and producer calibration are separate from the requested-action charges.

MathQA-Geo subset. The 451 cases come from the geometry category in MathQA's development and test splits, with normalised-text duplicates removed. The builder requires five option letters, a valid gold letter and at least three numeric option denotations, then orders by a fixed hash of the instance identifier. Gold formulas and rationales are reserved for offline diagnostics; producers do not read them.

Pipeline. Four stages, all deterministic given the saved pools: (1) `*_records` builds the label table and leak-free per-claim features; (2) `*_build_graph` emits one provenance graph per instance; (3) `decide` runs the budgeted router; (4) `geva_eval` joins decisions against sealed outcomes and is the only module permitted to read them. `challenge` sits between (3) and (4) where repairs are used.

Commands (KG regimes, including the live KB traversal). Run from the supplied `code/` directory. The shell variable below names the actual Python package; all data paths are relative to that directory.

```
GEVA=code.eval.codeplus.geva
KB=data/benchmarks/metaqa_official/raw/kb.txt
for D in metaqa1 metaqa3 kqapro; do
  python -m $GEVA.kgqa_records     --domain $D  --output results/geva/${D}_records.jsonl
  python -m $GEVA.kgqa_build_graph --records results/geva/${D}_records.jsonl \
                                   --output  results/geva/${D}_graphs_v2.jsonl
done
```

```
for D in metaqa1 metaqa3; do
  STEPS=6; [ "$D" = metaqa3 ] && STEPS=7
  python -m $GEVA.decide --graphs results/geva/${D}_graphs_v2.jsonl \
      --policy graph_agent --max-steps $STEPS --live-kb $KB \
      --output results/geva/${D}_dec_kbpath.jsonl
  python -m $GEVA.geva_eval --decisions results/geva/${D}_dec_kbpath.jsonl \
      --records results/geva/${D}_records.jsonl \
      --output  results/geva/${D}_eval_kbpath.json
done
python -m $GEVA.kgqa_kb_probe --records results/geva/metaqa3_records.jsonl \
      --kb $KB --audit --output results/geva/metaqa3_kbprobe_walk.jsonl \
      --summary-output results/geva/metaqa3_kbprobe_walk_summary.json
# sealed-label premise audit; reporting only, never read by the policy
python -m $GEVA.audit_metaqa_certificate_witness --kb $KB \
      --records data/benchmarks/metaqa_official/test_1hop_1000.jsonl \
                data/benchmarks/metaqa_official/test_3hop_1000.jsonl \
      --output results/geva/metaqa_certificate_witness_audit.json

# cross-family judge: record verdicts once, then replay (no GPU needed later)
python -m $GEVA.cross_family --records results/geva/kqapro_records.jsonl \
      --bench data/benchmarks/kqapro/val_1000.jsonl --style neutral --audit \
      --output results/geva/kqapro_judge.jsonl
# audit the judge against the graph's own certificates, then route
python -m $GEVA.reliability --graphs results/geva/metaqa3_graphs_v2.jsonl \
      --slice 0.2 --live-kb $KB --judge-verdicts results/geva/metaqa3_judge.jsonl \
      --producers judge.cross_family,kb.path_proof \
      --output results/geva/metaqa3_reliability.json
python -m $GEVA.decide --graphs results/geva/metaqa3_graphs_v2.jsonl \
      --policy graph_agent --max-steps 7 --live-kb $KB \
      --judge-verdicts results/geva/metaqa3_judge.jsonl \
      --reliability results/geva/metaqa3_reliability.json \
      --output results/geva/metaqa3_dec_audited.jsonl
```

The recorded step limits are 6 for MetaQA-1hop, 7 for MetaQA-3hop and 4 for code and Spider. ACQUIRE-KB-PATH is unavailable on graphs that cannot support the traversal. `-audit` joins the certificate against sealed labels for reporting only and writes nothing back into the pipeline.

Commands (MathQA-Geo, the regime without a certificate). The pool, the sandbox transcripts and the derivability lattice come from the shared upstream stages; only the three GEVA-side stages below are specific to this paper. The lattice is a property of the question, so it is passed to the graph builder rather than stored per candidate.

```
GEVA=code.eval.codeplus.geva
python -m $GEVA.mathqa_geo_records \
      --bench  data/benchmarks/mathqa_geo/geometry_devtest.jsonl \
      --scored results/mathqa_geo/tts_v1_geometry_scored.jsonl \
      --output results/geva/mathqa_geo_records.jsonl
python -m $GEVA.mathqa_geo_build_graph \
      --records results/geva/mathqa_geo_records.jsonl \
      --lattice results/mathqa_geo/lattice_d3_all.json \
      --output  results/geva/mathqa_geo_graphs.jsonl
python -m $GEVA.decide --graphs results/geva/mathqa_geo_graphs.jsonl \
      --policy graph_agent --max-steps 4 \
      --output results/geva/mathqa_geo_dec_nojudge.jsonl
python -m $GEVA.geva_eval --decisions results/geva/mathqa_geo_dec_nojudge.jsonl \
      --records results/geva/mathqa_geo_records.jsonl \
      --output  results/geva/mathqa_geo_eval_nojudge.json

# the adjudicator is harmful here and, with no definite counterexample in the
# graph, `reliability` reports zero opportunities instead of withdrawing it
python -m $GEVA.reliability --graphs results/geva/mathqa_geo_graphs.jsonl \
      --judge-verdicts results/geva/mathqa_geo_judge_phi4.jsonl \
      --judge-model judge_phi4_14b \
      --output results/geva/mathqa_geo_reliability_phi4.json
```

Commands (new controls). The strict pool control, disjoint audit split and full risk curve are reproduced by:

```
python -m $GEVA.scaling --config results/geva/scaling_nested32_config.json \
  --grid 8,16,32 --seeds 0 --subset-mode prefix \
  --persist-dir results/geva/nested_8_16_32 \
  --output results/geva/nested_scale_8_16_32.json
python -m $GEVA.scaling --config results/geva/scaling_config.json \
  --only metaqa1,metaqa3,kqapro --grid 1,2,4,8 --seeds 0,1,2 \
  --subset-mode nested_permutation --persist-dir results/geva/nested_figure9 \
  --output results/geva/scaling_kg_nested.json
python -m $GEVA.split_cases --graphs results/geva/metaqa3_graphs_v2.jsonl \
  --records results/geva/metaqa3_records.jsonl --calibration-fraction .2 \
  --manifest results/geva/metaqa3_audit_split.json \
  --heldout-graphs results/geva/metaqa3_graphs_heldout800.jsonl \
  --heldout-records results/geva/metaqa3_records_heldout800.jsonl
python -m $GEVA.geva_eval --decisions results/geva/DOMAIN_decisions.jsonl \
  --records results/geva/DOMAIN_records.jsonl --output results/geva/DOMAIN_eval.json \
  --curve-output results/geva/risk_curve_DOMAIN.csv
```

The repair control runs `decide --no-challenge` on each initial, pre-challenge graph and evaluates without a repair log. The Spider veto control uses `--disable-operational-veto`; negative evidence remains in support and risk.

