MEASURING REASONING REQUIRES LETTING MODELS REASON

Anonymous authors
Paper under double-blind review

ABSTRACT

Logical reasoning in language models is evaluated one inference rule at a time, and often under a prompt that asks for a bare `True`/`False` label. We built a matched-arm benchmark to measure what a weak rule costs inside a chain—three arms sharing a four-step derivation and differing only in the form of one rule—and under that answer-only format the result is dramatic: across five frontier models a rule with a negated premise is applied at 0.999 and its contrapositive at 0.209, with GPT-4.1 correct 0 times in 600 attempts while every other step of its chain is at 1.000. Asked for the same contrapositive on its own, GPT-4.1 is correct on 100 of 100 instances and rejects 100 of 100 foils, so the rule is present and is not a surface cue. The natural reading is that chain context destroys a rule the model has. It is the wrong reading. Lifting one sentence from the system prompt, so the model may write its derivation before answering, moves GPT-4.1 from 0.000 to 0.613 on the chain and flattens a depth sweep that had fallen $1.000 \rightarrow 0.000$ over eight derived steps to $1.000 \rightarrow 0.917$ on the same instances. What the answer-only format measures is whether a model can perform an inference *without externalising it*, which is not the question such benchmarks are taken to ask, and the gap between the two questions can be the entire result. One model is not rescued: GPT-5.2 answers the contrapositive correctly 0 times under every condition we tried—answer-only, internal reasoning tokens, and explicit step-by-step—while applying the negated-premise rule at 1.000 throughout. Code, data and per-run logs: https://anonymous.4open.science/r/lemo-D708/README.md.

1 INTRODUCTION

Tell GPT-4.1 that Anne is rough, give it the rule *"if someone is not young then they are not rough"*, and ask whether Anne is young. It answers correctly on 100 of 100 instances, and on 100 matched instances whose fact does not fire the rule it correctly declines to conclude anything. Now derive *"Anne is rough"* through two prior steps instead of stating it, leave the rule, the wording and the question untouched, and the same model answers correctly 0 times in 600 attempts, while answering the two steps that precede it at 1.000.

The obvious conclusion is that chain context destroys a rule the model demonstrably has. We drew it, and it is wrong. The system prompt in both conditions ended *"Always answer with ONLY True or False"*—a common instruction, and the one our own earlier work used. Delete that sentence, let the model write its derivation first, change nothing else, and GPT-4.1 goes from 0.000 to 0.613 on the chain, while a depth sweep that had fallen from 1.000 to 0.000 over eight derived steps flattens to 1.000–0.917 on the same instances. The model could always run the implication backwards. It could not do so without writing it down, and the benchmark had forbidden that.

What we set out to measure. The field evaluates logical reasoning by decomposing it into rules tested in isolation—one rule per instance in LOGICBENCH, atomic skills in LOGICASKER—which is what makes those benchmarks diagnostic and also what leaves the compound cost unmeasured: a failed step caps every conclusion downstream of it, and an aggregate cannot separate a primary failure from its consequences. We built matched arms to measure that cost (§4), and under an answer-only prompt the numbers are the ones above. The format control is what reinterprets them.

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.

Controls. Perturbation benchmarks assign labels deterministically, so a perturbed split is single-class and a constant response saturates it. Every claim below is made against a stated majority baseline and a matched condition under which the capability is absent (§5).

Contributions. **(i)** *The answer format can be the whole result.* On matched arms that differ only in the form of one inference, we measure the same models under an answer-only prompt and under one that permits visible steps (§4.3). For GPT-4.1 the contrapositive moves $0.000 \rightarrow 0.613$ inside the chain, and the depth curve goes from a collapse to a flat line. Benchmarks of this shape measure whether a model can perform an inference without externalising it; that is a legitimate question, but it is not the question they are read as asking, and here the two answers differ by the size of the effect being reported.

(ii) *The controls that let us catch it* (§5). Every accuracy is paired with a matched condition under which the capability is absent—a foil—and a stated majority baseline. The foils are why we could tell that GPT-4.1's isolated 1.000 is a capability rather than a cue, and the same discipline applied to our own earlier training work removes most of what it claimed (§6).

(iii) *One failure that survives every condition.* GPT-5.2 answers the contrapositive correctly 0 times in 450 attempts under answer-only, 0 in 150 with visible steps, and 0 in 100 in isolation, while applying a negated-premise rule at 1.000 in all three. Neither reasoning budget nor externalisation recovers it.

2 Related Work

Evaluating logical reasoning, one rule at a time. The field has converged on a decomposition by inference rule. LogicBench (Parmar et al., 2024) covers 25 patterns with *a single inference rule* per instance; LogicAsker (Wan et al., 2024) defines 34 atomic rules, expands them into 208 skills and reports per-skill accuracy, finding failure rates from 29% to 90%. Both establish a result we rely on: modus tollens is substantially harder than modus ponens. Isolating each rule is what makes those designs diagnostic and also what leaves the compound cost unmeasured, since in a chain a failed step caps every conclusion downstream of it.

