RAVEN: REASONING ACROSS PREMISE ORDERS WITH SOURCE-TO-RESPONSE CERTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Logical consequence is invariant to premise order, yet language models are not: reordering the same facts can change both the predicted answer and the symbolic program executed by a solver. Consequently, a locally valid proof may still certify a mistranslated input. We introduce RAVEN, a neuro-symbolic reasoner that turns premise-order sensitivity into testable evidence. It produces direct labels and formal results from source-equivalent premise orderings, validates each formalization, repairs syntax or type failures, executes each theory, and checks proof and source support. When the orderings disagree, a fixed decision rule uses their checked evidence to select a label; a separate reasoning-release certifier subsequently binds the selected formalization, proof, displayed reasoning, and answer to the source. With Qwen2.5-32B, RAVEN's accuracy-first controller reaches 88.76% macro-average accuracy across four benchmarks. Full reversal improves the fixed two-ordering study by 2.60 points over its same-order control. Under a common source-bound certification protocol on the same 600 ProofWriter test items, 46.33% (278/600) of RAVEN-Core outputs and 9.83% (59/600) of SymbCoT outputs are accepted, with no observed error in either accepted subset. Finally, on 19,790 audited finite-Horn items under registered translation evidence, the Core-attached certifier authorizes 10,901 responses with no observed accepted error and a 0.035% item-level 95% Wilson upper bound.

1 INTRODUCTION

Neuro-symbolic reasoning combines two useful capabilities: a language model interprets a natural-language problem, while a solver executes an explicit formal theory. The combination introduces two reliability boundaries. First, although logical consequence is invariant to premise order, the model's direct answer and its generated theory can change when the same premises are reordered. Prior work finds that proof-aligned orders can help reasoning (Chen et al., 2024), but the privileged proof order is unavailable when solving a new problem. A reasoner must therefore decide what to do with conflicting evidence from source-equivalent orderings.

Second, executing a generated theory does not establish that it represents the source or that the answer shown to a user follows the execution. On ProofWriter-full, checking local execution alone retains 10,055 of 10,190 responses but admits 137 wrong answers; a joint source-to-response check retains 3,677 with no observed accepted errors. Thus an executable proof and a releasable response are distinct objects.

RAVEN addresses these boundaries with two linked stages. Its premise-order reasoning stage obtains direct and executable evidence from the observed ordering and, when needed, its reversal. A fixed decision rule checks proof and source support before selecting an inference label. Its reasoning-release certifier then checks a frozen record connecting the source, formalization, solver witness, displayed derivation, and final answer; it authorizes or withholds that exact response without changing the selected label. Figure 1 shows how the two stages connect.

We ask how source-preserving premise reordering changes executable evidence, whether checked cross-order evidence improves inference over one ordering and matched-budget resampling, and how source-to-response certification trades coverage for accepted error across RAVEN and competing formal methods. Our contributions are:

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.

- **Premise-order reasoning consistency.** We turn reordering into a controlled intervention and combine direct reasoning with validated, source-supported solver evidence across orderings. The fixed two-ordering study improves the four-task macro-average by 2.60 points over its same-order control, with heterogeneous effects across tasks.
- **Reasoning-release certification.** We specify and evaluate a source-to-response checker over a frozen inference record. Its eight obligations separate executable validity from source and response binding; we measure both RAVEN's native release behavior and a common certificate applied to other neuro-symbolic methods.

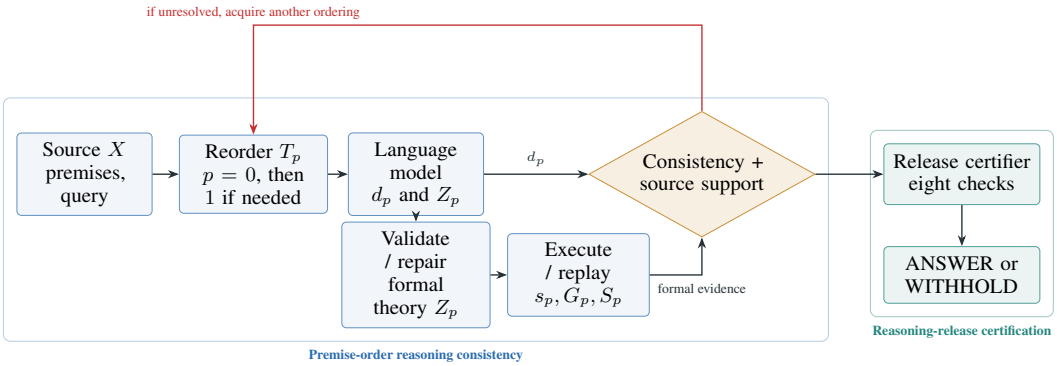


Figure 1: RAVEN framework. Each acquired premise ordering produces a direct label and a checked formal result. When evidence is insufficient, the inference controller requests another ordering; after selecting a label, a separate certifier checks the frozen source-to-response record and either releases or withholds the answer. The primary controller uses $p = 0$ and, when needed, $p = 1$; the selective conflict analysis may also acquire $p = .5$.

2 Related Work

Premise order and reasoning paths. Chen et al. (2024) find gains from proof-aligned orders, which are unknown at inference time. RAVEN uses disagreement among source-equivalent orderings as checked evidence. CoT and self-consistency generate or sample reasoning under a fixed rendering (Wei et al., 2022; Wang et al., 2023); Tree of Thoughts, Selection-Inference, LAMBADA, LogicTree, and Aristotle organize search over reasoning paths or premises (Yao et al., 2023; Creswell et al., 2023; Kazemi et al., 2023; He & Roy, 2025; Xu et al., 2025). COP, DetermLR, and HBLR further structure logical inference (Liu et al., 2025; Sun et al., 2024; Li et al., 2026). Proof-chain supervision and training target related but distinct behavior (Han et al., 2024b; Wen et al., 2025; Tan et al., 2025; Xu et al., 2026a).

Translation and executable reasoning. PAL, Logic-LM, and LINC execute model-generated programs (Gao et al., 2023; Pan et al., 2023; Olausson et al., 2023). SymbCoT, CLOVER, FoVer, NL2Logic, and PEIRCE develop symbolic translation, verification, or refinement (Xu et al., 2024; Ryu et al., 2025; Pei et al., 2025; Putra et al., 2026; Quan et al., 2025). RAVEN connects an executable result to its source and selected visible response across premise orders.

Certification and selective release. Proof-carrying code, translation validation, and runtime verification separate evidence from its checker (Necula, 1997; Pnueli et al., 1998; Leucker & Schallhart, 2009). Selective prediction studies coverage and risk (Angelopoulos et al., 2025; 2024). LoCo studies model-level logical consistency; NSF-CoT and SATQuest verify different reasoning artifacts (Calanzone et al., 2025; Pramanik et al., 2026; Zhao et al., 2026). RAVEN checks the particular source-to-answer response and measures source-bound release coverage.

3 RAVEN METHOD

RAVEN first selects an inference label from direct and executable evidence over source-equivalent premise orderings. It then checks whether the exact response built from that selection may be released. Figure 1 shows the conditional second-ordering loop and the separate release decision. Answer-only and CoT are comparison methods, not inputs to this loop.

3.1 OVERVIEW

Let $X = (P, q)$ be a natural-language reasoning problem, with premises $P = ((i_1, p_1), \ldots, (i_N, p_N))$ and query q. Each i_j is a stable source ID, not a new fact; it travels with p_j through reordering and identifies the source of a generated formula or proof step. A reordering gives $X^{(p)} = (T_p(P), q)$. The registered task profile π separately fixes the answer labels, formal syntax, logic, and solver; the model cannot select it.

On acquired ordering p, direct reasoning yields d_p. The formal branch yields a theory Z_p, and execution returns solver label s_p with machine-checkable witness G_p. For a definite result, S_p denotes its replayed source-premise support. The controller selects inference label $\hat{y}$ and records the decision in η. The selected witness is rendered as a user-visible derivation D whose terminal answer is Y; D is distinct from the machine witness G. Section 4 checks their binding.

Three outcomes remain distinct. U is a semantic task label under a complete theory; $\perp$ marks unavailable evidence; WITHHOLD is a later release action on a selected response. Profile details appear in Appendix E.

3.2 p-REORDERING OF PREMISES

The proof-aligned order studied by Chen et al. (2024) is informative but ordinarily unknown at inference time. RAVEN therefore probes the same source using controlled reorderings, without consulting a gold proof or label. The operator T_p swaps each mirrored pair of premise-ID positions with probability p, using an item-derived fixed seed so paired evaluations are nested and reproducible. Thus $X^{(0)}$ is the observed ordering, $X^{(1)}$ its exact reversal, and $X^{0.5)}$ a midpoint reordering. The primary controller requests the reversal only when needed; the stricter selective analysis requests the midpoint only after a checked endpoint conflict. The five-value sweep measures sensitivity rather than selecting a new ordering per item. Appendix G gives the exact permutation, hash rule, and distance identities.

3.3 REASONING CONSISTENCY CHECK

Direct reasoning. For an acquired ordering $X^{(p)}$, a direct model call returns a task label d_p. The direct label is evidence about the model's interpretation of that ordering; it is not a solver certificate. The main controller compares it with the independently executed formal result from the same source.

Neuro-symbolic reasoning. For that ordering, the model also produces a symbolic formulation Z_p mapping every premise ID and the query ID to a formula. RAVEN checks this output before calling a solver. A schema check verifies the required fields. An ID check verifies that no source sentence or query has been dropped or duplicated. A parser checks the formula syntax and predicate signatures. A task check rejects operators outside the supported logic. Execution begins only if all four checks pass.

A failed formulation may trigger one repair call. The model receives its invalid formulation and the concrete validation error, but not the reference answer. It must return a complete replacement, which passes through the same checks. A second failure makes that ordering's formal evidence unavailable ($\perp$). This is bounded self-repair from a machine-detected error, not repeated answer sampling.

These checks establish structural validity and executability. The later Translation obligation checks source equivalence under a registered evidence mode; parser success alone does not establish it.

Only parsed formulas reach the solver. It returns a label $s_p \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ and witness G_p: a proof, countermodel, or complete fixed-point result under the task profile. RAVEN records formulas, solver status, proof steps, and source IDs rather than treating an unchecked model string as evidence.

For a definite result, RAVEN re-executes the solver on support S_p. The retained premises must reproduce s_p, and deleting any retained premise must change it. This is subset-minimal support, not a globally smallest proof. A separate source-language audit asks whether the retained premises suffice and, when support is nonempty, whether a selected deletion removes the answer. These checks determine which formal evidence the controller may prioritize; the exact replay test appears in Appendix G.

Consistency resolution. The accuracy-first controller first acquires the direct and formal results from the observed ordering. If $d_0 = s_0$ is a definite label with source-supported core evidence, it selects that label. A definite s_0 with source-supported core evidence may also be selected when the direct channel disagrees. Otherwise, the controller acquires the reversed ordering and compares the two direct and formal channels.