Adding this regime required registering three MathQA-specific producer names in `decide.PRODUCER_GROUPS` (`gate.exec_form` with the other hard gates, `lattice.derivable` and `exec.self_contradiction` with the cached benchmark-side probes). The graph also uses the shared `probe.answer_consensus` producer and `property.option_form`; all five appear in Table 5. The router group table is the only domain-specific data in the router. We re-ran the six existing regimes afterwards; their reported aggregate results were unchanged. Because no paired file-hash manifest is bundled, this check supports non-regression at the reported-metric level rather than byte identity.

Artifacts. The following table specifies the filenames emitted by the full research pipeline. The submission bundle includes the implementation and key aggregate JSON/CSV summaries, but not the complete per-instance JSONL files in the first six rows; those paths are therefore an artifact contract rather than a claim that every listed file is shipped.

Path	Contents
`{dom}_records.jsonl`	pools, leak-free features, sealed labels (T2)
`{dom}_graphs*.jsonl`	one provenance graph per instance (no sealed field)
`{dom}_kbprobe.jsonl`	pre-computed traversal facts (batch mode)
`{dom}_dec_*.jsonl`	decisions, action trace, cost, trust state
`{dom}_repairs_*.jsonl`	challenge log with post-hoc repair grades
`{dom}_judge.jsonl`	recorded cross-family verdicts (replayable)
`{dom}_reliability.json`	the certificate audit and its retain/withdraw calls
`{dom}_eval_*.json`	routed accuracy, false-trust, AURC, bootstrap
`nested_scale_8_16_32.json`	strict-prefix metrics and paired intervals
`nested_scale_*_manifest.json`	prefix, greedy and graph-reference checks
`metaqa3_heldout800_walkfix_paired.json`	walk-safe disjoint audit evaluation
`metaqa_certificate_witness_audit.json`	full gold-answer walk-premise audit
`repair_effect_paired.json`	initial-graph no-repair control
`spider_ablation_final_paired.json`	final-snapshot gate/veto controls
`risk_curves_final.csv`	all ranked risk–coverage points

The recorded filenames for the four ACQUIRE-KB-PATH variants of Table 6 are `metaqa3_dec_kbpath{,2,3,4}.jsonl` with matching `_eval_` files; variant (e) is `kbpath4`.

Cost accounting. Action utility uses probe/property 1, KB traversal 2, LLM probe 4, adjudication 6 and challenge 12. The saved `cost_units` counters stop before charging a terminal challenge; they record acquired evidence groups, not complete verification costs. For example, charging the 36 terminal MBPP+ requests at 12 would raise the reported mean from 1.38 to 2.52, even before separately measuring repair execution. We therefore retain the counters as replay-acquisition diagnostics, not evidence of end-to-end savings. Generation, graph preparation, calibration, repair calls

and re-validation require separate token/runtime accounting. Historical timings—47 s per 1,000 MetaQA-3hop traversals, about 200 s of MBPP+ graph construction and 87 s for Spider probes—come from different snapshots and workloads; they are not summed or substituted for a current pipeline measurement. Current traversal replay timing is reported in Appendix M.

K Scope: Transferability, Legitimacy, and the Second Base Policy

When does this transfer? The transferable design is the acquisition–management loop, not a fixed set of tools or a guarantee that certificates exist. New domains require producers with explicit source, dependence and validity contracts. Propositions 1 and 2 specify sufficient premises for two certificate types; their failure does not prevent ranking or evidence acquisition. KQA Pro can benefit from a judge without audit certificates, while MathQA-Geo shows that this fallback can also hurt. Dataset provenance determines certificate eligibility, not a producer's empirical precision or utility. A setting with only same-model consistency supplies much weaker evidence; no cross-domain accuracy or cost guarantee follows from using a graph.

Legitimacy of an acquisition action. Verification actions evaluate existing candidates: counterexamples identify defects without supplying target answers. Code and Spider repairs provide crashing inputs or SQLite errors, while ACQUIRE-KB-PATH preserves the claim count by assertion. In MetaQA-1hop, returning the KB's typed-tail set would disclose the gold answer set, so that set is used for verification rather than candidate generation.

Candidate selection for model self-improvement. The verification state could inform which generated candidates enter subsequent training or how they are weighted in verifiable-reward learning (DeepSeek-AI, 2025; Setlur et al., 2025; Liao et al., 2026). Definite counterexamples can identify candidates to reject, provenance can distinguish repeated support from independent evidence, and producer audits can identify evaluators that fail the chance comparison. These mechanisms suggest a way to inspect the selection signals before they influence a policy update. A claim emitted as unaccepted would not acquire a correctness certificate simply by being the highest-ranked candidate. Evaluating this use requires policy-update experiments that measure learning from the selected data and reassess verifier reliability as the generator changes. The present experiments evaluate frozen-policy inference, including repaired claim versions, and do not measure training gains.

The second base policy in full. Table 1 reports the common-pool Qwen3.6-27B comparison. MetaQA-3hop reaches 44.10 against common-pool Fixed V's 36.80 ($+7.30$ points). MetaQA-1hop reaches 40.80 against a 41.40 oracle, leaving 0.60 points; HumanEval+ matches Fixed V numerically at 90.85. KQA Pro reaches 67.10 against its 84.20 oracle. Both base policies are Qwen models, and second-policy KQA Pro uses Phi-4 rather than the first-policy Qwen judge. These additional resource configurations complement the separate archived-pool compatibility check below.

A frozen archived-pool compatibility check. We additionally evaluate the archived Mistral-7B-labelled pool on all $1{,}000$ MetaQA-3hop questions, with one greedy answer and eight samples per case. The existing MetaQA form gate, cached verifier, graph builder, migration rules, risk scorer and controller are reused without tuning ($\rho = .35$, seven steps). The adapter removes sealed fields and updates the generator identity; no new model request, judge or repair execution is used. Greedy is scored with the same KB verifier; sample scores remain those recorded with the pool. Decisions are frozen before evaluating any gold answer.

Under strict scoring, GEVA$-$Fixed V is $+4.50$ points (paired 95% CI $[+3.20, +5.90]$, exact $p = 4.36 \times 10^{-12}$; 47 improved, 2 harmed). The permissive difference is $+3.50$ ($[+1.80, +5.20]$, $p = .000103$). The result provides a supporting one-domain compatibility check for reuse of the frozen verifier pipeline on a Mistral-labelled pool. It is interpreted at the complete-pipeline level rather than as a selector-only or independent-generation estimate. Acceptance is 22.6% and strict conditional accepted error is 46.90%. We report model identity at the family level because the archived pool metadata records no checkpoint revision; original generation cost is outside this saved-pool evaluation. Code, manifests, frozen outputs and paired outcomes are in `code/results/geva/cross_generator_final_20260918/`.