Premise validation and contradictory inputs. PCBench (Li et al., 2025) evaluates premise critique across 15 models and finds that most rely on explicit prompting to notice an error and that general reasoning ability does not predict premise-critique ability. RULEBREAKERS (Chan et al., 2025) studies the complementary case, where formal validity and world knowledge disagree.

RULEBREAKERS also supplies the precedent for our controls: its perturbation determines the label, so it scores matched instances jointly and a constant policy earns zero. We reached the same requirement from the opposite direction, after our own single-class splits awarded 1.000 to a constant.

What the model is allowed to write down. Chain-of-thought prompting made visible intermediate steps a standard part of how models are queried and of why they succeed (Wei et al., 2022). Evaluation has not kept pace: a benchmark scoring an exact-match label often asks for the label and nothing else, because a free-form derivation is harder to parse. Our matched arms were built that way, and so was our own earlier evaluation. The point of §4.3 is not that chain-of-thought helps, which is established, but that the size of a reported reasoning failure can be set by the format alone, and that an aggregate reported without the format stated is not comparable to one reported with it.

Directional asymmetry. That the failing rule is the contrapositive connects to the reversal curse (Berglund et al., 2023): a model trained on "A is B" does not thereby answer "B is A". That concerns retrieval of a memorised relation on single facts; ours concerns applying a rule given in the prompt, inside a derivation, and largely recoverable once the model may write the steps out. Not the same phenomenon, but the same direction.

What verifier rewards can and cannot install. On the training side, RLVR is reported to sharpen rather than extend: RL-trained models win at pass@1 while base models overtake them at large k, indicating the solutions were already in the base distribution, and exploration narrows (Yue et al., 2025). Closer to our setting, outcome-only rewards are known to collapse onto a degenerate constant—a small static refusal reward drives near-total abstention—and reinforcement finetuning

reduces refusal on unanswerable questions by over 80%, repaired by mixing unanswerable examples back into training (Song et al., 2025). Our verifier-reward runs reproduce this collapse under an exactly computable oracle (§6.2); we attribute it not to reward noise but to a surface cue in the training distribution, and report in §6.4 that the cue is present with probability 1.000 and that supervised training exploits it as well.

3 BENCHMARK AND SEMANTICS

Every split used below is generated from one rule-based system, so an instance and its perturbation differ in a single controlled way and nothing else. This section fixes that generator and the semantics its labels carry; §4 uses it to build the matched arms, and §6.2 to build the two-class control.

3.1 STRUCTURAL ROBUSTNESS BENCHMARK

The benchmark is both a diagnostic tool and a training corpus. Unlike conventional benchmarks composed of heterogeneous tasks, all evaluation instances are derived from a single canonical rule-based system with systematically controlled perturbations; each variant preserves domain semantics and vocabulary while differing only in structural properties, isolating reasoning robustness from domain shift.

Canonical backbone and perturbations. All instances derive from one canonical backbone (Appendix A.21) and are probed along four axes: redundant versus essential rule deletion; contradictory-fact injection; semantics-preserving rewrites (De Morgan, contrapositive, double negation); and stacked rewrites. **Variant 2** removes a key rule on the deduction chain, breaking downstream entailments; **Variant 3** injects a contradictory fact. All splits share one CSV schema (`facts`, `rules`, `questions`, `answers`).

Contradiction Semantics. We adopt a *conservative reasoning semantics*: when the premise set is inconsistent, the reasoning process is logically invalid and deduction must halt. Formally, with Γ the premises and Q the queries, if $\Gamma \vdash \bot$ then $\text{Label}(q) = \texttt{False}$ for every $q \in Q$. Throughout, $\vdash$ denotes derivation under our forward-chaining engine on Horn-clause rules (Algorithm 1), *not* classical $\vdash_{\text{CL}}$, which would explode under contradiction. This keeps contradiction detection measurable with a standard binary metric under one consistent semantics.

Scoring. The perturbation splits use *instance-level exact match*: correct iff every query matches, so a model persisting with its original answers on Variant 3 scores 0.0000 (Appendix A.14).

4 WHAT THE ANSWER FORMAT DECIDES

The instrument. Three arms share a four-step chain, the same entities and query positions, and differ only in the form of the third rule: **MP** (*"If someone is rough then they are young"*), **MP¬** (*"If someone is not sleepy then they are young"*), and **MT** (*"If someone is not young then they are not rough"*, the contrapositive of MP). Q1 and Q2 probe the steps before the manipulated rule, Q3 the rule itself, Q4 the step after it. MP¬ separates the two candidate explanations: a model that mishandles negated antecedents should fail it too, while one that cannot run an implication backwards should pass it and fail only MT. Every instance is oracle-verified; we draw 150 per arm (1800 questions per model).

