VERITASMATH: SCOPED EVIDENCE AND BOUNDED AUTHORITY FOR LLM MATHEMATICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generated verifiers for large language model (LLM) mathematical reasoning can execute successfully while checking the wrong claim, omitting constraints, or covering only a finite subset of a universal statement. Reliability therefore depends not only on whether verification occurs, but also on what each piece of evidence is authorized to conclude. We introduce VERITASMATH, an evidence-constrained reasoning architecture that operationalizes bounded epistemic authority through claim-level evidence contracts and an authority-constrained state machine. Each verification record captures the checked claim, polarity, domain, method, coverage, and provenance; executable conflict re-computation, selection-only arbitration, repair re-entry, and answer-locked rendering constrain how evidence can affect candidate selection and output, while an obligation graph tracks whether all requested deliverables are supported. On a pre-specified, latency-capped six-benchmark subset spanning final-answer and proof tasks, VERITASMATH achieves a 96.01% problem-weighted mixed aggregate, compared with 92.42% for code-based self-verification, a 3.59-point improvement. In fixed-trajectory replays, the same contracts are associated with lower unsupported promotion and payload mutation, and with higher deliverable completeness.

1 INTRODUCTION

Large language models (LLMs) solve increasingly difficult mathematics through chain-of-thought prompting (Wei et al., 2022), repeated sampling (Wang et al., 2023), and tool-integrated reasoning (Gao et al., 2023; Chen et al., 2023; Gou et al., 2024b). As inference-time scaling produces increasingly many plausible candidates, however, generating a good solution and reliably selecting one become distinct problems (Brown et al., 2024). Reliable selection requires reasoning not only about whether a verification signal passes or fails, but also about what claim it actually supports and over what scope. This distinction separates three related but non-equivalent notions: **correctness**, the mathematical validity of a conclusion and its derivation; **evidence support**, the extent to which available checks substantiate particular claims over their stated domains; and **deliverable completeness**, whether all obligations posed by the problem are adequately addressed. Conflating these notions can lead to misleading verification outcomes: a correct final value may omit a required proof, a complete response may contain an invalid step, and a passing test over finitely many instances does not by itself establish a universal claim.

Verify-and-repair pipelines often convert a local signal directly into a global action. Agreement may come from a shared stem misreading. Generated code may print `PASS` without evaluating the asserted variable, omit a constraint, or use finite tests to support a universal claim. A free-form judge may combine candidates into a new answer, and a repair step may damage a correct payload. Presentation and order effects further perturb model-based judging (Zheng et al., 2023; 2024). These failures share one cause: evidence receives more decision authority than its scope warrants.

Our central principle is *bounded epistemic authority*: evidence polarity, scope, and coverage determine which action a module may take. VERITASMATH realizes this principle with a claim-level evidence ledger and an authority-constrained state machine. Executable checks append records instead of issuing final verdicts. Conflict resolution prioritizes re-computation; semantic arbitration

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.

is selection-only; a repair is a new candidate with no inherited support; and the renderer copies an answer-locked payload. An obligation graph links each required deliverable to the claims and evidence that cover it. The architecture supports two output policies. Fail-closed mode emits a payload only in the SUPPORTED state. Best-effort mode may expose a PARTIALLY_SUPPORTED or UNRESOLVED payload together with its open obligations. This makes reliability and coverage an explicit operating tradeoff instead of an implicit fallback decision. The headline comparison and all component ablations use the fail-closed policy; best-effort appears only as a deterministic sensitivity replay of the mechanism cohort.

Our contributions are:

- **Evidence representation:** a claim-level contract that records scope and provenance and maps evidence classes to permitted actions;
- **Authority transition design:** a state machine that separates generation, refutation, selection, repair, and rendering, with repair re-entry and an answer-lock invariant; and
- **Conditional mechanism evaluation:** an obligation-aware, fixed-trajectory replay that measures false support, erroneous promotion, repair damage, payload mutation, completeness, and selective risk.

We instantiate the architecture on COMP-8, a suite of eight competition benchmarks covering both final-answer and proof-based tasks. The full system is evaluated on all eight benchmarks, while scaffold comparisons are conducted on a pre-specified controlled six-benchmark subset. Together, the system-level results characterize deadline-bounded end-to-end performance, and the paired mechanism replays, conditioned on fixed candidates and checker outputs, provide conditional estimates of individual policy decisions within each trajectory.