Table 13: Mistral-labelled saved-pool compatibility check with common initial candidates, accuracy (%). Strict exact-entity matching is primary; the archived permissive scorer is separate. GEVA additionally acquires executed path evidence, so the comparison measures the complete frozen pipeline rather than selector-only effects.

Scorer	Greedy	Random	Majority	Fixed V	GEVA	Oracle
Strict entity	20.00	19.30	20.30	19.70	24.20	25.00
Archived permissive	26.50	26.50	26.60	27.00	30.50	37.30

L Cost, Pool Scaling, and Prior Sensitivity in Full

This section expands §4.5. The pool-scaling grid it discusses is Table 3, in the body.

Table 14: Measured CPU phases for the 1,000-case Mistral saved-pool run. Seconds from one final run, not a comparative speedup or complete inference cost. Fresh KB objects are loaded; operating-system cache state is uncontrolled. The total includes baseline selection and bookkeeping beyond the listed subphases.

Phase	Seconds
Input loading	0.065
Verifier KB loading	0.268
Traversal KB loading	0.505
Adapter and greedy verifier	17.342
Graph construction and migration	1.699
Routing including executed KB traversal	57.222
Decision serialization	0.036
Total saved-pool-to-decision	77.172
Offline sealed scoring (excluded from total)	0.037
Original generation time / tokens / GPU-seconds	unavailable

Table 14 adds measured work rather than converting action units to seconds. All 1,000 cases acquire KB evidence; the KB tool's own reported 51.869 s is nested within routing, not an additional charge. Protocol hashing precedes the timed region; statistical bootstrap evaluation is also excluded. No judge or repair model is called. The old MBPP+, MetaQA and KQA caches do not contain complete generation/verification usage logs, so tokens and GPU-seconds are unknown, not zero. A full-system comparison still requires new matched generation, judge and repair runs with concurrency, warmup, retries and cache state controlled. Table 3 is a strict nested-prefix control. For each case, one saved N=32 master pool supplies $S_8 = S_{32}[:8]$ and $S_{16} = S_{32}[:16]$; the greedy payload, case set and generation run are identical at all budgets. Every graph is rebuilt, and manifests assert prefix inclusion, identical greedy hashes and absence of claims or evidence from discarded samples. The paired intervals therefore condition on one saved generation run and isolate candidate-set growth; they do not estimate between-generation-seed variance. Figure 7 likewise draws one per-case permutation for each of three seeds and takes nested prefixes at $N = 1, 2, 4, 8$.

MathQA-Geo gains an unresolved 1.11 points from $N = 8$ to 32 (95% CI $[-0.67, 2.88]$, exact paired $p = .332$), while its requested-action charge stays near 5 units. It remains below Fixed V at every displayed pool size.

Common-threshold recorded-charge comparison. Table 4 compares full revelation with adaptive requests at the same $\rho = .35$. Adaptive requests lower recorded acquisition charges in five of six regimes; the MetaQA-3hop reduction is only 0.03 unit after rounding. HumanEval+ and Spider trade $0.61/0.48$ accuracy points for larger reductions, while KQA Pro's charge increases by 0.27 unit. Equal aggregate accuracy and AURC on MetaQA-3hop and KQA Pro do not establish identical emitted claims or traces. Joint false-trust is not uniformly improved. These replay counters omit generation, preprocessing, audit, terminal challenges, graph preparation and repair execution; they are not end-to-end costs.

The mean risk scores in Figure 7 change from .583 to .561 (MetaQA-1hop), .402 to .395 (3-hop), and .395 to .385 (KQA Pro) as N grows from 1 to 8. The limited movement describes the scoring rule's response to dependent evidence; it does not estimate a corresponding change in true error.

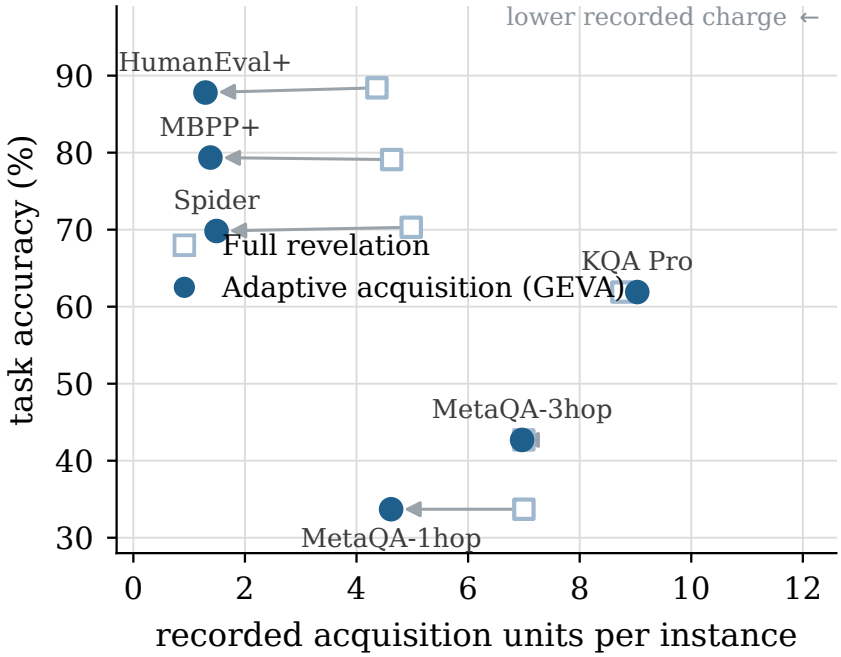


Figure 6: Accuracy versus recorded acquisition charge: full revelation (squares) and adaptive acquisition (circles), using Table 4's corrected-risk artifacts. Replay units exclude generation, preprocessing, audit, terminal challenges, graph preparation and repair; they are not end-to-end costs.

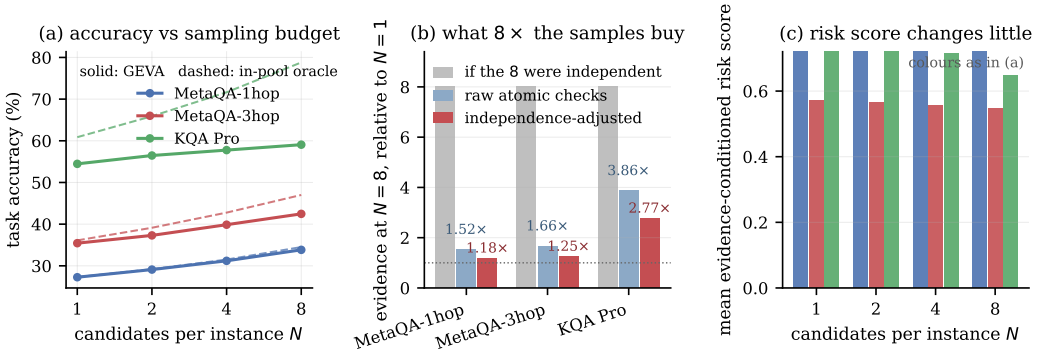


Figure 7: Strict nested-permutation sweep on saved $N{=}8$ knowledge-graph pools (§4.5). For each case and seed, one permutation supplies nested prefixes at $N = 1, 2, 4, 8$; the graph is rebuilt at every N, and three seeds are averaged. **(a)** the ceiling climbs faster than the selector. **(b)** raw versus independence-adjusted evidence. **(c)** the mean evidence-conditioned risk score barely moves.