One instance. Facts `Anne is green or blue | Anne is not green`; rules `green→cold`, `blue→cold`, `cold→rough`, the manipulated rule, `young→nice`, one distractor; queries `cold`, `rough`, `young`, `nice`, gold `T|T|T|T`. In the MT arm, Q3 means reading *"not young → not rough"* with the derived *"Anne is rough"*; predicates come from a pool of 20.

The two formats. **Answer-only**, the format our earlier work used and a common one, ends the system prompt *"Always answer with ONLY True or False for each question, in order"*. **Step-allowed**

lifts that sentence and asks for a final parseable block after the work. Nothing else changes. Under answer-only GPT-4.1 emits 1.8 output tokens per question; under step-allowed, 49.

4.1 Under answer-only, the contrapositive collapses

Table 1: Accuracy on the manipulated inference (Q3), answer-only format, mean ± s.d. over n runs drawing different halves of the instance pool. Upstream is Q1 and Q2 over all three arms. Run-to-run s.d. is at most 0.024 on every column but MT.

Model	n	Upstream	MP	MP¬	MT
GPT-4o	4	0.964	0.968	1.000	0.107 ± 0.026
GPT-4.1	4	1.000	0.993	1.000	$\mathbf{0.000 \pm 0.000}$
GPT-5-nano	3	0.999	1.000	1.000	0.096 ± 0.043
GPT-5-mini	3	1.000	1.000	1.000	0.602 ± 0.056
GPT-5.2 (medium)	3	0.952	0.960	1.000	$\mathbf{0.000 \pm 0.000}$

MP¬ is at least 0.998 in every row while MT averages 0.209; GPT-4.1 answers MT correctly 0 times in 600 attempts and GPT-5.2 0 in 450, upstream at 1.000 and 0.952. The failure is not negated antecedents. A wrong Q3 takes Q4 with it in 76–100% of cases, and since each arm is a quarter of the questions the collapse surfaces as an aggregate of 0.80–0.93. Raising `reasoning_effort` to medium never improves MT (-0.013, -0.026 and 0.000 on GPT-5-nano, -mini and GPT-5.2, three runs each).

4.2 The rule is there when asked for on its own

We ask each arm for exactly the inference it asks at Q3, with the upstream steps replaced by a fact and the downstream step removed. Since a model could separate "*young* occurs in a rule" from "*nice* does not" without inferring anything, each arm is paired with a foil carrying the same rule and a fact that does not trigger it, making the gold answer `False` so that predicate presence alone scores 0.500. All five models answer MP and MP¬ at 1.000 on both halves.

Table 2: The contrapositive in three conditions. *Isolation* removes the chain, 100 instances per cell; *trigger* fires the rule, *foil* shares the rule but not the fact so a policy keying on predicate presence scores $1.000/0.000$, and *paired* scores both halves of one instance jointly (* determined by halves of $1.000/1.000$ or $0.000/1.000$). *In chain* is Q3 of Table 1 under each answer format; the two GPT-5 rows marked — were not run under step-allowed. Upstream, MP and MP¬ are at 0.927–1.000 everywhere.

	In isolation			In chain		
Model	trigger	foil	paired	answer-only	step-allowed	Δ
GPT-4.1	1.000	1.000	1.000*	**0.000**	**0.613**	+0.613
GPT-4o	0.970	0.980	0.905	0.107	**0.407**	+0.300
GPT-5-mini	1.000	1.000	1.000*	0.602	—	—
GPT-5-nano	0.440	1.000	0.370	0.096	—	—
GPT-5.2	0.000	1.000	0.000*	0.000	0.000	0.000

GPT-4.1 applies the contrapositive on 100 of 100 isolated instances and rejects 100 of 100 foils: the capability is present and is not a surface cue. GPT-5.2 is the opposite case—`False` on trigger and foil alike is a constant on that arm, so the rule is absent for it and nothing about context needs to be invoked to explain its chain score.

4.3 Letting the model write the steps restores most of it

The depth sweep makes the point sharply. Holding the rule, its wording, the query and the oracle fixed and varying only d, the number of steps that must be derived before the rule fires, GPT-4.1 under answer-only goes $1.000, 0.217, 0.033, 0.000, 0.000$ for $d = 0, 1, 2, 4, 8$. Under step-allowed, on the same instances, it goes $1.000, 0.917, 0.883, 0.900, 0.917$, with modus ponens at 1.000 and every foil at 1.000 throughout. **The depth effect is a property of the format, not of the model.** An earlier draft of this paper read that first row of numbers as evidence that chain context destroys a rule the model has; it is evidence that the model cannot execute the rule without writing the derivation down.