After reversal, checked cross-channel agreement and source-supported formal evidence take precedence. Conflicting machine-certified formal results may be resolved by stable direct agreement only when the applicable replay, source-support, and veto checks permit it. If no checked core decides, the controller falls back through the registered non-vetoed channel consensuses and original-order evidence; if no branch applies, inference remains unresolved. The controller returns an inference label $\hat{y}$ or unresolved and records the selected branch in η. Appendix G.1 specifies the exact ordered tie breaks, vetoes, and early exits.

One terminal rule completes the main controller. On a three-label task, if the hierarchy would retain a stable direct label but the independently executed formal results from the observed and reversed orderings both return U, RAVEN selects U. This rule consumes no additional call and never reads a reference label. It is inapplicable to binary tasks; every other decision remains unchanged. The final controller reported as RAVEN in the main comparison was selected on development evidence; the Core hierarchy and the final rule are reported separately in Section 5.2.

We study a stricter inference threshold separately to quantify the cost and benefit of verification. A definite formal label is admitted only when its formulation is valid, the solver returns that label, machine replay verifies subset-minimal support, and the source-support check accepts the retained premises. A semantic U is admitted only after a valid formulation completes under a finite, complete solver profile. Invalid, unexecutable, or source-rejected results cannot vote.

Let $a_p \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}, \perp\}$ denote this selectively admissible result from $T_p(P)$, where $\perp$ means that the ordering supplies no admissible label. The selective rule returns a shared endpoint label, or the one admissible endpoint label if the other is unavailable. It requests $a_{.5}$ only when two admissible endpoints conflict, then returns an in-task label only if it receives two checked votes; otherwise it remains unresolved. Appendix G, Table 28 gives the full rule.

Answer-only and CoT predictions are produced by separate baseline protocols and never enter either controller. The accuracy-first controller supplies the main benchmark result. The selective controller supplies the coverage-reliability analysis. In both cases, unresolved inference remains distinct from semantic U, and WITHHOLD remains a later response-release action.

The selected label, executable witness, support audit, and decision record become the exact frozen input to response certification; the release stage does not reconstruct an independent explanation.

4 REASONING-RELEASE CERTIFIER

Input and output. Inference selects $\hat{y}$; it does not by itself authorize a public answer. The certifier receives a frozen candidate response bundle

$$b = (A, \hat{y}, \eta, \pi, W), \qquad A = \{X, Z^\star, G^\star, S^\star, D, Y\}, \tag{1}$$

where starred objects are the selected formal theory, executed witness, and source support, η is the recorded inference decision, π is the registered task/solver profile, and W contains admitted assumptions. The implementation binds typed Source, Formalization, Derivation, RenderedTrace,

TerminalLabel, ScopeWitness, Profile, and ReleasePolicy artifacts by canonical digests. Equation 1 is a conceptual projection of these typed artifacts: $G^\star$ is machine-checkable proof or closure evidence, whereas D is the deterministic visible rendering of that evidence, not a second solver proof. The checker makes no model call and never edits $\hat{y}$. It returns a release decision r and an audit record σ containing the status of every registered check. The input bundle is not itself a certificate.

The release checks follow the source-to-response chain

$$X \longrightarrow Z^\star \longrightarrow G^\star \longrightarrow D \longrightarrow Y, \tag{2}$$

Translation checks $X \to Z^\star$; Profile and LocalStep replay $Z^\star \to G^\star$; Display and Terminal check $G^\star \to D \to Y$; Artifact, Global, and Scope bind all entries to the same source, inference decision, and admitted assumptions. These checks distinguish mistranslation, an invalid proof step, a missing displayed dependency, a mismatched final answer, and an out-of-scope premise. The release decision is

$$J(b) = \bigwedge_{k \in \mathcal{K}} A_k(b), \tag{3}$$

$$\mathcal{K} = \{art, prof, trans, step, disp, term, global, scope\},$$

$$r(b) = \textsc{Answer} \iff J(b) = 1. \tag{4}$$

Release procedure. Given a frozen candidate bundle b, the certifier evaluates the eight registered interfaces, Artifact, Profile, Translation, LocalStep, Display, Terminal, Global, and Scope, under the registered translation-evidence mode. Their predicates are evaluated independently, and the audit record σ records the status of every interface rather than stopping at the first failure. As defined in Equation 4, the response is released only when all eight predicates pass; otherwise it is withheld. The exact mapping from these interfaces to runtime predicates appears in Table 25.

Each interface blocks a distinct mismatch in the frozen response. **Artifact** compares canonical digests and upstream dependencies so that a proof or answer from another source cannot be substituted. **Profile** fixes the registered logic fragment, parser, and solver; a proof from another calculus is not interchangeable evidence. **Translation** compares every source premise and query with the normalized theory under controlled-compiler or registered-oracle evidence, rejecting a locally valid proof over a mistranslated input. **LocalStep** re-executes the theory and replays its witness. A definite finite-Horn label needs a proof; semantic U needs complete closure showing neither query polarity, not a timeout.

Display matches the visible steps and parent edges to the machine witness, blocking invented or omitted reasoning. **Terminal** binds witness end, displayed conclusion, selected label, and public answer so that a substituted answer is withheld. **Global decision consistency** checks that source support and acquired order records justify the selected inference branch, rather than merely some valid proof. **Scope** binds source IDs, versions, and admitted assumptions, excluding hidden premises and cross-item evidence. These checks act after inference: they can withhold an unbound response but cannot silently change its label.

For example, in a scored ProofWriter item, p_2 states that Bob is round; $p_{10}, p_9, p_{12}, p_8, p_{13}$ carry a six-step proof to "Bob is green." The checker verifies the source-aligned theory, replayed proof, six displayed steps, and terminal answer T, then authorizes that response. If only the visible terminal is changed to F, the proof still executes, but Terminal fails and the response is withheld. Appendix E.2 gives the per-interface procedures and paired item-level checks. In the registered exact-one mutation study, removing any one interface admits all 200 faults targeted at that interface, while the full product rejects all 1,600 negatives (Appendix D, Table 19). This establishes independence for those fault families, not a universal minimality claim.

Conditional source-to-response guarantee. Under the registered trust assumptions that translation preserves source semantics, witness replay is sound for the registered solver, rendering preserves the selected witness dependencies, and the remaining artifact, profile, decision, and scope bindings are sound, passing all eight interfaces implies that the selected label follows under the registered task semantics and that the visible terminal answer agrees with it.

The argument follows the source-to-response chain in Equation 2: Translation transfers the source problem to the checked theory, sound replay licenses the selected witness, and Display and Terminal carry its selected conclusion into the visible answer. Artifact, Profile, Global, and Scope bind

these objects to one registered response. Appendix H gives the corresponding justification and trust assumptions. This guarantee is conditional on the controlled compiler or registered oracle and does not establish unrestricted English-to-logic equivalence.

5 EXPERIMENT

5.1 EXPERIMENTAL SETUP

Table 1 maps each inference protocol to its evaluation; response checking is a separate release layer.

Table 1: Protocol map. The 88.76% benchmark result uses Final, whereas the native response checker evaluates Core artifacts. Unresolved is an inference outcome; release is a later decision.

Protocol	Orderings/evidence	Decision rule	Reported in
SingleView	observed	one-ordering direct/formal rule	Table 2
Core	original, then reverse if needed	accuracy-first hierarchy	Table 2
RAVEN Final	Core orderings	hierarchy plus terminal two-formal-U rule	Table 2
Fixed two-ordering	observed and T_p	common accuracy-first parent; vary p	Table 6; Appendix B
Selective inference	checked endpoints; midpoint on conflict	checked formal label or unresolved	Table 15
Native release	frozen Core responses	eight-interface source-to-answer check	Section 5.3.1

We evaluate on FOLIO-v2 validation ($n = 203$), PrOntoQA development ($n = 500$), ProofWriter test ($n = 600$), and ProverQA-hard ($n = 500$) (Han et al., 2024a; Saparov & He, 2023; Tafjord et al., 2021; Qi et al., 2025). Methods share item IDs, label mappings, and checkpoint revisions; unresolved outputs count as wrong. The primary backbone is Qwen2.5-32B-Instruct, with Qwen3-8B and Llama-3.3-70B-Instruct replications. CoT uses one sample and CoT-SC uses 16 at temperature 0.7. Unsupported combinations remain absent. Tables abbreviate PrOntoQA as PQA and ProofWriter as PW.

The separate fixed two-ordering protocol holds the checkpoint, exact IDs, seed, prompts, call ceilings, and decision rule constant. Its first ordering uses the observed order at temperature 0; its second uses T_p at temperature 0.7 for $p \in \{0, 0.25, 0.5, 0.75, 1\}$. The sweep and the staged diagnostic controllers in Table 2 are evaluated as distinct protocols. Formal resampling matches the $p = 1$ parent's call count and token caps; component removals retain the same full denominators. The development selection criterion for the fixed $p = 1$ parent is given in Appendix Table 14.

We report full-denominator accuracy. Paired differences use exact McNemar tests and unadjusted paired-bootstrap 95% intervals; Holm correction applies only to the McNemar p-values within each declared family. Assurance results report coverage, accepted accuracy, accepted count, and a 95% upper confidence bound. Mutation banks change one typed interface at a time and include semantics-preserving controls.

For cross-method certification, Level E tests executable evidence using Artifact, Profile, LocalStep, and Terminal. Level C adds Translation and Scope under the same registered benchmark contract; it is the common source-to-answer comparison. Native RAVEN release adds Display and Global bindings (Level R). Methods without a formalization, such as CoT, have no native Level-E/C artifact and are not assigned zero certification coverage. LINC's ten candidates are certified separately; its predeclared aggregation requires at least six certified candidates for its released label and none for a conflicting label.

Wilson intervals describe accepted error in the audited release pool. Dataset identity is checked by source keys and content hashes before pooling.

5.2 EXPERIMENTAL RESULTS

5.2.1 PRIMARY QWEN2.5-32B COMPARISON

Core gains 27.26 macro points over answer-only, 14.22 over CoT, and 13.78 over CoT-SC$_{16}$. Relative to SINGLEVIEW, it gains 1.50 points on ProofWriter and 3.80 on ProverQA, ties on FOLIO, and

Table 2: Qwen2.5-32B full-denominator accuracy (%). The table compares baselines with staged RAVEN controllers. Avg uses all four tasks; unsupported cells are not averaged. The RAVEN row carries Core forward when its three-label terminal rule is inapplicable.