The same conclusion at $4\times$ the pool. Table 3's within-master-pool paired differences from $N = 8$ to 32 are $+5.30$ points on MetaQA-1hop (95% CI $[+4.00, +6.70]$), $+6.90$ on MetaQA-3hop ($[+5.20, +8.70]$), $+2.40$ on KQA Pro ($[+1.00, +3.90]$), $+1.11$ on MathQA-Geo ($[-0.67, +2.88]$), and -0.29 on Spider ($[-1.26, +0.68]$). Thus the effect is resolved on the two MetaQA regimes and KQA Pro, and unresolved on MathQA-Geo and Spider even though Spider's oracle rises from 79.98 to 83.27. The action-charge column is nearly constant, but each charge covers an entire batch or

traversal; it excludes pool generation and graph preparation and therefore does not imply constant runtime or an end-to-end saving.

L.1 Current T1-Only Parameter Robustness

Protocol and reference check. We rerun the current batch-invariant, T1-only controller on all $5{,}027$ saved instances across seven regimes. Before perturbation, every emitted claim, acceptance flag, terminal action, reveal set, risk score and action charge must match the corresponding saved reference. All seven default accuracy, acceptance, false-trust and AURC values are reproduced. Each run reloads fresh graphs; KB and judge responses are revealed only on request from strict caches that raise on a missing entry. Thus routing is rerun, rather than applying a new threshold to fixed predictions. Saved code repairs and producer availability are held fixed; this measures robustness conditional on those resources, not regenerated repair trajectories.

The grid has 25 settings per regime: the default, six thresholds $\rho \in \{.20, .25, .30, .40, .45, .50\}$, four common offsets $\pm.10, \pm.20$ for each of weights and correlations, three independent-entry $U(-.15, .15)$ perturbations for each prior type (seeds $0, 1, 2$), and four joint weight/threshold settings ($\rho \in \{.25, .45\}$ crossed with weight offsets $\pm.10$). Weights are clipped to $[.05, 1]$ and correlation entries to $[0, 1]$; ancestor overlap is unchanged. A zero-valued cross-family prior is now preserved instead of incorrectly falling back to $.10$. The unperturbed constants are unchanged. No setting is selected as a new default. Three random seeds are a bounded sensitivity check, not a probabilistic guarantee over priors or generation seeds.

Table 15: Current T1-only accuracy (%) under parameter perturbations. Each range includes the default and all settings in that group; these are observed ranges, not confidence intervals. Joint varies weights and ρ together.

Regime	Default	Weights	Correlations	Threshold ρ	Joint
MBPP+	79.37	79.37–79.37	79.37–79.37	79.37–79.37	79.37–79.37
HumanEval+	87.80	87.80–87.80	87.80–87.80	87.80–87.80	87.80–87.80
Spider	69.83	69.83–69.92	69.83–69.83	69.83–70.02	69.83–70.12
MetaQA-1hop	33.70	33.70–33.70	33.70–33.70	33.70–33.70	33.70–33.70
MetaQA-3hop	42.70	42.50–43.30	42.70–42.70	42.70–42.70	42.60–43.20
KQA Pro	61.90	61.60–62.10	61.90–61.90	61.90–61.90	61.60–62.10
MathQA-Geo	86.92	86.47–86.92	86.92–86.92	86.92–86.92	86.47–86.92

Table 16: Across the same 25 settings: acceptance and joint false-trust (FT) ranges, AURC ranges, and minimum agreement with default emitted claim IDs / terminal actions. All except AURC are percentages. Range endpoints need not occur at the same setting.

Regime	Acceptance	Joint FT	AURC	Claim	Action
MBPP+	71.43–89.15	7.14–12.43	.0809–.0979	98.68	83.33
HumanEval+	90.24–92.07	4.88–4.88	.0312–.0356	100.00	98.17
Spider	68.38–97.39	15.47–28.14	.1988–.2118	99.13	78.53
MetaQA-1hop	0.00–14.10	0.00–0.70	.3166–.3187	100.00	89.30
MetaQA-3hop	0.00–30.60	0.00–12.10	.4485–.4555	99.10	80.80
KQA Pro	0.00–21.40	0.00–3.20	.1887–.2091	98.70	78.60
MathQA-Geo	0.00–37.03	0.00–1.33	.0642–.0707	98.23	62.97

Accuracy stability is not acceptance calibration. Weights affect support, effective trial counts and the acquisition penalty. On this grid, accuracy changes by at most $.60$ percentage points from default, while acceptance can change substantially. MetaQA-3hop accuracy stays 42.70% throughout the threshold-only sweep, but acceptance spans 0–30.60%; conditional accepted error, when defined, spans 38.08–39.54%. On Spider, threshold-only acquisition charges span 1.10–2.26 units. KQA Pro and MathQA-Geo have zero acceptance at $\rho = .35$ but nonempty accepted sets at higher settings. Zero joint FT with zero acceptance is not calibrated safety. Neither observed accuracy stability nor these operating-point changes justify selecting a threshold using the test outcomes.

Table 17: Historical semantic-prior sensitivity, before T1-only correction. "shift" adds a constant to every prior; "random" draws an independent $U(-.15, +.15)$ offset per entry (mean $\pm$ SD over 8–12 seeds). "claim" / "act." compare emitted claim IDs / terminal actions with the historical default. Weights affect both support and risk counts. Current results are in Tables 15 and 16.

		accuracy under a common shift				random $\pm.15$		
Regime	Prior	$-.20$	$-.10$	$+.10$	$+.20$	accuracy	claim	act.
MBPP+	ρ_f	79.63	79.63	79.63	79.63	79.63 ± 0.00	1.000	0.999
	w_f	79.63	79.63	79.63	79.63	79.63 ± 0.00	1.000	0.998
HumanEval+	ρ_f	87.80	87.80	87.80	87.80	87.80 ± 0.00	1.000	1.000
	w_f	87.80	87.80	87.80	87.80	87.80 ± 0.00	0.998	1.000
Spider	ρ_f	70.21	70.21	70.21	70.31	70.21 ± 0.00	1.000	0.996
	w_f	70.21	70.31	70.21	70.21	70.29 ± 0.08	0.999	0.981
MetaQA-1hop	ρ_f	33.70	33.70	33.70	33.70	33.70 ± 0.00	0.998	0.942
	w_f	33.70	33.70	33.70	33.70	33.70 ± 0.00	0.997	0.925
MetaQA-3hop	ρ_f	42.70	42.70	42.80	42.80	42.73 ± 0.05	0.999	0.981
	w_f	43.40	42.90	43.40	43.40	43.01 ± 0.48	0.971	0.701
KQA Pro	ρ_f	61.90	61.90	61.90	61.90	61.90 ± 0.00	1.000	0.992
	w_f	62.10	62.10	61.50	61.80	61.78 ± 0.26	0.989	0.945

Table 18: Historical pre-T1-only threshold sweep: accuracy / joint false-trust / acceptance rate (%). These are not current operating points; see Table 8. Each ρ reruns the controller. Zero acceptance gives zero joint false-trust but undefined conditional accepted risk.