Two things follow. The certain one: a benchmark that constrains output to a bare label measures whether a model can perform the inference *without externalising it*, a different and harder question, and the gap between the two can be the whole result. The tentative one: for GPT-4.1 a residual gap remains, since the same step-allowed runs give MP 1.000, MP¬ 1.000 and the isolated contrapositive 1.000 against 0.613 inside the chain. Something is still lost; we can no longer attribute most of it to the chain.

4.4 One model the format does not rescue

GPT-5.2 at medium effort answers MT correctly 0 times out of 450 under answer-only, 0 out of 150 under step-allowed, and 0 out of 100 in isolation, while answering MP¬ at 1.000 in every one of those conditions. It has internal reasoning tokens and it has been given an explicit invitation to use visible ones; neither recovers the inference. For this model the contrapositive is not a capability that context suppresses. It is not there.

5 Controls

Every number above is paired with a condition under which the capability is absent. This is not bookkeeping; without it none of the claims can be told apart from a constant response, and we found that out the hard way.

Perturbation fixes the label, and a constant then scores perfectly. A benchmark built by perturbing a base instance assigns the perturbed split's labels deterministically. Under the conservative semantics of §3, every query on a contradictory instance is `False`, so that split is single-class and its majority-class baseline is 1.000: a model answering `False` on sight scores perfectly while checking nothing. The same holds for any split whose perturbation determines the answer. §6 is this paper's own case study: an objective that scored 1.000 on such a split answers `False` on all 12,000 questions of a two-class control in two runs out of four.

The foil. For the measurement in §4 the corresponding control is the foil: the same rule, a fact that does not trigger it, and the same query, whose answer is now `False`. A model keying on "the queried predicate occurs in some rule" answers `True` and is wrong, so predicate presence scores 0.500 rather than 1.000. Foils are why GPT-4.1's isolated 1.000 is a capability rather than a cue, and why GPT-4o's depth columns can be marked as unattributable rather than reported as a failure of the inference.

What else we report. Majority baselines, prediction distributions, structural rather than row sample sizes, and spread over seeds rather than questions—each changed a conclusion we had written down.

6 The training pipeline this project began as

This section is the case study for §5: the work this project began as, and what the controls did to it. It targets a different failure from §4—premises that contradict each other rather than a rule that goes unused—and we keep the two apart.

6.1 Training objectives

Stage 1 (structural SFT) fine-tunes on canonical, perturbed and contradiction-containing variants, every target opening with the verification preamble, and initialises everything else (Figure 1). **Conflict-aware SFT** continues from it on traces that carry out the check and halt when it fails; it is the only objective whose claims survive §6.2. **RLVF** trains from the same initialisation with a symbolic forward-chainer as reward:

$$\mathcal{L}_{\mathrm{RLVF}} = -\mathbb{E}_{\tau\sim\pi_\theta}\big[(r-b_t)\,\log \pi_\theta(\tau|\mathbf{x})\big],\quad r = \mathbb{1}[\Pi(\tau)=a^*]-\mathbb{1}[\Pi(\tau)\neq a^*],\quad b_t = \mu b_{t-1}+(1-\mu)\bar{r}_t, \tag{1}$$

Table 3: Trained methods on Qwen2-1.5B, re-measured in September 2026 with the released checkpoints. **V3 and V3-mixed measure the same capability and disagree completely.** Variant 3 is single-class—every query on a contradictory instance is False—so its baseline is 1.000 and a model answering False on sight scores perfectly while learning nothing. V3-mixed (§6.2) interleaves contradictory instances with consistent ones of the same surface form, at a 0.5835 baseline that only Fusion-LRA exceeds. Each row names its evaluation path: a two-logit classification head and free generation are not equally hard. Header row 2 gives each split's majority baseline; $\rho_\top$ is the fraction of V3-mixed items answered True (gold 0.417). † not significantly above baseline (p=0.127); every other V3-mixed figure differs from it at $p<10^{-240}$.

Method	**Path**	**Base** *0.549*	**V2** *0.503*	**V3** *1.000*	**V3-mixed** *0.5835*	$\rho_\top$
Stage-1 SFT	classifier	1.000	1.000	0.004	0.432	0.98
+ LIRE	classifier	0.688	0.590	0.000	0.417	1.00
+ RLVF	classifier	1.000	1.000	0.000	0.433	0.98
RLVF (generative, corrected oracle)	generative	0.767	0.799	1.000	0.590†	0.02
Fusion-LRA (conflict-aware SFT)	generative	0.870	0.610	0.982	**0.698** ± 0.054	0.34