2 RELATED WORK

Candidate generation and verifier-based selection. Self-consistency and inference-time scaling allocate computation to candidate generation and selection (Wang et al., 2023; Brown et al., 2024; Snell et al., 2024). Program-aided reasoning and code-based self-verification execute checks (Gao et al., 2023; Chen et al., 2023; Gou et al., 2024b; Zhou et al., 2024); related training uses verifiable rewards, self-verifiable reasoning, process supervision, and generative verifiers (DeepSeek-AI, 2025; Shao et al., 2025; Lightman et al., 2024; Wang et al., 2024; Zhang et al., 2024). VERITAS-MATH instead encodes each check as scoped evidence with explicitly bounded authority.

Tool feedback and verify–repair loops. Debate, AutoGen, MACM, MALT, and ReMA coordinate specialized or interacting agents (Du et al., 2024; Khan et al., 2024; Wu et al., 2023; Lei et al., 2024; Motwani et al., 2025; Wan et al., 2025). ReAct, Tree of Thoughts, Reflexion, Self-Refine, and CRITIC combine search, feedback, iterative revision, or external tools (Yao et al., 2023b;a; Shinn et al., 2023; Madaan et al., 2023; Gou et al., 2024a). AMTFV separates verification planning from execution for adjudication and revision, while error localization supports correction (Zou et al., 2026; Tyen et al., 2024). Our critics identify claim-linked defects; repairs create new candidates that are revalidated without inheriting prior verdicts.

Claim-level and fixed-pool reliability. Claim-Level Reliability Assessment targets decision-critical claims and falsification; Oracle Gap and Signal Fidelity diagnose fixed-pool selection and harm; Hard2Verify provides human step-level annotations for open-ended mathematics (Xu et al., 2026; Hu, 2026; Pandit et al., 2025). We integrate these ideas with polarity-aware evidence-to-action rules, domain and provenance records, repair re-entry, obligation dependencies, and immutable output.

Constrained neuro-symbolic control. LLM-Modulo places an LLM under critics and hard constraints (Kambhampati et al., 2024). We instead address fallible generated or model-based verifiers that may check the wrong object or share generator errors: claim-level uncertainty and separate authority limits govern rejection, selection, repair, and rendering. Randomized order and delimited inputs reduce presentation effects (Zheng et al., 2024).

Proof, process, and completeness evaluation. MATH, OlympiadBench, undergraduate benchmarks, and MathArena/Apex span distinct mathematical regimes (Hendrycks et al., 2021; He et al., 2024; Xu et al., 2025; Chernyshev et al., 2025; Balunović et al., 2025; Dekoninck et al., 2025a; 2026). Proof evaluation reveals failures missed by answer matching, while ProcessBench targets trace-level error localization (Petrov et al., 2025; Dekoninck et al., 2025b; Zheng et al., 2025). We therefore report final-answer accuracy and normalized proof credit separately from evidence support and obligation coverage, and validate the automatic audit against human annotations.

3 METHOD

3.1 PROBLEM SETTING AND RELIABILITY TARGETS

Given a problem statement q, the system generates candidates under wall-clock and token budgets $T_{\max}$ and $B_{\max}$. Candidate a_i contains an immutable payload, claims Γ_i, and provenance. The problem is compiled into an obligation graph $G_q = (O, R_O)$, where O contains requested deliverables and R_O their dependencies; for an extremal problem, the optimum requires both a global bound and an admissible attaining construction.

The system returns a pair (y, z), where y is an optional protected payload and

$$z \in \{\mathsf{SUPPORTED}, \mathsf{PARTIALLY_SUPPORTED}, \mathsf{UNRESOLVED}, \mathsf{ABSTAINED}\}.$$

SUPPORTED requires the selected candidate to satisfy the configured evidence and obligation policy; PARTIALLY_SUPPORTED exposes named gaps, UNRESOLVED an unresolved conflict, and ABSTAINED no payload. Fail-closed emits y only for SUPPORTED; best-effort may also emit the intermediate states with open claims and obligations. This is an operational state, not an unconditional truth claim: each key non-universal claim needs claim-linked, domain-admissible positive E2, each universal claim positive E4 or complete finite-domain coverage, and every necessary obligation must be covered. An admissible E3 counterexample blocks its linked claim. Optimality requires a supported bound and checked attaining construction; "determine all" requires valid instances and exhaustiveness; each multi-part component is a separate obligation.

3.2 CLAIM-LEVEL EVIDENCE CONTRACTS

Each check appends an evidence record

$$e = \langle c, Q, D, M, K, P, R, S \rangle, \tag{1}$$

where c is the checked claim, Q its logical form, D its domain and constraints, M the method, K coverage, P provenance, R result, and S evidence class. Provenance identifies the source candidate, checker, executable artifact, input, and, when applicable, random seed.