Regime	ρ=.05	ρ=.20	ρ=.35 (historical default)	ρ=.50
MBPP+	79.37 / 0.00 / 0.0	79.63 / 10.32 / 81.7	79.63 / 12.17 / 88.4	79.63 / 12.43 / 89.4
HumanEval+	88.41 / 0.61 / 62.2	87.80 / 4.88 / 90.2	87.80 / 4.88 / 92.7	87.80 / 4.88 / 92.7
Spider	70.70 / 0.00 / 0.0	70.50 / 15.38 / 64.1	70.21 / 24.56 / 92.2	70.31 / 27.95 / 97.7
MetaQA-1hop	33.70 / 0.00 / 0.0	33.70 / 0.00 / 0.0	33.70 / 1.60 / 28.6	33.60 / 6.50 / 39.9
MetaQA-3hop	42.80 / 0.00 / 0.0	42.80 / 0.00 / 0.0	42.70 / 10.10 / 26.7	33.80 / 65.30 / 98.9
KQA Pro	61.90 / 0.00 / 0.0	61.90 / 0.00 / 0.0	61.90 / 3.80 / 23.5	59.50 / 19.80 / 55.3

Artifacts and scope. `robustness_review` saves every perturbed decision vector and all metrics, including conditional error and revealed groups, under `code/results/geva/robustness_20260920/`. Its protocol records the grid and input/module hashes. The 168 perturbed runs contain 120,648 case decisions; reference vectors are the separately hashed risk-fix files. Historical manifests remain descriptions of their original source versions, predating the zero-prior correction. No new model calls or physical tool executions occur here; replay seconds are not end-to-end costs. The grid does not exhaust individual-family perturbations, all hyperparameter interactions, second-generator settings or changes to the audit/repair policy.

Historical sensitivity of the semantic priors. Table 17 predates the T1-only, batch-invariant accounting correction and is retained as a diagnostic rather than a current headline result. In that snapshot, the family weights w_f and correlations ρ_f were fixed from producer semantics and never fitted. Re-running every regime under $\pm.10/\pm.20$ common shifts, independent $U(-.15, +.15)$ perturbations over 8–12 seeds, and the same two treatments on the weights (Table 17) leaves emitted accuracy comparatively stable but not invariant: it spans 42.20–43.40 on MetaQA-3hop and 61.50–62.30 on KQA Pro. Some perturbations are mildly *better* than the historical default priors, which is expected of untuned numbers and is why we do not present ours as optimal. The priors move purchases more than answers: random-weight action agreement is 70.1% on MetaQA-3hop while mean accuracy is 43.01 ± 0.48. The correlations matter little for the emitted answer for a structural reason: Eq. 1 charges each pair $\max(\rho_f, \mathrm{Jac})$, so on real graphs the *measured* ancestor overlap dominates the assumed family prior for most pairs.

Historical acceptance-threshold sweep. Table 18 predates the T1-only risk correction and is retained only to document the former operating-point sensitivity; current $\rho = .35$ results are in Table 8.

M Certificate Validity and Decision Ablations

Table 19 evaluates operational vetoes and candidate negative signals, including those excluded from the router. Table 21 compares the complete decision rule with relaxed risk and coverage gates. Exact calibration values for Figure 2 are supplied in `supplementary_data/calibration_cached.csv` and `supplementary_data/calibration_proof_chain.csv`.

Table 19: Veto flags and candidate negative signals (§3.3). Precision is the fraction of flagged claims that are indeed wrong, measured post hoc; no threshold in the pipeline reads it. Bold marks the maximal precision, 1.000. The table includes implemented flags and candidate signals. Sound-certificate eligibility follows the declared source and premises, not this empirical precision: code probes and cached reachability are veto-only even when precise; KQA Pro and MathQA-Geo signals are not operational vetoes. Corrected MBPP+ counts are omitted because the corresponding claim-level flag record is not included in this bundle.

Regime	Signal	Flagged	False pos.	Precision
MBPP+	exception on generated type-preserved probe	—	—	— (veto only)
HumanEval+	exception on generated type-preserved probe	26	0	**1.000** (veto only)
Spider	raises on an official database instance	1153	0	**1.000** (certificate)
Spider	unparsable SQL	9	0	**1.000** (certificate)
MetaQA-1hop	cached unreachable	6599	0	**1.000** (veto only)
MetaQA-1hop	no exact-1-hop walk	6599	0	**1.000** (certificate)
MetaQA-3hop	cached unreachable	571	16	0.972 (veto only)
MetaQA-3hop	no exact-3-hop walk	513	0	**1.000** (certificate)
KQA Pro	unreachable in the Wikidata subgraph	4179	1772	0.576 (chance)
MathQA-Geo	program lands on another legal option	667	28	0.958 (unsound)

Table 20: **Controls separating acquisition, selection and evidence management.** Frozen-pool comparisons, not additive contributions or uniformly single-component ablations. A/B follow the order named in each row; accuracy is in % and Δ is B minus A with 95% CI. Matched paths fix the final path observations and certificate flags, but not the entire acquisition schedule or total cost.

Comparison (regime)	Acc. A	Acc. B	Δ [95% CI]	Endpoint / scope
Fixed Selector (same path evidence) vs. GEVA (MetaQA-3hop)	41.30	42.70	$+1.40$ $[+0.53, +2.27]$	Alternative selector; $p = .002577$, not storage-only
No repair $\rightarrow$ repair (MBPP+)	77.78	79.37	$+1.59$ $[+0.53, +2.91]$	Candidate expansion: six improved, none harmed
Reactivated rows $\rightarrow$ lifecycle (MBPP+)	79.37	79.37	0.00 [0.00, 0.00]	All top-k errors equal; only two exposed cases
Relaxed gates $\rightarrow$ full policy (Spider)	67.70	69.83	$+2.13$ $[+0.97, +3.38]$	AURC .3383 $\rightarrow$ .2086; actions also change
Unaudited $\rightarrow$ audited judge (MetaQA-3hop, 800)	39.13	43.50	$+4.38$ $[+2.50, +6.25]$	Restores no-judge; accepted error unchanged

Matched evidence and metric controls. MetaQA's cached first-keyword heuristic can map "share directors ... which genres" to `directed_by`. Executed traversal supplies additional path/alignment observations. Both selectors receive exactly the same final path observations and certificate flags. The fixed selector does not use GEVA's risk/coverage aggregation, but uses a deterministic lexicographic ordering: candidates without an active premise-qualified path-absence certificate first, followed by path alignment, exact-hop reachability, cached verifier score and draw order. Table 20 reports 41.30% for the fixed selector and 42.70% for GEVA (GEVA$-$selector $+1.40$ points, 95% CI $[+0.53, +2.27]$, $p = .002577$). The comparison concerns selector choice using the same final path evidence and certificate flags; it does not isolate the causal effect of graph storage or acquisition. Static state semantics are evaluated separately in Appendix N.