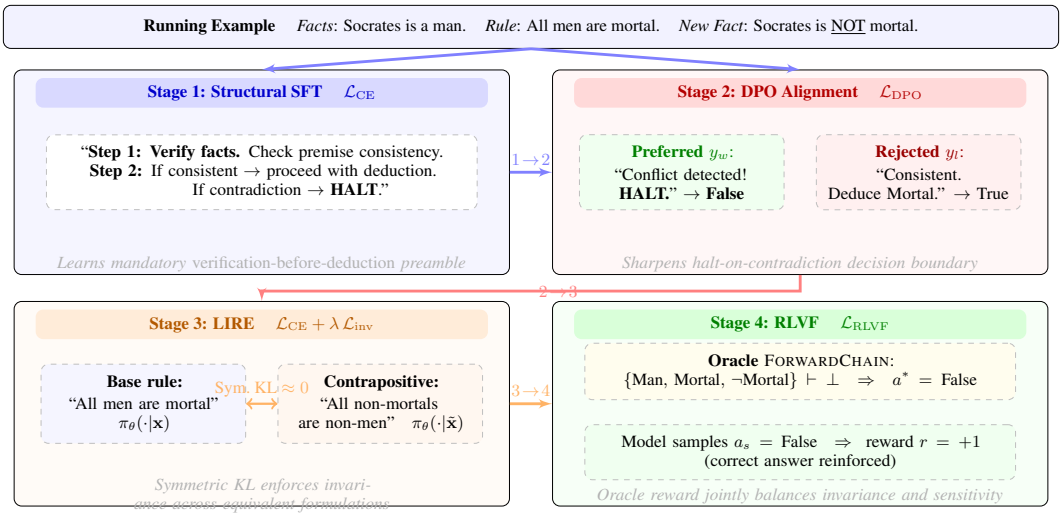


Figure 1: The training pipeline on a running contradiction example. **Stage 1** (top-left, SFT) teaches the verification-before-deduction preamble and is the shared base; the other three panels are objectives trained *from* it and compared, **not** a left-to-right chain. RLVF (bottom-right) uses a symbolic oracle reward. Only conflict-aware SFT survives the control of §6.2.

where b_t is an EMA baseline over the mini-batch reward. §6.2 reports that it collapses to a constant responder in four seeds out of four, and §6.4 why. Two further objectives, DPO and LIRE, are in Appendix A.6.

6.2 A TWO-CLASS CONTROL FOR CONTRADICTION DETECTION

Variant 3 is the single-class case of §5: all-False by construction, baseline 1.000, so the 0.982 and 1.000 in Table 3 are consistent with detecting every contradiction and with never checking. Nor does the trace settle it—the word "conflict" appears in 99.98% of Fusion-LRA's Variant-3 traces and in 95.0% of its *base*-split traces, where there is nothing to detect.

The two-class control, `variant3_mixed`, is 3,000 instances (12,000 questions) in three classes sharing one surface form: a base chain plus exactly one extra negated fact, which contradicts the chain (**contra**), negates one disjunct while the other still fires (**consistent-disj**), or concerns something a chain with one rule removed can no longer derive (**consistent-mixed**). The third class discriminates: its extra fact is frequently the *same sentence* that makes a base instance contradictory, harmless only because the rule that would derive it is gone. Every row is oracle-checked; labels are 41.7% True for a baseline of 0.5835.

Table 4: Per-class behaviour on the control, mean $\pm$ s.d. over six seeds (Fusion-LRA) and four (RLVF). The discriminating quantity is the gap between the consistent and contradictory True-rates: 0.420 ± 0.200 for Fusion-LRA (p=0.0018 against zero), with the variance almost entirely in **cons.-disj**. RLVF answers False on essentially everything (0.5830 ± 0.0065 against the 0.5835 baseline, p=0.44; two of four seeds answer False on all 12,000 questions).

		contra	**cons.-disj**	**cons.-mixed**
gold True rate		*0.001*	*1.000*	*0.247*
Fusion-LRA	% True	**0.096** $\pm$ 0.108	**0.516** $\pm$ 0.295	0.098 $\pm$ 0.076
accuracy		0.903 $\pm$ 0.108	0.516 $\pm$ 0.295	0.676 $\pm$ 0.067
RLVF (gen.)	% True	0.023 $\pm$ 0.039	0.038 $\pm$ 0.055	0.001 $\pm$ 0.001
accuracy		0.977 $\pm$ 0.039	0.038 $\pm$ 0.055	0.753 $\pm$ 0.001

Two consequences. The contradiction-halting claim survives for conflict-aware SFT alone: gaps of 0.420 ± 0.200 (n=6) against 0.014 ± 0.018 (n=4) with no overlap between seed sets (Welch p=0.0020), every other objective at or below a constant responder. And the failure mode is over-flagging: the model catches contradictions reliably (0.903) but rejects plainly consistent premise sets about half the time (0.516 ± 0.295), which carries nearly all the seed variance. A system built on this would escalate far more often than it should.