Evidence classes describe what a check establishes jointly with R, rather than a scalar strength: E0 is no valid check (including timeout or parse failure), E1 is successful execution without claim linkage, E2 is a claim-linked finite or heuristic test, E3 is a replayable counterexample in the claim domain, and E4 is a complete finite-domain check in the reported implementation. A literal `PASS` without claim-linked computation remains E1; for $\forall x \in D, \phi(x)$, a witness with $\neg\phi(x)$ can supply E3, whereas success on $S \subset D$ supplies at most E2.

No Lean, Coq, Isabelle, or other formal prover is connected, so E4 requires the enumerated set to equal the finite claim domain, every element to pass, and a claim-linked, domain-admissible script. The router rechecks the verifier's quantifier, domain, coverage, linkage, and result before deriving E0–E4; missing or malformed fields become E0/inconclusive. Benchmark answers are unavailable until final scoring.

An authority policy maps evidence and state to allowed actions:

$$\mathcal{A}(e, s) \subseteq \{\texttt{retain}, \texttt{reject-claim}, \texttt{select}, \texttt{repair}, \texttt{render}, \texttt{abstain}\}. \tag{2}$$

E1 cannot promote or refute a claim. Positive E2 may support selection of a non-universal claim, affect ranking, or trigger another check, but cannot certify a universal claim; negative E2 only signals conflict. E3 rejects only its linked claim, while either polarity of E4 applies only within the

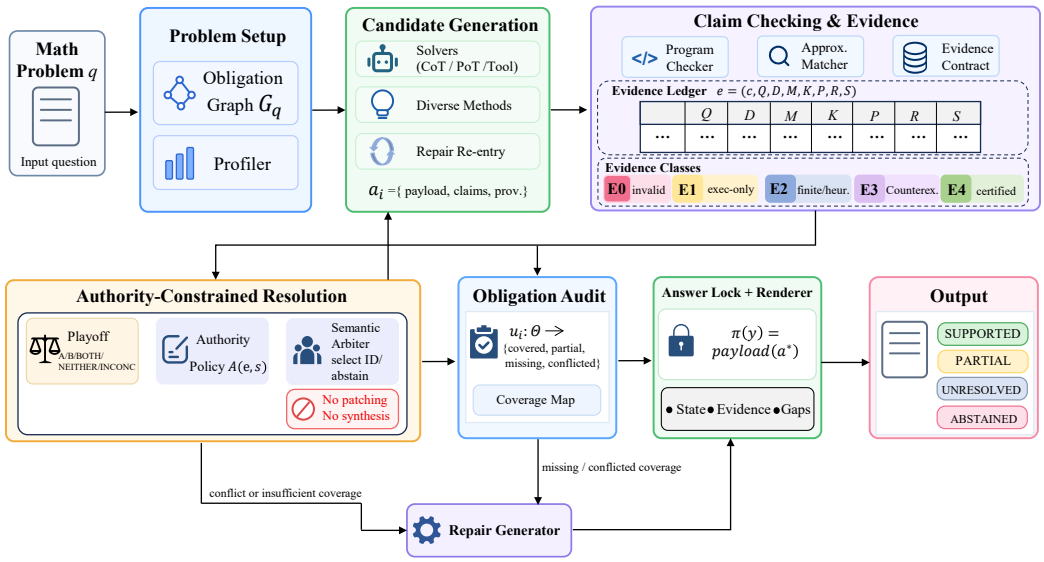


Figure 1: Overview of the VERITASMATH workflow. Diverse candidates are generated and checked through claim-linked evidence; executable conflict resolution and authority constraints govern selection and bounded repair. Obligation auditing tracks coverage, while answer locking preserves the selected payload and its support status during rendering.

Table 1: Authority matrix. C: create a new candidate; M: modify an existing payload; S: select a final candidate; T: terminate with an output state.

Module	C	M	S	T	Permitted output
Solver / repair generator	yes	no	no	no	New candidate with provenance
Program checker	no	no	no	no	Evidence record
Cross-validator / playoff	no	no	limited	no	Conflict state or existing candidate ID
Semantic arbiter	no	no	yes	no	Existing candidate ID or abstain
Obligation auditor	no	no	no	no	Coverage map and missing obligations
Renderer	no	no	no	yes	Locked payload and status fields

checked finite domain. Actions depend on class, polarity, linkage, domain, and coverage, without scalar aggregation.

The approximate matcher $\simeq$ contributes records rather than a global proof verdict. It combines normalized identity, symbolic simplification, bounded sampling, and labelled multi-part comparison, with each layer recording method and coverage. Appendix B specifies the matcher and sandbox.