The archived MetaQA scorer accepts normalised exact or either-direction substring matches. Re-scoring the same outputs by exact entity identity lowers 1-hop from 33.70 to 31.10 and 3-hop from

42.70 to 41.70; the latter changes $10/1000$ outputs. On Spider, relaxing risk and coverage gates reduces accuracy by 2.13 points (95% CI $[-3.38, -0.97]$, exact $p < .001$) and raises AURC from .2086 to .3383. Disabling only the operational veto leaves paired correctness unchanged.

Repair and candidate-space expansion. We route the initial, pre-challenge graphs with challenge disabled, excluding every repair claim and its derived evidence, and compare them with the full system using the same initial pool and remaining parameters. On MBPP+, accuracy rises from 77.78 to 79.37: $+1.59$ points, 95% paired CI $[+0.53, +2.91]$, exact $p = .031$. Six instances change from wrong to correct, none from correct to wrong, and 372 are unchanged. Relative to the common-pool Fixed V used in Table 1, the no-repair system is $+1.32$ points (95% CI $[0.00, +2.65]$, $p = .125$), while the repaired system is $+2.91$ ($[+1.06, +4.76]$, $p = .00342$). HumanEval+ remains 87.80 with or without repair. A further matched replay holds the saved repair claims and budget fixed but reactivates invalidated/consumed evidence; on MBPP+ all 378 correctness outcomes remain unchanged. Thus this snapshot supports candidate-space expansion, but does not identify a separate accuracy gain from lifecycle bookkeeping alone.

Table 21: Final-snapshot Spider decision ablations. Values are accuracy / joint false-trust / AURC. Gate relaxation sets ρ=1, $C_{\min}$=0 while retaining vetoes. Veto-off keeps every negative observation in support and risk, but removes its ranking, acceptance and repair veto effects.

Regime	Full GEVA	Risk/coverage gates relaxed	Operational veto off
Spider	69.83 / 24.47 / .2086	67.70 / 32.30 / .3383	69.83 / 24.47 / .2086

Gate relaxation costs -2.13 points (95% CI $[-3.38, -0.97]$, exact $p < .001$). Operational veto-off has identical reported aggregate metrics and zero paired correctness difference relative to the full final run. This ablation detects no accuracy effect of the binary veto path at this operating point; it does not establish instance-wise identity of emitted claims or traces.

N Common-Pool and Provenance Controls

Common initial candidate eligibility. We re-evaluate the saved candidates with a nonempty greedy anchor followed by all $N = 8$ samples, retaining draw multiplicities. Verifier ties and majority ties prefer the earliest entry (greedy first). Random selection uses seed 2026 and the original record order. This changes baseline eligibility, not GEVA decisions, tool evidence or repair candidates. A 10,000-replicate paired percentile bootstrap uses NumPy's `default_rng(2026)`; p uses the exact paired sign test. These nominal tests are unadjusted for multiple comparisons. Source SHA256 hashes and per-case outcomes are in `code/results/geva/common_pool_review_20260923/`.

Table 1 reports Greedy, Majority and Fixed V alongside GEVA. Random selection uses up to nine entries: 12 MathQA cases have empty greedy payloads. For MBPP+, HumanEval+, Spider, MetaQA-1hop, MetaQA-3hop, KQA Pro and MathQA-Geo, respectively, Rand.@9 accuracy is $65.34/79.27/63.73/23.20/31.10/50.30/69.62\%$ under the 9B policy. For the first six regimes under the 27B policy, it is $77.51/88.41/71.86/33.10/34.40/60.20\%$. The original resource-configuration summaries remain in the result artifacts; pool-headroom references are shown in Table 3.

The 9B MBPP+ difference has 12 improvements and one harm ($p = .00342$), but removing repairs reduces GEVA to 77.78% and its difference to $+1.32$ ($p = .125$). Thus common initial eligibility does not remove the repair-resource advantage. Spider's 9B difference has nominal $p = .0114$; the MetaQA-3hop differences remain resolved, whereas both MetaQA-1hop differences are -0.10 with $p = 1$. KQA Pro requires rescoring greedy with the original KG verifier before a matched Fixed-V comparison can be claimed; missing scores are not replaced by zero or a weaker form gate. Its majority and random controls remain available.

Intervention exposure, not just task count. The current post-repair MBPP+ graph file contains 378 cases, but only two have consumed evidence (eight surviving rows total), and no row is marked invalidated. The consumed rows have zero pass strength. Re-enabling them changes the selected claim's risk only on `mbppplus/448`, from .349800 to .396574, without changing its emitted repair or

acceptance. All 378 emitted claim IDs and acceptance flags agree across the saved runs. Five rank positions change, all containing wrong answers, so every top-k selective error is identical, not just aggregate AURC (.09537149 in both runs). This is weak exposure to disclosure and no exposure to supersession, not evidence that dynamic lifecycle management is generally ineffective.

The graph has 16 consumes edges but only eight surviving consumed rows; mutable row replacement means edges and terminal flags are not a lossless event sequence. We do not reconstruct missing disclosure or repair histories from these snapshots. The per-case exposure audit and all 378 matched-coverage points are saved in `code/results/geva/provenance_review_20260918/`.

A same-semantics flat view. We replace evidence lookup with a flat row scan, retaining the same migrated fields, visibility filters, batch identifiers and scorer. Across 17,730 claim/view assessments in five fixed reveal prefixes of the 378 saved graphs, every trust state and ranking agrees. This establishes static representation equivalence for these queries, not an independently implemented lifecycle engine or a runtime advantage. The learned ledger in Figure 3 is a different experiment: it uses a feature extractor and supervision, not this same-semantics adapter.

N.1 Prospective Events and Dependency-Local Revalidation

Implemented extension and boundary. The legacy snapshot controls above do not reconstruct missing history. Instead, an opt-in journal now starts from a complete, migrated graph and records subsequent claim/evidence revisions, consumption, invalidation, revalidation links, artifacts and tool runs. Sequence numbers, before/after images and a hash chain make revisions replayable; hashes detect inconsistent supplied records, not malicious replacement of an entire log. KB calls record their actual inputs and response facts. Challenge instrumentation records the rendered prompt, request attempt and raw response; a request exception has unknown delivery status. Bookkeeping evidence IDs are not claimed to identify every span actually shown in the prompt. No live model-based repair was run in this extension experiment. A sealed-key guard rejects T2 fields. Unrecorded node mutations fail the journal finalization check; metadata and direct edge writes are captured at the action boundary, not assigned invented earlier times. Logging is off by default and does not alter legacy graph serialization.