Method	FOLIO	PQA	PW	ProverQA	Avg
Answer only	69.46	73.20	49.00	49.20	60.21
CoT	73.89	98.80	66.33	54.00	73.26
CoT-SC$_{16}$	79.80	99.20	63.17	52.60	73.69
SymbCoT	76.85	98.80	79.00	N/A	N/A
LINC$_{10}$	70.94	N/A	98.50	N/A	N/A
DetermLR	73.89	82.60	51.50	N/A	N/A
RAVEN-SINGLEVIEW	81.77	**100.00**	96.83	66.00	86.15
RAVEN-CORE	81.77	**100.00**	98.33	69.80	87.48
RAVEN	**84.73**	**100.00**	**98.50**	**71.80**	**88.76**

reaches the PQA ceiling. Across five paired 32B endpoints, only the PQA-matrix contrast passes Holm correction (Appendix Table 4).

The final zero-call rule raises the mean by 1.28 points over Core. It adds 0.54 points on ProofWriter-full ($p = 6.0 \times 10^{-5}$) and 2.0 on ProverQA-hard (unadjusted $p = .0664$). Its benefit is conditional: replacing a definite label by U improves risk only when gold U is more likely on the trigger event than the definite Core label.

The fixed-ordering experiment below isolates premise order from the staged controller; Section 5.3 separately measures the coverage and accepted error of selective inference.

5.2.2 PREMISE-ORDER AND COMPONENT MECHANISMS

The full sweep in Appendix B, Table 6, documents a single-seed, endpoint-dependent order effect. Full reversal improves the macro-average by 2.60 points over $p = 0$ and is best on ProofWriter and ProverQA; FOLIO peaks at $p = 0.25$, while PQA is saturated. Thus p controls a representation intervention rather than a universal difficulty scale.

Formal-SC matches call counts and token ceilings but averages 14.37 points below the fixed two-ordering parent and leaves 287 items unresolved. Removing source support costs 8.38 macro points, while removing bounded repair costs 1.74; these effects are not additive or realized-compute matched. Appendix B reports the component rows and paired repair transitions in Tables 7 and 13.

Model-scale replication. Core exceeds answer-only by 21.36 macro points on Qwen3-8B and 25.84 on Llama-3.3-70B; seven of eight 70B comparisons with CoT and SC$_{16}$ survive joint Holm correction. Full results appear in Appendix B.

Figure 2 shows two scored selective-policy cases in which source-equivalent orderings produce different executable results.

A. Observed order retains the useful translation (FOLIO).
Source: p_1: Heinrich Schmidt was a German politician; p_2: he was a member of the Prussian State Parliament and the Nazi Reichstag. **Query:** German or Russian or both?
Observed $p = 0$**:** $p_1 \mapsto kind(h, german) \wedge politician(h)$; $q \mapsto kind(h, german) \vee kind(h, russian)$; solver T, support $\{p_1\}$.
Reversed $p = 1$**:** $p_1 \mapsto politician_h_german$; $q \mapsto german(h) \vee russian(h)$; solver U, no proof.
Resolution: endpoint conflict T/U; midpoint T with $\{p_1\}$; selective result T.

B. Reversal restores a missed fact (ProofWriter).
Source: p_1: Anne is quiet; p_{14}: quiet ⇒ rough; p_{11}: rough ⇒ green; p_{16}: green ⇒ kind. **Query:** Anne is kind.
Observed $p = 0$**:** $p_1 \mapsto \neg nice(Anne)$; solver U, no proof.
Reversed $p = 1$**:** $p_1 \mapsto quiet(Anne)$; solver T, support $\{p_1, p_{11}, p_{14}, p_{16}\}$.
Resolution: endpoint conflict U/T; midpoint T with the same support; selective result T.

Figure 2: Two scored order-sensitive cases under the selective checked-evidence rule. Source-equivalent premise orders yield different executable labels; the midpoint resolves each conflict. Shown premises are proof-relevant excerpts, not the full input; entity identifiers are abbreviated in A.

5.3 SELECTIVE CHECKED INFERENCE: COVERAGE AND RELIABILITY

Verification strength changes what the system is willing to return. The accuracy-first $p = 1$ controller answers every item and attains 1,614/1,803 correct predictions. Requiring checked formal evidence

reduces coverage. Checked definite labels return 618/1,803 answers, of which 616 are correct (99.68% accepted accuracy at 34.28% coverage). Admitting semantic U raises coverage to 67.33% but admits 195 errors in 1,214 returned labels. Table 15 in Appendix D gives the full threshold comparison.

The definite threshold has a 1.172% Wilson 95% upper bound on accepted error. The all-label row has lower accepted accuracy because executable U is reliable only when the generated theory is also semantically complete for the source. Across the checked endpoints, 19 admissible endpoint pairs conflict; the midpoint supplies a second checked vote in 16 cases and three remain unresolved. This analysis treats strict verification as a selective decision threshold, while the accuracy-first controller remains the primary benchmark policy.

5.3.1 REASONING-RELEASE CERTIFICATION

Why local execution is insufficient. On ProofWriter-full, LocalStep-only checking returns 10,055/10,190 responses (98.68% coverage) but accepts 137 wrong answers. The joint source-bound check returns 3,677 (36.08% coverage) with zero observed accepted errors. This comparison motivates source binding in addition to solver replay; it is not the 600-item cross-method comparison below.

Common checks across methods. Table 3 applies the common Level-E and Level-C criteria to methods that emit formal artifacts on identical item IDs. It reports coverage, not a new inference accuracy score. Level E requires executable evidence; Level C additionally checks source translation and scope. All five Level-C accepted subsets have zero observed errors; Appendix D.1, specifically Tables 20 and 21, reports accepted-error bounds and correct-withholding rates.

Table 3: Common certification on identical items. E checks executable evidence; C additionally binds source and scope. Parentheses give the C accepted count; all five C-accepted subsets have zero observed errors.

Endpoint	Method	n	Base acc.	E cov.	C cov. (accepted)
PW test	RAVEN-Core	600	98.33	99.17	46.33 (278)
PW test	LINC_{10}	600	98.50	64.50	42.67 (256)
PW test	SymbCoT	600	79.00	59.67	9.83 (59)
PQA dev	RAVEN-Core	500	100.00	100.00	91.20 (456)
PQA dev	SymbCoT	500	98.80	93.40	90.40 (452)

On PW, RAVEN's Level-C coverage exceeds SymbCoT by 36.50 points (Holm-adjusted McNemar $p = 1.22 \times 10^{-43}$). Its 3.67-point difference from LINC on PW and 0.80-point difference from SymbCoT on PQA are not statistically distinguishable under the registered paired tests. The native Level-R checker has a different, stronger interface set and is not used for those head-to-head claims. On PW, RAVEN-Core's first Level-C failures are Translation (316) and Scope (6), whereas SymbCoT first fails Translation (363), Profile (88), Scope (78), or LocalStep (12). LINC_{10} lacks a qualifying aggregate in 344 cases. These counts locate the coverage gap rather than implying that every failure is an incorrect answer.

Native response release. Native release evaluates frozen CORE responses on a content-disjoint 19,790-item finite-Horn pool with controlled-compiler or registered-oracle translation evidence. It authorizes 10,901 responses (55.08%) with no observed accepted errors. The item-level Wilson 95% upper endpoint is 0.035%, and the source-grouped endpoint is 0.051% over 7,484 accepted contexts. All 248 wrong Core labels are withheld: Translation first fails 168 and LocalStep 80. Translation also first withholds 7,536 correct Core predictions, making it the principal coverage bottleneck under this benchmark contract. Endpoint results and controlled mutations appear in Appendix D.

6 DISCUSSION

Why premise-order comparisons add information. Logical consequence is order-invariant, but generated theories need not be. Proof-aligned order can help (Chen et al., 2024), yet is unknown for a new problem. Source-equivalent reorderings reveal missing formulas, recover support, or expose

conflict (Figure 2). The direct $p = 1$ versus $p = 0$ macro contrast is +2.60 points, but FOLIO peaks at $p = 0.25$ and PQA is saturated. The 14.37-point gap to formal resampling is a broader matched-call/token controller contrast, not a pure order effect.

What checked inference changes. Validation, repair, proof replay, and source-support checks govern which formal evidence the accuracy-first controller can prioritize. Removing source support costs 8.38 macro points, with branch-dependent effects. Selective inference has a different trade-off: 616/618 definite labels are correct at 34.28% coverage, but semantic U accounts for 193/195 checked-all errors because it depends on translation completeness.

Release certification asks whether the selected source, theory, witness, visible reasoning, and answer form one authorized response. LocalStep-only checking retains 137 wrong outputs. Common Level-C checks expose a source-binding gap across formal-output methods, while native Translation withholds many correct responses; better source-faithful formalization is therefore central to higher certified coverage.

Limitations The source-to-response guarantee assumes registered controlled-compiler or typed-oracle evidence, sound solver replay, and complete dependency extraction; it does not establish unrestricted-English equivalence. Native release covers finite-Horn profiles, and common Level C is weaker than full Level R. Semantic U requires translation completeness. Supports are subset-minimal, not minimum-cardinality; mirrored reorderings omit arbitrary permutations, the sweep uses one seed, and matched call/token ceilings do not equalize realized FLOPs. Controlled mutations test registered interfaces, not natural fault prevalence; zero observed errors do not imply zero risk.

7 Conclusion

RAVEN turns premise-order sensitivity into checked evidence for neuro-symbolic inference. With Qwen2.5-32B, its Final controller reaches 88.76% macro accuracy across four tasks, and reversal yields a task-dependent +2.60-point gain over a same-order control. Separately, the Core-attached certifier authorizes 10,901/19,790 audited finite-Horn responses under registered translation evidence with no observed accepted errors. Common certification compares source-bound formal outputs across methods. Together, these results separate label selection from response authorization and, under the registered benchmark contract, identify source-faithful translation as the dominant bottleneck to higher certified coverage.

AI Use Statement

Generative AI tools were used to assist with drafting, language editing, and polishing the manuscript. The authors independently checked all experimental results, mathematical statements, citations, claims, and final text and take full responsibility for the content.

Reproducibility Statement

The paper and appendices specify the datasets and evaluation splits, model and tokenizer revisions, inference protocols, decoding settings, call and token budgets, statistical tests, and certification criteria used in our experiments. Appendix E records the benchmark and trust-root contracts, while Appendix G.1 specifies the deterministic ordering and decision procedures, replay checks, and release criteria. Additional paired results, resource measurements, assurance analyses, and proof details are reported in the appendices. Generation does not access reference labels, and all reported comparisons use the stated full denominators without imputing unsupported endpoints.

References

Anastasios N. Angelopoulos, Stephen Bates, Adam Fisch, Lihua Lei, and Tal Schuster. Conformal risk control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=33XGfHLtZg.

Anastasios N. Angelopoulos, Stephen Bates, Emmanuel J. Candès, Michael I. Jordan, and Lihua Lei. Learn then test: Calibrating predictive algorithms to achieve risk control. *The Annals of Applied Statistics*, 19(2):1641 – 1662, 2025. doi: 10.1214/24-AOAS1998. URL https://doi.org/10.1214/24-AOAS1998.

Diego Calanzone, Stefano Teso, and Antonio Vergari. Logically consistent language models via neuro-symbolic integration. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/871a06b60cf087bbdb854ebfcdf5349a-Abstract-Conference.html.