3.3 AUTHORITY-CONSTRAINED STATE MACHINE

Figure 1 summarizes the VERITASMATH workflow. The profiler compiles an obligation graph and configures each problem before method-diverse solvers generate candidates. Claim checkers append scoped records to an evidence ledger; authority constraints route them to retention, rejection, selection, repair, rendering, or abstention. Executable conflict checks precede a selection-only arbiter, while obligation auditing and bounded repair address unresolved gaps. The answer lock then preserves the selected payload.

Table 1 separates the ability to generate mathematical content from the ability to select or terminate. The policy is enforced by typed outputs and deterministic routing, rather than by role instructions alone. Algorithm 1 gives the state transitions. The active candidate pool is round-local; earlier candidates remain in history for traceability and fallback. A repair hint names a failed claim or uncovered obligation and is consumed once.

Algorithm 1 Evidence-authority routing under a fixed budget

$G_q \leftarrow$ CompileObligations(q); $H, L \leftarrow \varnothing$
$\sigma \leftarrow$ ProfileOrDefault(q)
for $r = 0, 1, 2$ **do**
  if $\neg$SolveAffordable(σ) **then break**
  end if
  $C_r \leftarrow$ GenerateCandidates(q, σ); $H \leftarrow H \cup C_r$
  $L \leftarrow L \cup$ CheckClaims(q, C_r)
  $(a^\star, g) \leftarrow$ ResolveByAuthority(C_r, L)
  if $a^\star \neq \perp$ **then**
    $\mu \leftarrow$ AuditObligations$(G_q, a^\star, L)$
    if SupportsAll$(a^\star, \mu, L)$ **then break**
    end if
    $g \leftarrow$ MissingOrConflicted(μ, L)
  end if
  if $r = 2$ or $\neg$RepairAffordable(σ) **then break**
  end if
  $\sigma \leftarrow$ RepairContext(g) ▷ next output is a new candidate
end for
$(a^\star, z) \leftarrow$ ApplyOutputPolicy(H, L, G_q)
return RenderLocked$(a^\star, z, L, G_q)$

3.4 Conflict resolution and repair re-entry

When candidates disagree, conflict resolution first recomputes a check against the original constraints and records separate evidence for each candidate. Residual evaluation, feasibility checks, and finite enumeration are selected according to the claim. The playoff verdict is one of A, B, BOTH, NEITHER, or INCONC; the associated records determine whether the policy may select, reject a linked claim, request another method, or defer. BOTH does not collapse two candidates into an equivalence class, and NEITHER does not authorize an in-place patch.

If executable evidence does not decide the conflict, the semantic arbiter acts over randomized candidate identifiers. Writing $I(C_r)$ for existing IDs, its action space is

$$j \in I(C_r) \cup \{\perp\}, \qquad a^\star = a_j \text{ if } j \neq \perp. \tag{3}$$

The arbiter cannot write mathematics or combine candidates. Abstention and NEITHER may trigger a repair generator when budget remains. Its output is assigned a fresh candidate ID, an empty evidence set, and a provenance link to the hint; it then re-enters claim checking from the start.

Profiling, domain retrieval, method-diverse fan-out, sandbox restrictions, and resource lanes instantiate this state machine. They affect performance and cost but do not alter the authority contract. Their configuration appears in Appendices A and B.

3.5 Obligation coverage, answer lock, and budgets

For each candidate, the auditor builds

$$\mu_i : O \to \{\mathsf{covered}, \mathsf{partial}, \mathsf{missing}, \mathsf{conflicted}\}.$$

A necessary obligation is covered only when the corresponding response span is present and its linked claims satisfy the configured evidence policy. Rule-based and model-derived missing sets are unioned,

$$M(q, a_i) = M_R(q, a_i) \cup M_C(q, a_i), \tag{4}$$

so each reported gap remains available for targeted repair. The auditor emits a coverage map, not a replacement answer. Human-labeled obligation sets are used to evaluate response-level completeness in Section 4.1.

After selection, the router stores the answer, derivation, and qualifications as an immutable payload. The renderer may add the output state, evidence summary, and unresolved obligations. With π extracting the submitted payload,

$$\pi(y) = \mathrm{payload}(a^\star). \tag{5}$$

Table 2: Mechanism-level endpoints computed from response traces and human labels. Appendix F specifies the records and denominators.

RQ	Primary endpoints	Event measured
1	FSR, FRR, promotion precision	Unsupported claim promoted; valid claim refuted; selected candidate supported by admissible evidence
2	C→I flip, mutation rate	Correct candidate damaged by selection/repair; locked payload changed during rendering
3	ADC, CCR	Requested obligations covered; correctness and completeness on the same response
4	Selective risk, coverage	Error among emitted responses at the fixed fail-closed operating point

A mismatch returns the stored payload directly and records a mutation event.