Rows are not independent samples. Each instance instantiates a fixed template, so the row count overstates the evidence: counting distinct (facts, rules) pairs, `variant1` has **30** structures across 1000 rows and our control 329 across 3000—replication factors of 33$\times$ and 9$\times$. The matched-form and depth splits introduced here draw predicates from a pool and are effectively unreplicated.

Any interval over questions is therefore too narrow: resampling whole structures widens Fusion-LRA's by 3$\times$ and RLVF's by 8$\times$, and the corrected $[0.5224, 0.6683]$ comfortably contains the baseline. We rebuilt the control with its chain drawn from a twenty-predicate pool (3000 rows, 3000 structures, baseline 0.5803) and re-evaluated all ten checkpoints. Estimates fall—conflict-aware SFT $0.6979 \to 0.6244$, its gap $0.420 \to 0.318$—and the spread collapses with them, the gap's s.d. $0.200 \to \mathbf{0.053}$: **the seed instability we reported earlier was mostly replication noise.** The comparison sharpens rather than weakens, separating at $p<10^{-4}$ against verifier-reward RL, whose own gap is 0.000 ± 0.006.

A symbolic solver wins outright on this benchmark, and that bounds what it can show. Asked for a systematic comparison against symbolic and verifier-based methods, we ran the obvious one end to end: parse the premises into logic, then forward-chain. Across all seventeen splits it achieves **coverage** 1.000 **and accuracy** 1.000. This is not a surprise—the labels are computed by the same procedure—but stating it plainly settles something. **Wherever the premises are already formalised, or are written in a grammar a parser covers, nothing here competes with a solver, and no result on this benchmark should be read as evidence that it does.** What the benchmark can do is diagnose model behaviour, which is how we use it.

The same pipeline has **coverage** 0.000 on LogicNLI and MNLI: not one instance of real natural language is parseable. That gap is the only regime in which a learned approach has a role—and §6.8 reports that ours does not generalise there either, transferring to premise-consistency tasks and not at all to first-order reasoning. We state both halves because reporting only the first would overclaim and only the second would be incomplete.

6.3 Main Results

The failure targeted here is that a model given premises that contradict each other keeps deducing from them instead of halting. We call it **Logic Inertia**; it is a different failure from the one in §4, since here the rule is applied and it is the premise set that has failed. The baseline pattern is uniform across BERT, Qwen2 and TinyLlama (Table 7, Appendix A.9): all three stay at 1.000 on the base task and Variant 1, drop to 0.25–0.295 under essential-rule removal (Variant 2), and collapse to 0.000 under explicit contradictions (Variant 3).

Table 3 summarises the trained objectives on Qwen2-1.5B. Figures marked ‡ come from single-class splits and are not evidence; they are kept only to show what §6.2 changes.

The capability gets worse with scale. At Qwen3-8B, four seeds against six, the same objective is better where contradiction detection is not needed (base 0.9075 vs 0.870) and significantly worse on the control: gap 0.146 ± 0.071 against 0.420 ± 0.200 (Welch p=0.0096). It stays above the 0.5835 baseline (p=0.0204), so this is a weaker capability rather than none. Our first 8B run was the worst of the four and supported a stronger claim than the data does; the best would have supported none.

6.4 THE CAPABILITY IS PARTLY A SURFACE CUE

The control measures the test set; it does not audit the training set. Doing so changes the interpretation of everything above. Across the training distribution, an instance whose facts contain a negation has answer `False` with probability $\mathbf{1.000}$ ($792/792$ sampled), while instances without one are `False` only 38.2% of the time. Contradiction injection is the only perturbation that adds a negated fact, so "a negation is present" and "the premises are inconsistent" are perfectly confounded in training.

A policy can therefore satisfy the objective by keying on the cue, and both of ours partly do. It explains the verifier-reward collapse without appealing to reward noise: the reward is exact, but the cue is free, and on `variant3_mixed`—where the consistent classes also carry a negated fact—the policy that learned it answers `False` on essentially everything. It also explains why conflict-aware SFT, which does discriminate (0.420 ± 0.200), is weakest exactly where the cue misleads.

The standard repair—adding consistent-but-negated counterexamples—failed informatively: the hardest class rose from 0.550 to 0.969 while contradiction detection fell from 0.861 to 0.444, and the retrained model emitted "no conflict" on 100% of inputs. It exchanged one cue for another. A two-class control on the test set is therefore necessary but not sufficient when the training distribution contains the shortcut.

The cue reappears on an independently built benchmark. Every instance of the matched-form split carries a negated fact—disjunction elimination leaves one behind—and every gold answer is `True`; our checkpoints answer `True` on 19.2% (1.5B) and 31.4% (8B) of it, with the three arms indistinguishable ($0.192/0.040/0.194$). The frontier models separate those arms sharply on the same file. What transfers from training is the reflex, not the inference.