Xinyun Chen, Ryan Andrew Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6596–6620. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24i.html.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=3Pf3Wg6o-A4.

Debargha Ganguly, Vikash Singh, Sreehari Sankar, Biyao Zhang, Xuecen Zhang, Srinivasan Iyengar, Xiaotian Han, Amit Sharma, Shivkumar Kalyanaraman, and Vipin Chaudhary. Grammars of formal uncertainty: When to trust LLMs in automated reasoning tasks. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=QfKpJ00t2L.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. 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. 10764–10799. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. FOLIO: Natural language reasoning with first-order logic. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1229. URL https://aclanthology.org/2024.emnlp-main.1229/.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, Martin Riddell, Wenfei Zhou, Yujie Qiao, Yilun Zhao, Semih Yavuz, Ye Liu, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Dragomir Radev, Rex Ying, and Arman Cohan. P-FOLIO: Evaluating and improving logical reasoning with abundant human-written reasoning chains. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 16553–16565, 2024b. doi: 10.18653/v1/2024.findings-emnlp.966. URL https://aclanthology.org/2024.findings-emnlp.966/.

Kang He and Kaushik Roy. LogicTree: Structured proof exploration for coherent and rigorous logical reasoning with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20852–20881, 2025. doi: 10.18653/v1/2025.emnlp-main.1054. URL https://aclanthology.org/2025.emnlp-main.1054/.

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. LAMBADA: Backward chaining for automated reasoning in natural language. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6547–6568, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.361. URL https://aclanthology.org/2023.acl-long.361/.

Martin Leucker and Christian Schallhart. A brief account of runtime verification. *Journal of Logic and Algebraic Programming*, 78(5):293–303, 2009. doi: 10.1016/j.jlap.2008.08.004.

Qingchuan Li, Mingyue Cheng, Zirui Liu, Daoyu Wang, Yuting Zeng, and Tongxuan Liu. From hypothesis to premises: LLM-based backward logical reasoning with selective symbolic translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31671–31679, 2026. doi: 10.1609/aaai.v40i37.40434. URL https://ojs.aaai.org/index.php/AAAI/article/view/40434.

Junjie Liu, Shaotian Yan, Chen Shen, Zhengdong Xiao, Liang Xie, Wenxiao Wang, and Jieping Ye. Concise and organized perception facilitates reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3469–3498, 2025. doi: 10.18653/v1/2025.findings-naacl.193. URL https://aclanthology.org/2025.findings-naacl.193/.

George C. Necula. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 106–119, 1997. doi: 10.1145/263699.263712.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5153–5176, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.313. URL https://aclanthology.org/2023.emnlp-main.313/.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3806–3824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248/.

Yu Pei, Yongping Du, and Xingnan Jin. FoVer: First-order logic verification for natural language reasoning. *Transactions of the Association for Computational Linguistics*, 13:1340–1359, 2025. doi: 10.1162/tacl.a.41. URL https://aclanthology.org/2025.tacl-1.61/.

A. Pnueli, M. Siegel, and E. Singerman. Translation validation. In Bernhard Steffen (ed.), *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 151–166, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. ISBN 978-3-540-69753-4.

Vishal Pramanik, Maisha Maliha, Nathaniel D. Bastian, Alvaro Velasquez, Susmit Jha, and Sumit Kumar Jha. NSF-CoT: Neuro-symbolic formal verification of chain-of-thought faithfulness in contextual question answering. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10645–10663, 2026. doi: 10.18653/v1/2026.findings-acl.516. URL https://aclanthology.org/2026.findings-acl.516/.

Rizky Ramadhana Putra, Raihan Sultan Pasha Basuki, Yutong Cheng, and Peng Gao. NL2Logic: AST-guided translation of natural language into first-order logic with large language models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6035–6051, 2026. doi: 10.18653/v1/2026.findings-eacl.317. URL https://aclanthology.org/2026.findings-eacl.317/.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. Large language models meet symbolic provers for logical reasoning evaluation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=C25SgeXWjE.

Xin Quan, Marco Valentino, Danilo Carvalho, Dhairya Dalal, and Andre Freitas. PEIRCE: Unifying material and formal reasoning via LLM-driven neuro-symbolic refinement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 11–21, 2025. doi: 10.18653/v1/2025.acl-demo.2. URL https://aclanthology.org/2025.acl-demo.2/.

Hyun Ryu, Gyeongman Kim, Hyemin S. Lee, and Eunho Yang. Divide and translate: Compositional first-order logic translation and verification for complex logical reasoning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3e592c571de69a43d7a870ea89c7e33a-Abstract-Conference.html.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations*, 2023.

Hongda Sun, Weikai Xu, Wei Liu, Jian Luan, Bin Wang, Shuo Shang, Ji-Rong Wen, and Rui Yan. DetermLR: Augmenting LLM-based logical reasoning from indeterminacy to determinacy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9828–9862, 2024. doi: 10.18653/v1/2024.acl-long.531. URL https://aclanthology.org/2024.acl-long.531/.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

Xingwei Tan, Marco Valentino, Mahmud Elahi Akhter, Maria Liakata, and Nikolaos Aletras. Enhancing logical reasoning in language models via symbolically-guided Monte Carlo process supervision. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31886–31900, 2025. doi: 10.18653/v1/2025.emnlp-main.1624. URL https://aclanthology.org/2025.emnlp-main.1624/.

Jinlong Tian, Jiang Yu, Kewei Cheng, Fengxiang Cheng, Yue He, Yunfei Wang, Haotian Wang, Haoxuan Li, Wenjing Yang, and Shixuan Liu. LogicSAGE: Neuro-symbolic reasoning with socratic-guided enhancement. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=GxSB5ZYfM4.

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*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Chengyao Wen, Qiang Cheng, Shaofei Wang, Zhizhen Liu, Deng Zhao, and Lei Liang. Logic-thinker: Teaching large language models to think more logically. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12955–12969, 2025. doi: 10.18653/v1/2025.findings-emnlp.696. URL https://aclanthology.org/2025.findings-emnlp.696/.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. Faithful logical reasoning via symbolic chain-of-thought. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13326–13365, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.720. URL https://aclanthology.org/2024.acl-long.720/.

Jundong Xu, Hao Fei, Meng Luo, Qian Liu, Liangming Pan, William Yang Wang, Preslav Nakov, Mong-Li Lee, and Wynne Hsu. Aristotle: Mastering logical reasoning with a logic-complete decompose-search-resolve framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3052–3075, 2025. doi: 10.18653/v1/2025.acl-long.153. URL https://aclanthology.org/2025.acl-long.153/.

Jundong Xu, Hao Fei, Huichi Zhou, Xin Quan, Qijun Huang, Shengqiong Wu, William Yang Wang, Mong-Li Lee, and Wynne Hsu. Logicreward: Incentivizing LLM reasoning via step-wise logical supervision. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=IRhYVOKFe0.

Lei Xu, Pierre Beckmann, Marco Valentino, and Andre Freitas. Adaptive LLM-symbolic reasoning via dynamic logical solver composition. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1187–1208, Rabat, Morocco, March 2026b. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.54. URL https://aclanthology.org/2026.eacl-long.54/.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Yaoning Yu, Ye Yu, Peiyan Zhang, Kai Wei, Haojing Luo, and Haohan Wang. SIPDO: Closed-loop prompt optimization via synthetic data feedback. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kpT5rbbLdY.

Yanxiao Zhao, Yaqian Li, Zihao Bo, Rinyoichi Takezoe, Haojia Hui, Mo Guang, Lei Ren, Xiaolin Qin, and Kaiwen Long. SATQuest: A verifier for logical reasoning evaluation and reinforcement fine-tuning of LLMs. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2109–2131, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.96. URL https://aclanthology.org/2026.acl-long.96/.

The appendix separates supporting evidence by purpose. Appendix A reports paired tests, Appendix B gives full model-scale results, and Appendix C presents trace-verified recovery cases. Appendix D analyzes assurance and its typed ablations, while Appendix E records benchmark and trust-root contracts. Finally, Appendix F positions related methods, Appendix G gives operational details, and Appendix H contains the proofs.

A Additional paired evidence

The aggregate gains in the main text are complemented by paired, identical-ID tests. Table 4 compares Core with SingleView at 32B, while Table 5 compares Core with CoT and SC_{16} at 70B. These tests isolate within-item changes that macro-accuracy alone cannot show.

Table 4: Paired Qwen2.5-32B Core versus SingleView effects on identical IDs. Intervals are unadjusted paired-bootstrap 95% CIs; Holm correction applies only to the exact McNemar p-values across the five registered endpoints.

Endpoint	n	Δ (pp)	95% CI	Holm p
PW test	600	+1.500	[0.327, 2.792]	.0957
PW full	10,190	+0.775	[0.156, 1.430]	.0744
PQA matrix	9,000	+0.156	[0.078, 0.244]	.0026
FOLIO-v2	203	0.000	[−5.825, 6.061]	1.000
ProverQA-hard	500	+3.800	[0.600, 7.000]	.0957

Table 5: Paired Llama-3.3-70B effects. Exact McNemar tests and paired bootstrap CIs use identical IDs. CIs are unadjusted; Holm correction applies only to the p-values across all eight contrasts.

Endpoint	Comparator	Core Δ	95% CI	Holm p
PW	CoT	+37.500	[33.667, 41.500]	1.69×10^{-65}
PW	SC_{16}	+38.333	[34.333, 42.500]	5.15×10^{-62}
PQA	CoT	+4.000	[2.200, 5.800]	4.39×10^{-5}
PQA	SC_{16}	+1.200	[0.200, 2.400]	.07031
FOLIO	CoT	+8.867	[3.448, 14.286]	.00666
FOLIO	SC_{16}	+9.852	[3.941, 15.764]	.00666
ProverQA	CoT	+16.000	[12.200, 19.800]	1.17×10^{-15}
ProverQA	SC_{16}	+15.600	[11.600, 19.800]	2.25×10^{-12}

At 32B, the largest paired gain over SingleView occurs on ProverQA-hard, whereas the high-powered PQA matrix contrast is the only member of this family that remains below .05 after Holm correction. At 70B, Core improves all eight registered contrasts; seven remain significant after joint correction. Thus, the model-scale pattern is not driven by a single endpoint, although effect size and uncertainty remain task dependent.

B FULL MODEL-SCALE ENDPOINT TABLES

Table 6 gives every registered p setting on the same 32B IDs. Full reversal has the highest four-task macro-average, but FOLIO peaks at $p = 0.25$ and PQA is at ceiling for all five values.