A monotonic clock and token ledger gate every dispatch. Per-call timeouts are clamped to the remaining hard budget, while reserved quotas protect conflict resolution and final rendering. At exhaustion, APPLYOUTPUTPOLICY derives the output state from the ledger and obligation map. It never upgrades an unresolved candidate to SUPPORTED. Exact ceilings and loop bounds are listed in Appendix A.

4 EXPERIMENTS

4.1 EVALUATION QUESTIONS AND PROTOCOL

The evaluation separates system-level final-answer accuracy, normalized proof credit, and their mixed aggregate from mechanism-level reliability. The benchmark comparison measures the complete policy under a shared deadline. The controlled study tests the contracts introduced in Section 3.2 through four questions:

Q1. Does claim-level evidence representation reduce false support and false candidate promotion?

Q2. Do selection-only arbitration, repair re-entry, and answer lock prevent correct candidates from being damaged?

Q3. Does obligation tracking detect incomplete deliverables without triggering excessive or harmful repair?

Q4. How does evidence authority change the fail-closed coverage–risk operating point?

Benchmarks and scoring units. COMP-8 contains 177 problems across eight benchmarks; their problem counts and task families are listed in Appendix D. Final-answer sets are AIME 2025 and 2026, HMMT February 2026, Apex Shortlist, and Apex 2025. Proof sets are Putnam 2025, IMO 2025, and USAMO 2026. The retained protocol uses four attempts per problem, totaling 708 intended runs for the full VERITASMATH workflow. The same-checkpoint scaffold comparison was fixed before evaluation on the controlled six-benchmark subset of 99 problems (396 slots per compared method): AIME 2025, HMMT February 2026, Apex 2025, Putnam 2025, IMO 2025, and USAMO 2026. AIME 2026 and Apex Shortlist have VERITASMATH workflow and latency statistics, but no scaffold comparison because the shared evaluation quota was allocated to complete all methods on the controlled subset. The problem is the statistical unit; attempts are repeated measurements nested within a problem, and all four attempts are retained without selecting the best attempt.

Final-answer accuracy measures answer correctness; normalized proof credit measures partial credit. IMO and USAMO use a 0–7 score range, and Putnam is reported on its recorded normalized scale. Proof columns therefore represent normalized credit rather than binary proof-validity estimates.

System-level endpoint. For a fixed subset $\mathcal{B}$, the primary summary is the problem-weighted mixed aggregate $S_{\mathcal{B}} = \sum_b n_b s_b / \sum_b n_b$, where s_b is the mean percentage value for the benchmark's

scoring unit and n_b the problem count. The controlled six-benchmark subset has 99 problems with available scaffold comparisons. We report its final-answer and proof-credit strata separately. The original single-pass HMMT entry, 91.57%, is incompatible with a binary mean over 33×4 trials, so it and the dependent single-pass suite aggregate are excluded. Putnam is reported on its normalized proof-credit scale.

Models, baselines, and budgets. All scaffolds use the May 2026 snapshot of `Intern-S2-Preview-397B`. Each method is evaluated under a common 1,200-second per-run ceiling, while retaining the tool access and parallel-work configuration prescribed by its design. Baselines include single-pass reasoning, self-consistency with $k = 5$, program-aided solving, code-based self-verification, two-round Self-Refine, three-agent two-round debate, and an AutoGen-style assistant/executor pair. Appendix F reports the implementation fields needed to reproduce each baseline exactly.

The aggregate comparison is *latency-capped*, reflecting a shared end-to-end deadline; token, call, parallelism, and reserved-validation profiles are reported in Appendix D.1. The full system averages 3.96 calls and approximately 58,000 tokens per run, compared with approximately 2.8 calls and 40,000 tokens for code-based self-verification. To complement this system-level protocol, the mechanism study uses paired replay with the same generated candidates, checker outputs, and total generated-token budget across policy variants. This replay design estimates the conditional effect of each decision contract on fixed upstream artifacts, providing a controlled view of how evidence representation and authority rules shape downstream decisions.