6.5 IS THE GAP PROMPT-INDUCIBLE? FRONTIER MODELS AND INFERENCE-TIME VERIFICATION

A training-time fix is only interesting if the behaviour is not simply promptable. We test two frontier models zero-shot, under a verification-first prompt, and behind a neuro-symbolic guard in the spirit of Feng et al. (2026) (Table 5). Neither closes the gap, and contradiction handling does not track general capability: Claude Opus 5, near-perfect on Base and V2, is *worse* on V3 than the weaker GPT-5.2.

What the failure looks like. Given $\{A \to B,\ A,\ \neg B\}$, GPT-4o often surfaces the inconsistency in its trace and proceeds anyway, overriding the $\neg B$ it just acknowledged (Appendix A.8). We read this as detection and halting being separate capabilities, but the reading is qualitative: §6.4 shows trace mentions are not a detection signal.

6.6 EXPERIMENTAL SETUP

The benchmark has 100 base groups (80 train / 20 test), yielding $11{,}200$ Stage-1 SFT instances and $2{,}160$ preference pairs. All objectives use AdamW + bf16 + LoRA (r=8, α=16); hyperparameters are in Appendix A.16 and ablations in Appendix A.18. The pipeline fits on one A100 (40 GB) at 8B: 5.1 GPU-hours at 1.5B and 9.7 at 8B (Appendix A.12).

Objective ablation. Comparing objectives over a shared base was this work's original argument, and §6.2 removes most of it; the stage-wise figures are kept in Appendix A.17, unverified against the current data.

6.7 DISCUSSION

We withdraw two arguments from the earlier draft. The Variant-3 ranking was read as showing that reliability follows from structural organisation rather than data exposure; those figures are single-

class, and the ablation table is kept in Appendix A.18 without that reading. And we argued that an exactly computable reward makes the trust region of PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024) unnecessary, using plain REINFORCE (Eq. 1): with no trust region the EMA baseline climbed to $+0.695$ and the policy degenerated into non-parsing output at step 250 (Appendix A.11). Unbiasedness constrains the reward's expectation, not the estimator's variance. A second failure is not variance at all—"answered False after detecting the contradiction" and "answered False unconditionally" earn *identical* reward, so an outcome-only signal is degenerate here (Table 4).

Inference-time guards (Feng et al., 2026; Xiong et al., 2025; Qin et al., 2026) re-invoke a verifier on every query, whereas a prior amortised into the weights costs nothing at runtime.

6.8 Out-of-distribution transfer: what the prior does *not* buy

Saturating a synthetic benchmark invites the objection that the policy has memorised a template family. We evaluate the checkpoints zero-shot on **LogicNLI** (Tian et al., 2021), the contradiction subset of **MNLI** (Williams et al., 2018), **FOLIO** (Han et al., 2024) and **ZebraLogic** (Lin et al., 2025) (Table 6, details in Appendix A.1).

The result is negative and bounds our claim. At 1.5B every checkpoint degenerates into a constant-`False` predictor, scoring exactly the majority baseline on LogicNLI and chance on MNLI; at 8B the collapse inverts rather than resolves (96% of LogicNLI labelled True, accuracy below baseline) even though that checkpoint saturates our own benchmark. The one positive is FOLIO at 8B (0.619 vs. 0.500); ZebraLogic moves at neither scale.

What the pipeline installs is robustness of the verify-then-deduce structure *within* the propositional regime it is trained on, not a general reasoning upgrade.

6.9 Formal verification of the benchmark's labels

Every label here comes from the forward-chainer that also computes the RLVF reward, so we audited it against a Lean 4 kernel: agreement is 0.9965 on derivable queries in a five-domain corpus but 0.6119 here and 0.2213 on non-derivable ones, where proving non-entailment needs a closed-world assumption our translator does not encode. **We therefore do not use Lean as an independent adjudicator** (Appendix A.2).

7 Limitations

One vendor. The five models span two generations but one provider, three of them one family; we claim the effect for these models, not frontier models in general. **The grid is incomplete.** Both formats were run for GPT-4.1, GPT-4o and GPT-5.2 and answer-only alone for GPT-5-nano and GPT-5-mini. **The scope is one rule.** That modus ponens is unaffected under either format shows the effect is not a property of chains as such, not which other rules behave like the contrapositive. **No mechanism.** The effect is not negation handling, a surface cue, upstream breakage or a reasoning budget, and most of it is the answer format; why the contrapositive should be the inference a model cannot execute silently, we do not know. **Single runs, and one arm we cannot attribute.** The matched arms are three to four runs per cell, the isolated and depth sweeps one each; and GPT-4o loses accuracy on every arm with depth, foils included, so its MT column mixes the effect with general long-chain failure, and the MP$\neg$ foils of GPT-4.1 and GPT-4o fall to 0.500 in one cell—the opposite error. We flag both rather than average them away.