Table 6: Qwen2.5-32B premise-order sweep under the fixed two-ordering protocol. Accuracy uses full denominators; Avg is the four-task macro. U counts unresolved outputs as wrong.

p	FOLIO	PQA	PW	ProverQA	Avg	U
0	80.30	100.00	97.17	65.00	85.62	0
.25	**84.24**	100.00	97.50	67.00	87.18	1
.50	83.74	100.00	97.17	68.20	87.28	0
.75	81.28	100.00	97.83	68.40	86.88	0
1	83.74	100.00	**98.33**	**70.80**	**88.22**	0

The fixed 32B component comparison in Table 7 separates the matched-call formal-resampling control from removals of formal repair and source support. The latter removals change which states reach the controller, so their differences are not additive.

Table 7: Qwen2.5-32B controls anchored at fixed $p = 1$. Formal-SC matches calls and token caps, not realized compute. Component removals preserve the full denominator; Δ is the macro change from the parent.

Controller	FOLIO	PQA	PW	ProverQA	Avg	Δ	U
Fixed two-ordering ($p = 1$)	83.74	100.00	98.33	70.80	88.22	0.00	0
Matched-call/cap Formal-SC	61.08	99.60	91.50	43.20	73.85	−14.37	287
No formal repair	82.27	100.00	97.83	65.80	86.47	−1.74	0
No support intervention	82.27	99.60	76.50	61.00	79.84	−8.38	0

Table 8 first summarizes the two replication backbones. Tables 9 and 10 then expose every endpoint so that the macro averages can be traced to their component scores.

Table 8: Four-task macro-accuracy (%) across the two replication backbones. All rows use the same exact-ID endpoint contracts.

Backbone	Answer only	CoT	SC_{16}	SINGLEVIEW	CORE	Final
Qwen3-8B	60.24	71.17	72.84	75.65	**81.60**	81.25
Llama-3.3-70B	58.84	68.09	68.43	78.37	84.68	**85.50**

Table 9: Qwen3-8B full-denominator accuracy (%) on the same endpoint contracts. A dash is unsupported. The final mean carries the Core value forward where the three-label terminal rule is inapplicable.

Method	FOLIO	PQA	PW	ProverQA	Avg
Answer only	67.00	66.20	54.17	53.60	60.24
CoT	71.43	94.00	64.83	54.40	71.17
CoT-SC$_{16}$	74.38	99.00	64.17	53.80	72.84
SymbCoT	64.70	97.80	64.66	N/A	N/A
RAVEN-SINGLEVIEW	68.97	99.60	82.83	51.20	75.65
RAVEN-CORE	**77.83**	**99.80**	91.17	**57.60**	**81.60**
RAVEN	76.85	99.80	**91.33**	57.00	81.25

Table 10: Llama-3.3-70B accuracy (%) on identical IDs. Seven of the eight Core comparisons against CoT and SC survive joint Holm correction.

Method	FOLIO	PQA	PW	ProverQA	Avg
Answer only	67.98	78.00	45.17	44.20	58.84
CoT	74.38	95.80	56.17	46.00	68.09
CoT-SC$_{16}$	73.40	98.60	55.33	46.40	68.43
SymbCoT	73.40	100.00	80.33	N/A	N/A
RAVEN-SINGLEVIEW	80.30	98.20	79.17	55.80	78.37
RAVEN-CORE	83.25	99.80	93.67	62.00	84.68
RAVEN	**85.22**	99.80	**94.17**	**62.80**	**85.50**

The endpoint tables show different scaling profiles. Qwen3-8B obtains most of its gain on ProofWriter, while Llama-3.3-70B improves over its answer-only and CoT baselines on all four endpoints. The final terminal rule helps the 70B model but is slightly below Core on the 8B macro average, so Core and Final are both reported rather than collapsed into one row.

Table 11 compares the structured controller with formal self-consistency under each item's realized call count and token ceilings.

Table 11: Exact-call-budget Qwen2.5-7B control. Formal-SC receives each item's realized Core call count and token caps. The comparison is restricted to the listed endpoints and protocol.

Endpoint	n	Formal-SC	Core	Δ
ProofWriter	600	59.67	73.67	+14.00
PrOntoQA	500	72.40	78.20	+5.80
FOLIO-v2	203	45.32	75.37	+30.05
ProverQA-hard	500	19.80	52.20	+32.40

The matched control favors Core on every endpoint, with gains from 5.80 to 32.40 points. Table 12 places this comparison alongside the observed call and token costs of the principal 32B methods.

Table 12: Observed Qwen2.5-32B resource use on the same 1,803 items. Token counts are totals per item across all calls and are reported as median/p95; solver time excludes model generation.

Method	Mean LLM calls	Input tokens	Output tokens	Solver ms
Answer only	1.00	374/485	4/4	0/0
CoT	1.00	396/507	92/168	0/0
CoT-SC$_{16}$	16.00	6,336/8,112	1,600/2,399	0/0
RAVEN-Core	5.52	2,218/5,218	918/2,986	6.6/39.2

RAVEN-Core uses fewer calls than SC$_{16}$ but more model output than a single answer-only or CoT pass because it constructs and checks formal evidence. Table 13 next separates repair attempts from repairs that produce executable evidence and from repairs that actually change correctness.

Table 13: Paired audit of the single bounded formal-repair round at 32B and $p = 1$. "Exec." counts triggered items with at least one repaired formalization that passes every gate and returns a solver label. W→C and C→W are final controller correctness transitions relative to the no-repair arm. Extra calls are per item.

Endpoint	Triggered	Exec.	Δ acc.	W→C	C→W	Extra calls
FOLIO	94/203	56/94	+1.48	3	0	0.84
PQA	4/500	4/4	0.00	0	0	0.02
PW	121/600	79/121	+0.50	4	1	0.28
ProverQA	325/500	185/325	+5.00	28	3	1.23

Executable repair is most consequential on ProverQA-hard, where it yields 28 wrong-to-correct and three correct-to-wrong transitions. The remaining first-attempt failure composition is reported below. Table 14 then compares the fixed $p = 1$ parent with alternative replay rules on development endpoints.

Table 14: Development comparison of replay rules for the fixed $p = 1$ mechanism study. Scores are full-denominator accuracy (%) on FOLIO train (FT), ProofWriter dev (PD), and a fixed 500-item PrOntoQA matrix prefix (PM). The selection rule requires at least $+0.25$ macro points with no endpoint loss exceeding 0.50 points; no alternative qualifies.

Replay rule	FT	PD	PM	Avg	Δ	Overrides
$p = 1$ parent	76.92	98.33	100.00	91.75	0.00	0
Raw vote, all	75.82	99.00	100.00	91.61	-0.14	145
Definite vote, all	77.02	98.33	100.00	91.79	$+0.03$	1
Raw vote, non-fast parent	76.72	99.00	100.00	**91.91**	$+0.16$	136
Definite vote, non-fast parent	77.02	98.33	100.00	91.79	$+0.03$	1
Raw vote, parent U	76.92	98.33	100.00	91.75	0.00	0
Definite vote, parent U	76.92	98.33	100.00	91.75	0.00	0
Verified vote, all	75.72	99.00	100.00	91.57	-0.18	144
Verified definite vote, all	76.92	98.33	100.00	91.75	0.00	0

Across 754 repaired formalizations, the initial typed failures are 175 malformed JSON, 30 source-ID mappings, 170 syntax checks, 279 static-signature checks, 67 consistency checks, and 33 unresolved-consistency checks. The parent and no-repair arms have identical first-attempt records on 99.9% of shared ordering/item pairs (100% on FOLIO and PQA); Table 13 therefore reports paired final transitions and does not treat every executable repair as a rescued answer.

C Qualitative recovery examples

We select recovery cases using only recorded evaluation fields, under a rule fixed before inspecting their natural-language content: RAVEN must match the reference label, at least one exact-ID comparator must fail or remain unresolved, verified recovery mechanisms are covered first, and remaining ties are resolved by comparator failure count, trace completeness, and stable item ID. These examples illustrate observed execution paths rather than population-level or causal effects. The serialized records contain solver states and certified source cores but do not always contain an explicit displayed derivation, so no missing derivation object is reconstructed. The first three cases characterize the staged Core diagnostics in Table 2 and the component analyses. Figure 2 traces selective premise-order conflict resolution.

Cross-order recovery

Query: Anne is not white. **Reference:** F.
Answer-only, CoT, and SC_{16} return U; DetermLR and Formal-SC are unresolved, while SymbCoT returns F. The staged Core controller's original direct prediction is U, and its original formal route remains invalid after one repair. The formal result from the reversed ordering is certified as F, with subset-minimal support $\{p_1, p_{15}, p_{19}, p_{22}\}$, and the registered cross-order branch selects F.

Typed formal-repair recovery

Query: The bear does not chase the bear. **Reference:** F.
Answer-only, CoT, SC_{16}, SymbCoT, and DetermLR return U; Formal-SC is unresolved. The staged Core controller's first formal candidate fails the registered syntax check. One typed repair passes validation and execution, yielding F. The certified subset-minimal support is $\{p_5, p_{17}\}$, and the support-aware branch selects F.

Executable-evidence and arbitration recovery

Query: Erin is red. **Reference:** T.
Answer-only, CoT, SC_{16}, and DetermLR return F; SymbCoT returns U; Formal-SC is unresolved. The staged Core controller's direct results from the observed and reversed orderings are F and U, while both validated formal results are T. The registered profile tiebreak therefore selects T. The source-language core audit rejects this core, so this example is evidence for executable formal evidence and arbitration recovery, not for source-support acceptance.

The fourth selected case is the source-support recovery shown in Figure 2.

D ASSURANCE DIAGNOSTICS AND ABLATIONS

The threshold comparison in Table 15 separates full-coverage accuracy-first inference from checked definite and checked all-label selective inference. Accepted accuracy conditions on returned labels, so rows with different coverage are not full-denominator accuracy comparisons.

Table 15: Coverage-reliability trade-off on 1,803 items. Definite returns only T/F; all-label also admits executable U.

Threshold	Returned	Coverage (%)	Errors	Accepted acc. (%)
Accuracy-first $p = 1$	1,803	100.00	189	89.52
Checked definite	618	34.28	2	99.68
Checked all labels	1,214	67.33	195	83.94

This section separates three questions that aggregate release coverage can hide. Table 16 reports endpoint-level response-checker coverage, Table 17 separates definite results from semantic U, and Table 18 identifies the first failed check for accepted and withheld parent predictions.

Table 16: Response-checker outcomes by endpoint. Coverage uses each full denominator; UCB is the Wilson 95% upper bound on accepted error.

Endpoint	Coverage (%)	Accepted	Errors	UCB (%)
PW test (600)	35.33	212	0	1.780
PW full (10,190)	36.08	3,677	0	0.104
PQA matrix (9,000)	77.91	7,012	0	0.055
FOLIO-v2 (203)	0.00	0	0	N/A
ProverQA-hard (500)	0.00	0	0	N/A

Table 17: Label-aware outcomes of the selective multi-ordering threshold. Coverage uses the full endpoint denominator; accuracy conditions on returned labels of that type. Semantic U is absent from the binary PrOntoQA contract.