Mechanism labels and metrics. Two blinded annotators decompose each sampled problem into obligations before viewing method identities, label claim validity and evidence coverage, and grade response correctness and completeness. A third annotator adjudicates disagreements. Obligation labels reach Cohen's $\kappa = 0.81$, and response-level ADC labels agree on 93.7% of responses (386/412). Separately from response coverage, obligation node detection attains 94.0% precision (612/651) and 91.2% recall (612/671) under one-to-one matching to annotated deliverables. False-support rate (FSR) is the fraction of promoted claim–evidence links judged insufficient; false-refutation rate (FRR) is the fraction of rejected claims judged valid. Candidate-promotion precision replays the declared evidence policy on adjudicated records for each selected candidate. Assessed deliverable completeness (ADC) requires all necessary spans and their linked evidence; partial, missing, or conflicted obligations fail. Correct-and-complete rate (CCR) additionally requires binary correctness on the same response. Repair outcomes are classified as incorrect-to-correct, unchanged, correct-to-incorrect, or unresolved. Payload mutation compares the stored selected payload with the rendered payload byte for byte after canonical framing is removed.

For output policy τ, coverage is the fraction of responses in the mechanism cohort that emit a payload, and selective risk is the error rate among those emitted payloads. The reported comparison uses one fixed fail-closed operating point: positive support requires at least E2 with valid claim linkage and domain scope, while a universal claim requires positive E4 evidence or complete finite-domain coverage. An admissible E3 counterexample instead refutes the linked universal claim and prevents it from satisfying the support predicate. Appendix F gives the denominators, confidence intervals, annotation protocol, and stress-test specification.

At the fixed fail-closed operating point, the mechanism cohort contains 412 responses: 382 are SUPPORTED (92.7%), 14 are PARTIALLY_SUPPORTED (3.4%), 9 are UNRESOLVED (2.2%), and 7 are ABSTAINED (1.7%). The resulting unconditional correctness rate is 90.0% (371/412), while the conditional correctness rate among emitted payloads is 97.1% (371/382); selective risk is 2.9% (11/382). These quantities characterize the enriched mechanism cohort and use distinct denominators by construction.

Paired statistical analysis. Given normalized response values $y_{ij}^{(A)}$ and $y_{ij}^{(B)}$, define $d_i = \frac{1}{4}\sum_{j=1}^{4}(y_{ij}^{(A)} - y_{ij}^{(B)})$. A paired cluster bootstrap resamples problems within each benchmark and retains all attempts and paired methods. We use 10,000 resamples. The reported 95% intervals for mechanism endpoints are policy-specific rates, not intervals for paired policy differences; adjusted tests of those differences are not reported. We therefore treat the observed contract-level gaps as descriptive conditional effects of replaying policies on fixed artifacts.

Table 3: Same-checkpoint, latency-capped scaffold comparison on the pre-specified controlled six-benchmark subset (four-attempt mean values, %). Final-answer accuracy (FA) and normalized proof credit (PC) columns are separated. Tool access, parallel work, token/call budgets, and reserved validation time are not compute matched. The six columns contain 99 problems (396 problem–attempt slots per method). '–' marks an excluded inconsistent value.

	Final-answer accuracy (FA)			Normalized proof credit (PC)		
Method	AIME 25	HMMT 26	Apex 25	Putnam 25	IMO 25	USAMO 26
Backbone, single pass	92.50	–	72.92	82.50	71.43	90.48
Self-consistency, k=5	94.17	92.42	75.00	84.38	73.21	92.26
Program-aided solve	95.00	93.18	77.08	85.42	73.81	93.45
Code-based self-verification	96.67	96.21	79.17	89.58	78.57	96.43
Self-Refine	93.33	92.42	75.00	83.33	72.62	91.67
Multi-agent debate	95.83	94.70	77.08	87.50	76.19	94.64
AutoGen-style pair	95.00	94.70	77.08	86.46	75.60	94.05
VERITASMATH (ours)	**100.00**	**100.00**	**83.33**	**92.71**	**82.14**	**100.00**

Table 4: Component ablations under the fixed fail-closed policy and latency ceiling. Realized compute is not matched. Values are four-attempt means (%), and Δ is the unweighted change across the four columns; values are rounded independently.

Configuration	AIME 2025	Apex 2025	Putnam 2025	IMO 2025	Δ
VERITASMATH (full)	100.00	83.33	92.71	82.14	0.00
w/o method-diverse fan-out (k=1)	98.33	77.08	89.58	79.76	−3.36
w/o evidence typing	96.67	77.08	91.67	80.95	−2.95
w/o conflict re-computation	97.50	75.00	88.54	79.76	−4.34
w/o selection-only constraint	98.33	77.08	90.63	80.36	−2.95
w/o obligation audit	99.17	81.25	85.42	76.19	−4.04
w/o repair re-entry	99.17	81.25	86.46	76.19	−3.78

4.2 SAME-CHECKPOINT SCAFFOLD RESULTS

Table 3 summarizes the same-checkpoint, latency-capped scaffold comparison on the controlled six-benchmark subset. On this subset, VERITASMATH achieves a problem-weighted mixed aggregate of 96.01%, compared with 92.42% for code-based self-verification, the highest aggregate among the listed baselines. The corresponding margin of 3.59 points is positive in each displayed benchmark column, ranging from 3.13 points on Putnam to 4.16 points on Apex 2025. These system-level results characterize deadline-bounded policy performance under the pre-specified evaluation protocol. All 396 planned slots per method are retained, with timed-out slots scored as zero within the fixed denominator. Appendix F provides the per-benchmark numerators and scoring conventions.