Contracts and dependency closure. Evidence applicability is separate from its correlation ancestry. The experimental replay mode permits explicit `validity_requires` edges to observation IDs; opaque ancestor strings alone never authorize validity propagation. Missing and cyclic dependencies are rejected. Consumption, invalidation or replacement revokes transitive dependents; a claim revision revokes its old checks. A fresh dependent observation can bind to the current valid parents, but an annotation or a revalidation link does not resurrect old support. A reverse index identifies affected owner claims. Their complete nonlinear batch/family statistics are recomputed, rather than subtracting an assumed additive evidence contribution. Global scoring-parameter changes invalidate all cached trust states. All candidates are still re-ranked when a decision is requested. This is an opt-in extension; it is not retroactively applied to Table 1.

Proposition 4 (Local recomputation equivalence)**.** *Fix the scoring configuration and a deterministic claim-local trust function. Assume complete, acyclic validity dependencies and a complete ordered mutation log. If an update invalidates its transitive dependent observations and recomputes every owner claim whose effective evidence changes (both owners on reassignment), then caching all other claim states gives the same states and decision ranking as full recomputation.*

Argument. Initially both methods compute identical states. At each update, unchanged claims have identical effective evidence and hence identical trust. Every changed claim is recomputed from its entire effective evidence set. This preserves equality inductively; the same deterministic ranking and acceptance predicate preserve decisions. This is an implementation-equivalence property, not a correctness or calibration theorem. Dependency traversal costs $O(|D| + |E_D|)$ for the affected closure; scoring costs remain those of the affected claim folds, and ranking is not eliminated. Dense dependencies can make the closure global.

Prospective KB protocol. We execute one fresh KB traversal for each of the first 100 lexicographically sorted MetaQA-3hop IDs, using saved pre-traversal graphs and no sealed outcomes. This bounded, non-random cohort measures software behavior, not population accuracy. Before running,

the artifact records the cohort rule, three rotating-order timing repetitions and source hashes. There are 3,108 events and 1,056 direct invalidations across 99 cases; this captures real supersession, unlike the low-exposure MBPP+ terminal snapshots. The legacy KB producer supplies these direct invalidations; transitive explicit requirements are exercised separately in the controlled fixtures, not inferred into legacy KB graphs. Every event-level trust state and decision agrees across full, indexed incremental and flat incremental replay. Every terminal graph hash matches the newly recorded execution, and recorder-disabled evidence and rankings agree.

Table 22: New KB event replay, 100 cases. Counts include initialization. Times are sums of per-case medians over three rotating-order repetitions and include replay initialization, event application and final selection, but not tool execution, file I/O or correctness comparisons. Action-boundary batching does not expose each intermediate state. No row avoids any of the shared 100 KB acquisitions.

Assessment schedule	Claim recomputations	Replay seconds	State availability
Full after every event	28,872	2.667	Every event
Indexed incremental	3,708	0.887	Every event
Flat incremental	3,708	0.890	Every event
Full at action boundaries	1,800	0.688	Initial and final

Indexed incremental replay reduces per-event claim folds by 87.16% and has similar timing to the flat implementation. Action-boundary batching is fastest when intermediate states are not required. The comparison separates update granularity from representation. Capture takes 5.409 s including about 5.028 s of traversal; finalization/replay validation adds 1.017 s and KB loading 0.361 s. Journals occupy 15,335,006 bytes (14.62 MiB). Raw phase timings and per-case states are retained; the replay timings exclude acquisition and do not measure end-to-end savings.

Actual tool reruns in controlled SQL fixtures. Eighteen handwritten in-memory SQLite conditions cross $N = 8, 32, 128$ candidate queries, change fanout $1, N/4, N$, and source replacement versus disclosure. Dependencies are known by construction. For source replacement the declared affected tables are updated; for disclosure, values need not change but independent-use eligibility does. A no- propagation baseline leaves all affected derived rows active. The propagated closure identifies exactly the affected queries. Re-running those and retaining unaffected cached values agrees with re-running all queries in every condition. At $N = 128$, this requires 1, 32 or 128 queries versus 128 for full revalidation, so dense change removes the saving. These counts exclude the common initial executions; raw artifact counters include all seven alternating-order timing repetitions. Query timings are microbenchmarks, exclude planning/logging, and do not establish end-to-end speedups. Crucially, re-running a disclosed test does not restore statistical independence: fresh private checks would still be needed. No fixture result is reported as benchmark accuracy, and no independent LLM repair or calibration benefit is inferred.

Reproduction. `event_experiment` accepts `-graphs`, `-kb`, `-limit`, `-repeats` and a new `-output` directory. The protocol, module hashes, complete journals and SQL conditions are in `code/results/geva/event_upgrade_20260918/`. These experiments cover one live KB action per case and controlled SQL revalidation. A separate limited model-repair pilot is reported in Appendix N.3; independent-policy closed-loop quality and cost comparisons remain untested.

N.2 Controlled Cache Revision with Recorded Repair Candidates

Question and fixed protocol. We test whether lifecycle updates affect downstream selection when every policy receives the same observations. The input is all 36 MBPP+ problems with four saved single-round repair candidates each. Every v1 candidate separately replaces its v0 parent in the original nine-slot pool, giving 144 episodes grouped into 36 problems. Other candidates and draw-order tie breaks remain fixed. These are sibling substitutions, not four sequential repairs. The low accuracies below describe this selected repair cohort and replacement protocol, not the full 378-problem benchmark.

The protocol is saved before scoring. Each episode starts with the recorded pre-repair state. After replacing the parent's code/version, responses arrive in the order: public gate, benchmark probe,

recorded disclosed replay, held-out probe, then remaining individual LLM-probe and purity rows. Missing responses add no evidence. This is a controlled arrival schedule for previously measured observations, not a reconstruction of historical request times. No new model or tool execution occurs, and no new disclosure to a solver is asserted. Public and private evidence retain their original visibility; only the eight recorded consumed rows, appearing in eight episodes, produce consumption events. New sibling-consensus rows are excluded because the four alternatives are not jointly present in one episode.

Policies and validity contracts. All policies use the unchanged scorer and gates ($\rho = .35$, $C_{\min} = .60$). *Stale* retains historical observations alongside arriving ones and ignores consumption and invalidation. *Overwrite* replaces a responding producer's own rows, including its old counterexamples, and honors direct consumption; other cached rows remain until that producer responds. *Lifecycle* additionally revokes the dependency closure after a version change. Individual observations depend on their owner's code/version, while cached consensus depends on all participating pool versions. These added dependencies leave initial scores and rankings unchanged in every episode. An independent *flat-contract* implementation scans version hashes and uses the same scorer; its active evidence and decisions agree with lifecycle at every checkpoint. This control tests evidence-use contracts rather than a graph storage advantage.