Endpoint (n)	Definite returned	Definite cov.	Definite acc.	U returned	U acc.
FOLIO (203)	37	18.23	97.30	99	58.59
PQA (500)	334	66.80	100.00	0	N/A
PW (600)	231	38.50	100.00	202	98.51
ProverQA (500)	16	3.20	93.75	295	49.49
All (1,803)	618	34.28	99.68	596	67.62

Table 17 localizes the reliability gap in Table 15. Definite proof or refutation evidence is high precision across the pooled endpoints. Most errors admitted by the all-label threshold are semantic U predictions on FOLIO and ProverQA, where an executable generated theory can still omit a source consequence.

Across the content-disjoint union of PrOntoQA matrix, ProofWriter full, and ProofWriter test (19,790 items), the hard product authorizes 10,901 responses (55.08% coverage) with no observed accepted error; the item-level two-sided Wilson 95% upper endpoint is 0.035%. The same check at the source-context level finds 7,484 contexts with at least one authorized response and no context with an accepted error; the corresponding upper endpoint is 0.051%. Among 248 parent errors, Translation blocks 168 and LocalStep blocks the remaining 80. Among correct responses that are withheld, 7,536 first fail Translation, 1,104 fail GlobalSupport, and one fails CrossViewConsistency. These counts describe natural artifacts, not injected mutations, and make the coverage cost of the observed error interception explicit. On the overlapping diagnostic development slices, it authorizes 455/500 PQA responses and 192/600 PW responses, again with no observed accepted error.

Table 18: Natural-failure waterfall on the content-disjoint 19,790-item pool. Each row is the first failed runtime check. Parent errors and correct responses are separated because both are withheld.

First failed check	Parent errors withheld	Correct responses withheld
Translation	168	7,536
LocalStep	80	0
GlobalSupport	0	1,104
CrossViewConsistency	0	1
All other registered checks	0	0
Authorized by full product	0 errors	10,901 correct

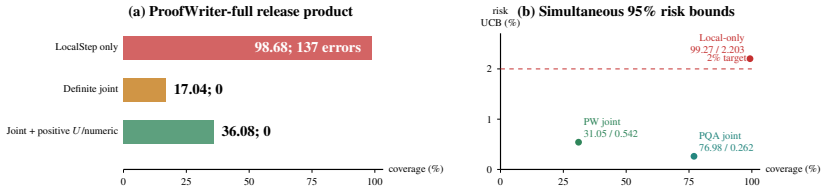


Figure 3: Assurance trades coverage against accepted-error risk. (a) Local replay has high coverage but admits wrong labels; hard source-to-response obligations remove those false assurances, while positive Unknown and numeric witnesses recover coverage without relaxing a type. Values are coverage / accepted errors. (b) LocalStep-only fails the 2% target, whereas the joint selectors pass; annotations give coverage / simultaneous risk UCB.

Figure 3 shows that local proof replay alone offers high coverage but does not control accepted error. The typed product trades some coverage for lower risk. Table 19 tests whether each interface is independently necessary against its registered exact-one mutation family.

Table 19: Controlled leave-one-type-out ablation. Each row removes one release type while retaining the other seven. It then admits every mutation targeted at the removed type; each mutation satisfies all retained predicates. The full product rejects all 1,600 negatives and retains all 200 positives.

Removed type	Target FA	All-negative FA	Removed type	Target FA	All-negative FA
Artifact	200/200	200/1,600	Profile	200/200	200/1,600
Translation	200/200	200/1,600	LocalStep	200/200	200/1,600
Display	200/200	200/1,600	Terminal	200/200	200/1,600
Global	200/200	200/1,600	Scope	200/200	200/1,600
Full-product false accepts		0/1,600	Positive retention		200/200

Every leave-one-type-out system accepts all 200 mutations aimed at the removed interface, whereas the full product rejects all 1,600 negatives and retains all 200 positive controls. This establishes independence for the registered fault families and complements the natural-failure counts above.

D.1 SELECTIVE BEHAVIOR OF THE COMMON CROSS-METHOD CERTIFICATE

For each method m, let A_m be the items admitted by the common Level-C certificate and Q_m the items whose predicted label matches the reference label. Figure 4 partitions each prediction along two independent axes: prediction correctness (Q_m versus Q_m^c) and certificate decision (A_m versus A_m^c). Accepted accuracy is $|A_m \cap Q_m|/|A_m|$, while the correct-withholding rate is $|Q_m \setminus A_m|/|Q_m|$. Reference labels are used only for post-hoc evaluation, never by the certificate. Methods share item IDs within each endpoint, but their correct-prediction sets need not coincide.

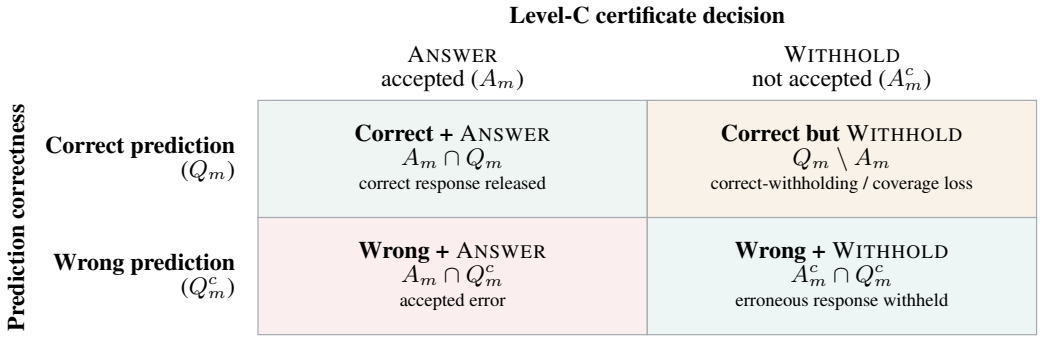


Figure 4: Acceptance–correctness matrix for the common Level-C certificate. The rows indicate whether a method's predicted label matches the reference label, while the columns indicate whether Level C releases (ANSWER) or withholds (WITHHOLD) the response. The upper-right cell, $Q_m \setminus A_m$, is the correct-withholding quantity reported in Table 21; the lower-left cell, $A_m \cap Q_m^c$, contains accepted errors. Reference correctness is used only for post-hoc evaluation and is not available to the certificate at inference time.

Table 20: Accuracy conditional on common Level-C acceptance on identical item IDs. All five accepted subsets have zero observed errors; UCB is the item-level Wilson 95% upper bound on accepted-error rate, not a guarantee of zero future error. This common contract is weaker than native Level R.

Endpoint	Method	n	Correct/ANSWER	Acc. (%)	Error UCB (%)
PW test	RAVEN-Core	600	278/278	100.00	1.363
PW test	LINC_{10}	600	256/256	100.00	1.478
PW test	SymbCoT	600	59/59	100.00	6.113
PQA dev	RAVEN-Core	500	456/456	100.00	0.835
PQA dev	SymbCoT	500	452/452	100.00	0.843

Table 21: Correct predictions withheld by common Level C. The withholding rate divides correct-but-withheld predictions by all correct predictions of that method, not by the full endpoint denominator. Wrong-withheld counts are shown separately; none of the wrong predictions is accepted in these observed subsets.

Endpoint	Method	Base correct	Correct withheld	Withheld/correct (%)	Wrong withheld
PW test	RAVEN-Core	590	312	52.88	10
PW test	LINC_{10}	591	335	56.68	9
PW test	SymbCoT	474	415	87.55	126
PQA dev	RAVEN-Core	500	44	8.80	0
PQA dev	SymbCoT	494	42	8.50	6

Table 21 instantiates the right-hand column of Figure 4. On ProofWriter test, RAVEN-Core withholds 312/590 correct predictions (52.88%), compared with 415/474 (87.55%) for SymbCoT and 335/591 (56.68%) for LINC_{10}. All observed wrong predictions fall in the wrong-and-withheld cell; none falls in the accepted-error cell, although the accepted-error bounds in Table 20 remain nonzero. On PrOntoQA development, RAVEN-Core and SymbCoT withhold 44/500 (8.80%) and 42/494 (8.50%) correct predictions, respectively. The correct-withholding contrast is therefore endpoint-dependent; it is not a universal dominance claim or a native Level-R comparison.

E BENCHMARK AND ASSURANCE CONTRACTS

E.1 FORMAL TASK CONFIGURATION

The main text treats the task configuration as fixed system metadata. Formally, $\pi = (\mathcal{L}_\pi, \mathsf{C}_\pi, \mathsf{E}_\pi, \mathsf{opp}_\pi)$ specifies the formal language, the source-to-formula format, the solver and proof checker, and the operation that constructs the explicit negation of a query. It also fixes the supported logical fragment and answer labels. Soundness refers to the semantics of $\mathcal{L}_\pi$, not merely successful program execution. The configuration is registered and versioned before inference; it is not part of the natural-language problem and is not selected by the model.

The pooled assurance unit is a unique premise-query item. Identity is checked both by normalized upstream source keys and by canonical hashes of premises and query, rather than by dataset-prefixed runtime IDs. Table 22 shows the only relevant subset relations. Hence the pooled set is $\mathcal{D}_{\text{PQA-matrix}} \cup \mathcal{D}_{\text{PW-full}} \cup \mathcal{D}_{\text{PW-test}}$ with cardinality 19,790; the two development slices remain endpoint diagnostics and are not counted again.

Table 22: Cross-endpoint overlap after canonical content matching. PrOntoQA development and ProofWriter development are excluded from the pooled assurance estimate; ProofWriter test is disjoint from ProofWriter full.

Endpoint pair	Left size	Right size	Shared items
PrOntoQA dev / matrix	500	9,000	500
ProofWriter dev / full	600	10,190	600
ProofWriter test / full	600	10,190	0

The overlap audit makes the 19,790-item assurance denominator explicit. Table 23 then records the split, size, labels, scorer, and claim boundary for every endpoint, preventing similarly named but nonidentical benchmark versions from being pooled or compared as if they shared a contract.

Table 23: Frozen benchmark contracts. Sharing a dataset name is insufficient for a head-to-head comparison unless the split, cardinality, label ontology, prompt, model, and scorer also match.

Endpoint	Split / size	Labels	Instance and scorer contract	Claim boundary
ProofWriter test	Logic-LM sampled test / 600	T, F, U	balanced 200/class; ID-joined exact accuracy	same-bank comparison
ProofWriter full	CaRing meta-test / 10,190	T, F, U	complete ordered bank; exact accuracy	full endpoint and paired tests
PrOntoQA 5-hop	development / 500	T, F	one query per story; exact accuracy	same-bank comparison
PrOntoQA matrix	nine development strata / 9,000	T, F	fixed stratum IDs; pooled reconstruction	controlled-domain endpoint
FOLIO-v2	validation / 203	T, F, U	corrected contract; ID-joined exact accuracy	not interchangeable with original FOLIO
ProverQA-hard	hard subset / 500	T, F, U	fixed manifest and label mapping	inference only; no source-faithful certificate

Tables 24 and 25 connect the release interfaces in Section 4 to their trusted assumptions, failure actions, and serialized implementation predicates.