We report final-answer and proof strata separately so that normalized partial credit is interpreted on its own scale. Within the 75-problem final-answer stratum (AIME, HMMT, and Apex), VERITASMATH attains 97.33% accuracy versus 93.67% for code-based self-verification, a 3.67-point advantage. Within the 24-problem proof stratum, normalized proof credit is 91.89% versus 88.54%, a 3.35-point advantage. The paired contract study below complements these system-level results by isolating where bounded evidence and module authority support valid promotion and preserve initially correct candidates.

4.3 COMPONENT ABLATIONS

Table 4 reports the component ablations under the fixed fail-closed policy and shared latency ceiling. The corresponding ablations are associated with 4.34-, 4.04-, and 3.36-point changes in the four-column macro-average for conflict re-computation, obligation auditing, and method-diverse fan-out, respectively. The obligation-audit ablation has its largest displayed changes on proof-based tasks: 7.29 and 5.95 points on Putnam and IMO, compared with 0.83 points on AIME. These system-

Table 5: Contract-level reliability under fixed-trajectory policy replays (%), conditional on shared candidates and checker outputs.

Policy	FSR ↓	Promotion P ↑	C→I ↓	Mutation ↓	ADC ↑	CCR ↑
Full evidence–authority contract	**2.3**	**97.7**	**1.4**	**0.0**	**94.2**	**90.8**
Flat evidence labels	9.6	91.2	3.9	N/A	93.9	87.9
Unrestricted judge	18.4	82.8	8.7	6.4	91.3	81.1
Repair inherits parent status	6.8	94.5	4.5	N/A	93.7	88.6
Flat obligation audit	N/A	N/A	5.7	N/A	86.4	82.8
Unlocked rendering	N/A	N/A	6.7	11.3	N/A	86.7

level ablations vary realized compute, while the fixed-artifact replays below complement them by isolating the direct effect of individual policy decisions.

4.4 Contract-level reliability study

Table 5 reports contract-level reliability under fixed-trajectory policy replays. Each replay changes one decision contract while holding candidate generation, checking opportunities, and total generated tokens fixed, allowing the conditional effect of individual authority rules to be examined on shared upstream artifacts. The variants collapse evidence records into supporting_test/contradiction_report/inconclusive labels, permit judge-side answer synthesis, inherit the parent verdict for repaired candidates, model obligations without dependency structure, or allow the renderer to regenerate the final response. The full method and every variant process the same candidates and injected checker outputs.

Across these controlled replays, the full evidence–authority contract lowers FSR from 9.6% to 2.3% and raises promotion precision from 91.2% to 97.7% relative to flat evidence labels. The structured obligation graph yields an ADC value 7.8 points higher than a flat audit, while answer locking preserves every stored final payload; three detected mismatches were returned to the selected answer. In addition, all 160 targeted failure-injection cases across the eleven categories in Table 17 are routed according to their expected policy actions, and the full evidence records yield lower false-refutation and selective-risk rates. Appendix F provides counts and intervals, and Appendix D.1 reports process reach.

5 Limitations

Generated checks can encode wrong constraints, numerical matching can accept inequivalent expressions, and model roles may share errors. The contract limits their authority but does not provide formal soundness without a trusted system such as Lean (de Moura & Ullrich, 2021); obligation compilation can also overconstrain valid proof styles and trigger false repair.

The evaluation is descriptive: small, ceiling-prone subsets limit discrimination; the checkpoint cutoff is unknown; no time-clean holdout or memorization probe is available; compute is not matched; and the enriched cohort is unweighted. Results cover one fail-closed point; Appendix F records development, sampling, annotation, and interval conventions.

6 Conclusion

Reliable mathematical reasoning requires interpreting each verification signal according to the claim it checks, its domain, and its coverage. VERITASMATH operationalizes this principle through claim-level evidence contracts and an authority-constrained state machine that separate checking, conflict resolution, selection, repair, and rendering. Across competition benchmarks, the latency-capped comparison reports a higher mixed aggregate, while fixed-artifact replays show more precise candidate promotion, better preservation of correct candidates and final payloads, and higher deliverable completeness. These results support scoped evidence and bounded authority as useful design principles for reliable LLM reasoning, and provide a foundation for richer obligation models, calibrated output policies, and trusted formal verification.