Metrics and independent unit. Outputs are frozen before joining the saved plus-suite outcomes, with exact code-identity checks for both original and repair candidates. Wrong acceptance is accepted-and-wrong per episode; conditional accepted error is undefined with no accepted episodes. AURC orders outputs by risk score, then problem and repair ID. Stale-support rate is the fraction of selected claims with at least one active positive risk-eligible row that violates the declared version/disclosure contract; it is not a correctness error rate. The primary endpoint is after all available individual responses. Intervals resample 36 problem clusters, preserving all four siblings, for $10{,}000$ paired replicates (seed 2026); AURC is re-ranked within every resample.

Table 23: Primary endpoint of the controlled repair-cache replay. Accuracy, acceptance, wrong acceptance (WA) and stale-support rate (SSR) are percentages over 144 episodes. All rows use identical candidates and arriving evidence.

Policy	Accuracy	Accept	WA	AURC	SSR
Stale reuse	18.75	0.00	0.00	.8566	90.28
Producer-local overwrite	15.97	0.00	0.00	.8809	83.33
GEVA lifecycle	18.75	0.00	0.00	.8515	0.00
Flat with same contracts	18.75	0.00	0.00	.8515	0.00

Lifecycle changes 112 selected slots across 31 problems relative to stale reuse, with four improved and four harmed episodes. Relative to overwrite it changes 18 slots across five problems, with four improved and none harmed. The clustered accuracy differences are 0.00 points (95% CI $[-8.33, +8.33]$) versus stale and $+2.78$ ($[0.00, +8.33]$) versus overwrite. AURC differences are $-.0050$ ($[-.1962, +.1641]$) and $-.0293$ ($[-.2014, +.0877]$), respectively. These data show sensitivity of decisions to cache validity, but do not resolve a predictive improvement over the comparators. Zero final wrong acceptance reflects zero acceptance, not calibrated selective safety.

Intermediate decisions. All pre-intervention policies agree. At the prespecified checkpoint immediately after the new public gate, overwrite accepts $10/144$ episodes, of which three are wrong; lifecycle accepts none. This is a coverage difference, not a matched-coverage safety claim. Later checkpoints and all four policies are retained in the artifact, including stages where the ordering of AURC values changes. The final endpoint is not replaced by the most favorable intermediate checkpoint.

Interpretation and reproduction. This experiment increases exposure to version replacement and dependent cache entries, while disclosure exposure remains limited to the recorded flags. The latest individual responses do not refresh all peer-consensus observations; the comparison therefore addresses partially revalidated caches. A simple flat implementation with the same validity contracts reproduces GEVA. A live shared-trajectory pilot follows; independent online policy utility remains a

separate question. The permanent entry point is `code.eval.codeplus.geva.lifecycle_utility` with `-research-root` and a new `-output`. Protocols, source hashes, complete shared event bundles, frozen decisions, all checkpoint outcomes and clustered intervals are in `code/results/geva/lifecycle_utility_clustered_20260919/`. The run uses no new model calls and does not estimate the original generation or verification cost.

N.3 Prospective Shared-Trajectory Repair Pilot

Protocol and evidence boundary. A separate feasibility pilot executes new model calls on 24 MBPP problems, selected from all 378 by a fixed hash ordering of seed 20260920 and case ID, without outcome filtering. Mistral-7B-Instruct-v0.3 (Mistral AI Team, 2024) generates two initial candidates per problem and up to two sequential repairs ($T = .8$, top-$p = .95$, 768 output-token limit). A common driver repairs the candidate passing the most base assertions, breaks ties by draw order, and stops if a candidate passes them all. Each request includes the current code, a failing assertion and that lineage's previously disclosed assertions. Fifteen problems reach two repair rounds, giving 30 repair requests. These are actual revisions, not sibling substitutions, but all state policies observe one shared trajectory rather than controlling separate rollouts.

Only public problem fields reach capture. The initial prompt shows the first base assertion; other *public* base assertions are withheld from this context until needed. Their public availability and possible training exposure preclude claims of fresh private or statistically independent tests. No Plus inputs or outcomes enter capture. Outputs are frozen before separate evaluation on at most the first 80 Plus checks per program. This small adapter is not the full headline MBPP harness.

State policies and results. Every checkpoint re-executes the base assertions on both current branch tips. Stale reuse retains obsolete rows. Version caching checks code hashes and disclosure eligibility; clear-and-recheck retains current-version observations; GEVA propagates explicit validity dependencies; the same-contract flat view checks them by scanning. All use the unchanged scorer and gates. The last completed checkpoint is the fixed primary endpoint. Table 24 shows no observed predictive advantage: GEVA changes 11 final code selections relative to stale reuse, but no correctness outcomes, and matches version caching's final selections. Initial accuracy is also 37.50%. With no observed paired correctness differences, the bootstrap interval degenerates and does not establish population equivalence.

Table 24: Prospective pilot, 24 problems. Accuracy and invalid-use rate are %. Invalid use means the selected claim has an active check violating its code/disclosure contract, not necessarily positive support. All acceptance rates are structurally zero.

State view	Accuracy	AURC	Invalid use
Stale reuse	37.50	.2699	16.67
Version cache	37.50	.2699	0.00
Clear-and-recheck view	37.50	.2699	0.00
GEVA lifecycle	37.50	.2699	0.00
Same-contract flat	37.50	.2699	0.00

Costs and interpretation. The shared run makes 54 model requests, executes 326 assertion checks and grades 76 distinct programs. Warm capture takes 197.389 s, including 166.618 s waiting for model requests and 29.721 s for assertion subprocesses. Usage is 11,815 input plus 19,442 output tokens; sealed evaluation adds 9.306 s separately. Model startup, hashing and the health check are excluded; existing GPU workloads remain active. These are single-run shared costs, not per-policy savings or exclusive GPU-compute time. Clear-and-recheck is a state view, not a separately timed deployment.

Only specification-conformance evidence is supplied, so coverage is at most .2 against $C_{\min} = .6$: zero acceptance is not a safety result. A post-run diagnostic finds top-level assertions in $67/76$ generated programs; some fail during loading, which limits interpretation of semantic repair quality. The fixed extractor and results are not changed afterward. Thus this pilot validates capture and exhibits validity-sensitive selection, but does not establish added predictive utility over simple version caching. Fresh independent verification and separate-policy closed-loop quality–cost comparisons remain untested. The `natural_repair` entry point separates `prepare`, `capture` and `evaluate`; protocols, model/source hashes, raw responses, event journals and outcomes are in `code/results/geva/natural_repair_pilot_20260920/`.