Table 24: Operational contract behind the release interfaces in Section 4. The trusted assumption and failure action are part of each type; generated text is never self-authenticating.

Type	Bound input/output	Checker or attestation	Trusted assumption	Residual failure/action
Artifact	source, AST, proof, display hashes	canonical serializer and SHA-256	canonical bytes; collision resistance	substitution/mismatch; withhold
Profile	AST, witness, profile ID	schema/type checker and registry lookup	registered semantics; version integrity	unsupported/cross-profile; withhold
Translation	source/query and aligned AST	controlled compiler or registered oracle	semantic adequacy within the profile	arbitrary English equivalence is unproved
LocalStep	AST, label, witness ledger	Horn/FOL/CSP profile executor plus replay	executor soundness and declared completeness	timeout/incomplete search; unavailable
Display	proof DAG and rendered graph	deterministic dependency-complete renderer/matcher	renderer serializes the selected witness and every required parent edge	omitted/invented edge; withhold
Terminal	proof end, display end, answer	typed equality check	fixed label ontology	answer-trace mismatch; withhold
Global	all ordering states, audits, branch, label	deterministic replay of δ	support prompts are diagnostics, not proofs	unsupported branch/conflict; withhold
Scope	source IDs, assumptions, dependency set	exact-set, version, and cross-item checks	complete declaration of allowed assumptions	hidden/cross-item premise; withhold

Table 25: Mapping from the eight public interfaces to ten serialized runtime predicates. Scope is a cross-cutting property discharged by three predicates, not an unreported eleventh check.

Public interface	Serialized predicate(s)
Artifact	ArtifactBinding
Profile	LogicProfile
Translation	Translation; EvidenceModeBinding
LocalStep	LocalStep
Display	DisplayBinding
Terminal	TerminalBinding
Global	GlobalSupport; ConsistencyWitness; CrossViewConsistency
Scope	Exact source/dependency fields checked across ArtifactBinding, Translation, and GlobalSupport

The mapping makes Scope explicit as a property checked across artifact, translation, and global-support records rather than as an additional unnamed predicate.

E.2 Release-interface procedures and paired examples

The checker input is the selected, frozen response rather than a fresh model generation. The procedures below expose the decision at each interface; FAIL means that the exact response is withheld. A recorded accepted ProofWriter test item, denoted P below, asks "Bob is green." Its source contains 14 immutable premises. The selected proof uses p_2 (Bob is round), p_{10} (round implies young), p_9 (young implies white), p_{12} (white implies red), p_8 (red and round imply quiet), and p_{13} (quiet implies green). The recorded solver label is T; witness terminal s_6 and display terminal d_6 agree. All ten runtime predicates pass, yielding ANSWER. The complete source, rather than only these six premises, is bound by the source digest and checked by Translation.

For compact identification, P is `ProofWriter_AttNeg-OWA-D5-1213_Q11`; T is `ProofWriter_RelNeg-OWA-D5-136_Q1` (query: the bear visits the lion); L is `ProofWriter_AttNoneg-OWA-D5-698_Q16` (query: Erin is cold); and G is `ProofWriter_RelNeg-OWA-D5-99_Q17` (query: the bald eagle does not chase the bear). All are test items. T, L, G are naturally occurring failed attempts. An asterisk on P denotes a single controlled change to the recorded accepted P response, followed by checker replay; it is not a naturally occurring failure. In particular, the Scope example tests the typed-product projection of P, not a claim that an unstated assumption was observed in the native release pool.

Artifact: identity before proof validity

1: **Input:** frozen typed artifacts and expected source/parent record.
2: Require exactly one artifact of each declared kind and matching item IDs.
3: **for** each artifact in dependency order **do**
4: Recompute its canonical payload digest and compare every upstream digest.
5: **end for**
6: **Output:** pass iff all IDs and digests match; else artifact mismatch.

Pass (P): Input is the recorded source, formalization, six-step witness, visible trace, terminal label, and dependency digests. Recalculation matches every stored binding; output is PASS. **Fail (P^*, controlled):** Input differs only in the formalization digest. Comparison with the frozen expected binding returns ARTIFACT-MISMATCH; the unchanged proof does not rescue the altered artifact graph.

Profile: one declared interpretation

1: **Input:** declared profile, formula fragment, parser and solver revisions.
2: Look up the profile in the fixed registry and check fragment membership.
3: Match the registered parser/solver revisions to the witness profile.
4: **Output:** pass iff registration, fragment, and revisions agree.

Pass (P): Input declares the finite signed-Horn profile and its recorded parser/solver revisions; registry and fragment checks pass, yielding PASS. **Fail (P^*, controlled):** Only the supported-fragment flag is false in the typed checker input. The product returns Profile failure while all other interface results remain unchanged; this is an isolated admissibility test, not a naturally failed translation.

Translation: source versus executable theory

Input: complete source-ID/text list, query, formal AST, evidence mode.
Require a promoted compiler version or registered typed oracle.
Normalize the AST and require exact source-ID multiplicities and query ID.
for each aligned premise and the query **do**
    Compare typed semantics; reject missing, injected, or mismatched formulas.
end for
Output: pass iff every comparison and evidence-mode check passes.

Pass (P): Input has all 14 premise-ID/formula pairs and the query, with controlled-compiler evidence. The normalized typed AST matches all 14 source premises and the query, so output is PASS. **Fail (T, natural):** Input is the original-order formulation with 23 IDs and the query. IDs are complete, but the formulas aligned to p_{17} and p_{22} disagree with the source; typed comparison outputs INVALID. Parser success or solver execution would not discharge this check.

LocalStep: replay the selected formal result

Input: selected formal program, query, claimed label, task profile.
Execute the registered solver again and replay each witness edge.
if the label is U **then**
    Require complete finite closure and absence of both query polarities.
else
    Require a source-indexed proof or counterproof ending at the label.
end if
Output: pass only for a matching, complete replay; else unresolved.

Pass (P): Re-execution derives $round(bob)$ through $green(bob)$, matches the selected T label, and ends at proof step s_6; output is PASS. **Fail (L, natural):** For the original-order input "Erin is cold," complete local replay does not certify the selected result; output is UNRESOLVED, so that ordering supplies no releasable proof.

Display: the proof the reader sees

Input: certified machine witness and proposed visible derivation.
Render the witness deterministically, including IDs and parent edges.
Compare every visible step, dependency, and terminal with the rendering.
Output: pass iff the two complete renderings are identical.

Pass (P): The input witness $s_1, \ldots, s_6$ renders to $d_1, \ldots, d_6$ with the recorded source IDs and parent edges; exact comparison outputs PASS. **Fail (P^*, controlled):** The input visible trace substitutes only the proposition in d_1. It differs from the unchanged solver witness and outputs INVALID. A natural Display failure may also be downstream of an absent local proof, so this isolated change tests Display itself.

Terminal: proof conclusion versus public answer

Input: replayed label and witness end, visible terminal, selected label, answer.
Require a valid proof terminal or complete-closure witness for U.
Compare terminal step IDs and replayed, displayed, and selected labels.
Bind the public answer to that checked selected label.
Output: pass iff witness, display, decision, and answer agree.

Pass (P): Input has witness terminal s_6, visible terminal d_6, selected label, and public answer all at T; equality outputs PASS. **Fail (P^*, controlled):** Only the visible terminal label becomes F while s_6 and the frozen answer remain T. Equality outputs INVALID; replaying the solver alone would miss this response substitution.

Global: the selected evidence, not any valid proof

Input: selected label, source-support audit, acquired-order ledger, local witness, terminal label.
For a definite result, require the frozen source-support audit for its order.
For U, require complete source-aligned finite closure.
Require local, terminal, support, and eligible-order labels to agree.

5: **Output:** pass iff support and cross-record consistency hold.

Pass (P): Input support is $\{p_2, p_8, p_9, p_{10}, p_{12}, p_{13}\}$ for selected T; local, terminal, and acquired-order records agree, yielding PASS. **Fail (G, natural):** The original-order input returns U for "The bald eagle does not chase the bear," but cannot bind that result to the complete source. GlobalSupport outputs UNRESOLVED; a locally executable U is insufficient.

Scope: no hidden premises or crossed tasks

1: **Input:** source/profile/solver bindings, used assumptions, admitted set.
2: Require exact source digest, profile ID, solver ID and revision.
3: Require complete extraction and $W_{\text{used}} \subseteq W_{\text{admitted}}$.
4: **Output:** pass iff bindings and inclusion hold; else withhold.

Pass (P): In the typed projection, input source digest, profile ID, solver revision, and assumption set match; output is PASS. **Fail (P^*, controlled):** Input adds one "unstated" used assumption without admitting it. The product checker fails Scope alone and withholds the response. This is not an estimate of natural hidden-premise frequency.

These paired cases establish what each check accepts or rejects on the specified records. The natural-failure distribution is reported separately in Table 18; controlled changes do not enter that distribution. Likewise, the common cross-method Level-C test in Table 3 is a smaller shared interface, not the complete eight-interface native release product.

F CONTEXTUAL RESULTS AND METHOD POSITIONING

Table 26 gives paper-reported accuracy under each system's original inference protocol. It excludes method-specific fine-tuning. A blank means the source paper did not report that endpoint; protocols differ across rows, so no cross-row statistical comparison is computed.

Table 26: Paper-reported original-backbone accuracy (%) for contextual positioning. Models, subsets, revisions, prompts, call budgets, and label distributions differ; the two local RAVEN rows anchor the reported endpoints.