AI USE STATEMENT

Generative AI tools assisted with prose and organization, the mathematical routing rules (the conceptual framework), the workflow diagram, LaTeX/code, and evaluation checks. The tools did not generate synthetic data or measurements. The authors supplied and checked the study configuration, reported results, and interpretations, reviewed all AI-assisted material, and remain responsible for the claims and artifacts. The studied system itself uses LLMs for reasoning, checking, selection, and auditing as described in the Method section.

REFERENCES

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2025. https://matharena.ai/.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

Konstantin Chernyshev, Vitaliy Polshkov, Vlad Stepanov, Alex Myasnikov, Ekaterina Artemova, Alexei Miasnikov, and Sergei Tilga. U-MATH: A university-level benchmark for evaluating mathematical skills in large language models. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pp. 974–1001, 2025.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *International Conference on Automated Deduction (CADE)*, pp. 625–635. Springer, 2021.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jasper Dekoninck, Nikola Jovanović, Ivo Petrov, and Martin Vechev. MathArena Apex: Unconquered final-answer problems, 2025a. https://matharena.ai/apex/.

Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Mislav Balunović, Martin Vechev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, and Kaloyan Tsvetkov. The open proof corpus: A large-scale study of LLM-generated mathematical proofs. *arXiv preprint arXiv:2506.21621*, 2025b.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning (ICML)*, 2024.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International Conference on Machine Learning (ICML)*, pp. 10764–10799, 2023.

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. *arXiv preprint arXiv:2305.11738*, 2024a.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *International Conference on Learning Representations (ICLR)*, 2024b.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3828–3850, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Jie Hu. Oracle gap and signal fidelity: A fixed-pool diagnostic for test-time collaboration. *arXiv preprint arXiv:2607.17531*, 2026.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. LLMs can't plan, but can help planning in LLM-modulo frameworks. *arXiv preprint arXiv:2402.01817*, 2024.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Trevor Darrell, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In *International Conference on Machine Learning (ICML)*, 2024.

LangChain Inc. LangGraph: Building stateful, multi-actor applications with LLMs, 2024. `https://github.com/langchain-ai/langgraph`.

Bin Lei, Yi Zhang, Shan Zuo, Ali Payani, and Caiwen Ding. MACM: Utilizing a multi-agent system for condition mining in solving complex mathematical problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison 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 (ICLR)*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. SymPy: Symbolic computing in Python. *PeerJ Computer Science*, 3:e103, 2017.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip H. S. Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. MALT: Improving reasoning with multi-agent LLM training. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2412.01928.

Shrey Pandit, Austin Xu, Xuan-Phi Nguyen, Yifei Ming, Caiming Xiong, and Shafiq Joty. Hard2Verify: A step-level verification benchmark for open-ended frontier math. *arXiv preprint arXiv:2510.13744*, 2025.

Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating LLMs on 2025 USA math olympiad. *arXiv preprint arXiv:2503.21934*, 2025.

Zhihong Shao, Yuxiang Luo, Chengda Lu, Z. Z. Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. DeepSeekMath-V2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*, 2025.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Ziyu Wan, Yunxiang Li, Xiaoyu Wen, Yan Song, Hanjing Wang, Linyi Yang, Mark Schmidt, Jun Wang, Weinan Zhang, Shuyue Hu, and Ying Wen. ReMA: Learning to meta-think for LLMs with multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2503.09501.

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 (ACL)*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 24824–24837, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Sen Xu, Wei Wang, Shixi Liu, Jixin Min, Yingwei Dai, Zhibin Yin, Yirong Chen, and Junlin Zhang. Claim-level reliability assessment for efficient test-time reasoning. *arXiv preprint arXiv:2608.11994*, 2026.

Xin Xu, Jiaxin Zhang, Tianhao Chen, Zitong Chao, Jishan Hu, and Can Yang. UGMathBench: A diverse and dynamic benchmark for undergraduate-level mathematical reasoning with large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 11809–11822, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*, 2024.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*, 2024.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*, 2025.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 36, 2023.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. Solving challenging math word problems using GPT-4 code interpreter with code-based self-verification. In *International Conference on Learning Representations (ICLR)*, 2024.

Rui Zou, Yutao Zhu, Mengqi Wei, and Ji-Rong Wen. AMTFV: Agentic mathematical tool-flow verification for LLM self-correction. *arXiv preprint arXiv:2607.29549*, 2026.