Method	Venue	Original backbone	FOLIO	PQA	PW	ProverQA
RAVEN-Core	this work	Qwen2.5-32B	81.77	100.00	98.33	69.80
RAVEN	this work	Qwen2.5-32B	84.73	100.00	98.50	71.80
LogicSAGE (Tian et al., 2026)	ICML 2026	Qwen3-8B	75.40	92.30	82.50	N/A
LogicSAGE	ICML 2026	GPT-4	82.26	99.20	83.84	N/A
HBLR (Li et al., 2026)	AAAI 2026	GPT-4	84.22	99.36	89.41	N/A
HBLR	AAAI 2026	DeepSeek-V3	85.22	99.55	89.48	N/A
HBLR	AAAI 2026	DeepSeek-R1	95.60	99.72	92.32	N/A
Adaptive Solver Composition (Xu et al., 2026b)	EACL 2026	GPT-4o	69.10	99.40	94.70	N/A
SIPDO (Yu et al., 2026)	ICLR 2026	GPT-4o	83.20	96.30	79.60	N/A
SIPDO	ICLR 2026	GPT-4o-mini	81.10	91.30	79.30	N/A
Aristotle (Xu et al., 2025)	ACL 2025 Oral	GPT-4o	N/A	99.60	88.50	N/A
Aristotle	ACL 2025 Oral	Claude-3.5-Sonnet	N/A	99.00	86.50	N/A
Aristotle	ACL 2025 Oral	Llama-3.1-405B	N/A	98.40	89.50	N/A
CLOVER (Ryu et al., 2025)	ICLR 2025	GPT-4o	78.80	N/A	96.70	N/A
LogicTree (He & Roy, 2025)	EMNLP 2025	GPT-4o	N/A	N/A	97.00	N/A
LogicTree	EMNLP 2025	Llama-3.3-70B	N/A	N/A	95.80	N/A
COP (Liu et al., 2025)	Findings NAACL 2025	GPT-4o	65.00	N/A	96.50	N/A
Formal Uncertainty (Ganguly et al., 2025)	NeurIPS 2025	o3-mini + SMT	50.00	99.80	94.18	N/A
SymbCoT (Xu et al., 2024)	ACL 2024	GPT-4	83.33	99.60	82.50	N/A
DetermLR (Sun et al., 2024)	ACL 2024	GPT-4	75.49	98.60	79.17	N/A
Logic-LM (Pan et al., 2023)	Findings EMNLP 2023	GPT-4	78.92	83.20	79.66	N/A
LINC (Olausson et al., 2023)	EMNLP 2023	GPT-4-0613	72.50	N/A	98.30	N/A

Table 27: Design-space positioning. A dash means that a component is not a hard public-release obligation in the cited method, not that the method cannot emit the corresponding artifact.

Method family	Primary target	NL translation	Local proof	Public binding	Selective release
Logic-LM / LINC	answer accuracy	generated	solver	-	fallback/vote
Aristotle	complete search	generated	resolver	trace	iteration cap
CLOVER	FOL translation	composition + verification	solver	-	fallback
FoVer	trace validity	trace-to-FOL	Z3	reasoning text	-
NSF-CoT	CoT faithfulness	text-to-logic	SMT + judge	step/context scores	scored
CRV	CoT verification	computation graph	graph checks	graph/trace	-
Formal uncertainty	artifact reliability	SMT generation	solver	-	uncertainty gate
RAVEN	**response assurance**	**source-bound**	**replay**	**display + terminal**	**8-way product + risk**

The contextual accuracy and design-space comparisons in Tables 26 and 27 place RAVEN among adjacent methods while keeping their evaluation protocols distinct.

G Operational algorithms and reproducibility

G.1 Primary accuracy-first decision rule

The following rule specifies the staged controller in Table 2. It uses no reference label. For order v, let d_v be its parsed direct label and f_v its executed formal label; an unavailable result is $\perp$, distinct from semantic U. For a definite formal label, proof-core extraction checks the same polarity by solver replay. A source-support audit then tests whether the retained premises elicit the formal label. For nonempty support it also removes one deterministically selected premise and requires a semantic U answer; empty support has a separate accepted verdict. Call the label *source-supported* only after the applicable tests pass. A failed extraction that ends unresolved supplies no contrary evidence. A replayed non-entailment vetoes its label unless another ordering certifies that label.

First obtain (d_0, f_0) from the original order. If they agree on a definite label and its core is source-supported, return it. Agreement on U requests the reverse order rather than terminating. If the original definite formal label has source-supported core evidence, return it even when d_0 disagrees. Every other case obtains both direct and formal results (d_1, f_1) from the reversed order.

Apply the following priorities in order, returning at the first applicable line. "Stable" means that both ordering labels in the named channel are present and agree; a channel consensus is its unique most frequent available label, with a two-way tie yielding no consensus.

1. If direct and formal channels have the same consensus and it is not vetoed, return it when it is U or has source-supported core evidence. This checkpoint precedes extraction of the reversed-order proof core.
2. Extract and audit the reverse definite formal core, if any. Return its label if source-supported. Otherwise, if exactly one definite label is source-supported across the orderings, return that label.
3. If exactly one label has a machine-certified core, return it, except when a stable, non-vetoed definite direct label disagrees and the source audit explicitly rejects that core in favor of the direct label. In that case, return the direct label. If certified cores have conflicting labels, return a stable, non-vetoed definite direct label, or unresolved if none exists.
4. Without a decisive core, return the first non-vetoed label in this sequence: stable direct consensus, stable formal consensus, direct consensus, formal consensus, original direct label, original formal label. An explicit source-audit rejection matching a stable direct label selects that label at the stable-direct step. If no label remains, return unresolved.

For the RAVEN row, a final zero-call rule changes one branch of the parent decision rule (64 item labels across the applicable three-label endpoints): on a three-label task, when the preceding rule selects the *stable-direct* branch and both original and reverse formal results are U, output U instead. Other branches and binary tasks are unchanged. The selective checked-vote policy in Table 28 is separate and is specified by the conflict-resolution procedure below.

Nested perturbation. For $m = \lfloor N/2 \rfloor$, let τ_i swap premise-ID pairs at positions i and $N - i + 1$. With $B_i(p) = \mathbf{1}[U_i < p]$, the reordered source is

$$\omega_p = \prod_{i=1}^{m} \tau_i^{B_i(p)}, \qquad T_p(P) = ((i_{\omega_p(1)}, p_{\omega_p(1)}), \ldots, (i_{\omega_p(N)}, p_{\omega_p(N)})). \tag{5}$$

For each mirrored pair, derive U_i from a stable content hash of the dataset ID, example ID, and the two positions. Swap iff $U_i < p$. This makes treatment assignment independent of labels and nested across p. The normalized Kendall distance and expected rank correlation obey

$$\mathbb{E}[d_K(\omega_p, I)/\tbinom{N}{2}] = p, \qquad \mathbb{E}[\rho_K(T_p(P), P)] = 1 - 2p. \tag{6}$$

Subset-minimal support. For a definite solver label s_p, the retained source support S_p must satisfy

$$\mathrm{Solve}(Z_p|_{S_p}, q) = s_p \quad \wedge \quad \forall a \in S_p : \mathrm{Solve}(Z_p|_{S_p \setminus \{a\}}, q) \neq s_p. \tag{7}$$

Execute the full AST for the selected polarity. Initialize S from the profile-specific proof/unsatisfiable core. Iterate over $a \in S$ in canonical source order; remove a when re-execution still returns the selected polarity. After convergence, re-run every one-deletion check. Emit a certificate only if the full support replays and every deletion changes the result.

Conflict resolution. Acquire the $p = 0$ and $p = 1$ formal results and check each independently. If both admissible labels match, return that label; if exactly one result is admissible, return its label; and if neither is admissible, return unresolved. When both results are admissible but disagree, acquire and check $p = .5$. Return a label only when it receives two checked votes and belongs to the registered answer space; otherwise return unresolved. Answer-only and CoT remain separate baselines, and no decision reads the reference label.

Table 28: Complete selective checked-evidence decision rule. The midpoint is acquired only after a conflict between two admissible endpoints.

Endpoint evidence	Midpoint	Inference result
$a_0 = a_1 = y$	none	y
Exactly one endpoint gives y	none	y
Neither endpoint gives a label	none	unresolved
$a_0 \neq a_1$, both admissible	acquire $a_{.5}$	continue below
$a_{.5} \in \{a_0, a_1\} \cap \mathcal{Y}_\pi$	acquired	majority label
Three distinct labels	acquired	unresolved
$a_{.5} = \perp$ or vote outside $\mathcal{Y}_\pi$	acquired	unresolved

Artifact and claim boundaries. Every evaluated row is bound to a dataset contract containing the ordered ID set, split, label ontology, reasoning profile, and content digests. Model and tokenizer revisions, prompts, seeds, token caps, solver configuration, raw completions, parser status, calls, and latency are recorded before an independent scorer accesses labels. Unsupported endpoints remain absent rather than being filled from a different subset or backbone. Contextual numbers reported by original papers are not used as controlled head-to-head evidence because their backbones, prompts, budgets, and dataset revisions differ.

The implementation uses a deterministic ordering constructor, schema and AST validators, profile executors, support-deletion replay, arbitration replay, the release checker, and exact-one mutation generators. Aggregate and paired statistics are reconstructed from per-item evaluation records. Each evaluated response bundle binds its source, formalization, witness, display, terminal answer, and selected branch by content digest; these bindings permit deterministic replay of the recorded release decision without treating generated prose as a trusted certificate.

H PROOF DETAILS

H.1 JUSTIFICATION OF THE CONDITIONAL SOURCE-TO-RESPONSE GUARANTEE

Because the Translation obligation passes, the selected formalization Z preserves the source semantics under the registered translation assumption. Order the selected witness G topologically. Scope binds

each base node to an admitted source premise or registered assumption, and sound replay licenses each derived node inductively under the registered executor. Because Display is dependency-complete, the rendered derivation D preserves every displayed node together with the proof dependencies required by the selected witness. Terminal binds the final displayed conclusion to the terminal answer, while Global binds that answer to the frozen inference result. Artifact and Profile ensure that these objects refer to the same committed source and registered reasoning semantics. Hence an accepted response has the selected label licensed by the source under the stated assumptions, and its visible terminal conclusion agrees with that label.

Proposition 1 (Threat-relative non-substitutability)**.** *Let $\Theta = \{\theta_k : k \in \mathcal{K}\}$ contain one realizable mutation per registered interface, and suppose c_k^- fails only A_k. Then the product $\bigwedge_{j \neq k} A_j$ is unsound against Θ for every omitted type k.*

Proof. The reduced product accepts c_k^- because every retained checker passes, but c_k^- realizes the registered threat θ_k. Hence at least that threat bypasses the reduced product. This establishes independence within the declared registry, not universal minimality over all representations. □

Proposition 2 (Finite-Horn forward/backward refinement)**.** *For a finite, function-free, range-restricted signed Horn program P and literal ℓ, complete forward closure derives ℓ if and only if tabled backward search from ℓ returns a finite source-bound proof. Every returned backward proof is a subgraph of a forward derivation.*

Proof. Soundness is induction over the returned proof: leaves are source facts and each internal node applies a registered Horn rule to proved antecedents. For completeness, assign each literal in the least fixed point its first derivation height and induct on height. Tabled search can choose the generating rule and recursively prove strictly lower-height antecedents. Finiteness, range restriction, and tabulation terminate cycles. □

Perturbation distance. A mirrored transposition $(i, N - i + 1)$ changes $2(N - 2i + 1) - 1 = 2N + 1 - 4i$ pairwise order relations. The transpositions are disjoint, and their inversion contributions sum to $\binom{N}{2}$. Taking expectations in Equation 6 gives p. Since normalized Kendall correlation is $\rho_K = 1 - 2d_K/\binom{N}{2}$, linearity then gives the second identity in Equation 6.

Finite Horn replay. For a finite, function-free, range-restricted Horn program, complete forward closure derives a literal iff tabled backward search returns a finite proof. Soundness follows by induction on the returned proof. For completeness, assign each literal in the least fixed point its first derivation height; induction on height lets backward search choose its generating rule and recursively prove strictly lower-height antecedents. Finiteness and tabulation terminate cycles.