8 Conclusion

GPT-4.1 applies the contrapositive on 100 of 100 isolated instances and 0 of 600 once two derived steps precede it. The control that should have come first was the answer format: deleting the instruction to answer with only a label moves it to 0.613 and flattens the depth collapse. Such a format measures whether a model can reason *without externalising it*, not the question it is read as asking. GPT-5.2 alone resists every condition, applying a negated-premise rule at 1.000 and its contrapositive 0 of 700.

Ethics Statement

This work studies logical robustness of language models on a synthetic, propositional rule-based benchmark generated from a single canonical template; it involves no human subjects, no personally identifiable information, and no scraped web content. The transfer evaluations use the publicly released LogicNLI and MNLI corpora under their original licences. The failure mode we characterise—*Logic Inertia*—is a safety-relevant one: a model that continues to deduce from contradictory premises will confidently produce unsound conclusions. Our mitigation trains models to halt rather than to continue, which we regard as the conservative direction. We see no dual-use concern specific to this work beyond those general to open-weight reasoning models.

Reproducibility Statement

The benchmark generator, the training objectives (SFT, DPO, LIRE, RLVF), the symbolic forward-chaining oracle, and the Lean 4 verifier are all released in the anonymous repository linked from the abstract. Contradiction semantics and the evaluation metrics (instance-level exact match and $\mathrm{SR}_{\mathrm{macro}}$) are defined in §3; benchmark generation and all structural-variant templates are given in Appendix A.21 and the following subsections; Stage-1 hyperparameters and the DPO preference-pair templates are listed in the Appendix. All reported numbers come from held-out splits generated by the released generator with the seeds recorded in the repository.

The Use of Large Language Models

Large language models were used only as a general-purpose writing aid (copy-editing and LaTeX formatting) and as coding assistance during implementation. They were not used to generate research ideas, to design experiments, or to produce any of the reported results. All experimental claims were verified by the authors against the released code and logs. The authors take full responsibility for the content of this paper.

References

AlphaProof and AlphaGeometry Teams. AI achieves silver-medal standard solving International Mathematical Olympiad problems. Google DeepMind Blog, 2024. URL https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: LLMs trained on "A is B" fail to learn "B is A". *arXiv preprint arXiv:2309.12288*, 2023.

Jason Chan, Riza Batista-Navarro, and Goran Nenadic. RULEBREAKERS: Challenging LLMs at the crossroads between formal logic and human-like reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2410.16502.

Ernest Davis. Benchmarks for automated commonsense reasoning: A survey. *ACM Computing Surveys*, 56(4):Article 81, 2024a.

Ernest Davis. Testing GPT-4-o1-preview on math and science problems: A follow-up study. *arXiv preprint arXiv:2410.22340*, 2024b.

Ernest Davis. Mathematics, word problems, common sense, and artificial intelligence. *Bulletin of the American Mathematical Society*, 61(2):287–303, 2024c.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Proceedings of the 28th International Conference on Automated Deduction (CADE)*, pp. 625–635, 2021.

Yu Feng, Nathaniel Weir, Kaj Bostrom, Sam Bayless, Darion Cassel, Sapana Chaudhary, Benjamin Kiesl-Reiter, and Huzefa Rangwala. VeriCoT: Neuro-symbolic chain-of-thought validation via logical consistency checks. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2511.04662.

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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 22017–22031, 2024. arXiv:2209.00840.

Jinzhe Li, Gengxu Li, Yi Chang, and Yuan Wu. Don't take the premise for granted: Evaluating the premise critique ability of large language models. In *Findings of the Association for Computational Linguistics (EMNLP)*, 2025. arXiv:2505.23715.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 37889–37905, 2025. arXiv:2502.01100.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.739/.

Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.

Yuehan Qin, Shawn Li, Yi Nian, Xinyan Velocity Yu, Yue Zhao, and Xuezhe Ma. Don't let it hallucinate: Premise verification via retrieval-augmented logical reasoning. *Transactions on Machine Learning Research (TMLR)*, 2026. arXiv:2504.06438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Linxin Song, Taiwei Chen, Yoshinari Fujinuma, Tanmay Gupta, and Robin Jia. The hallucination tax of reinforcement finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://arxiv.org/abs/2505.13988.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3738–3747, 2021.

Yuxuan Wan, Wenxuan Wang, Yiliu Yang, Youliang Yuan, Jen-tse Huang, Pinjia He, Wenxiang Jiao, and Michael R. Lyu. LogicAsker: Evaluating and improving the logical reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://aclanthology.org/2024.emnlp-main.128/.

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. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 1112–1122, 2018.

Siheng Xiong, Ali Payani, Yuan Yang, and Faramarz Fekri. Deliberate reasoning in language models as structure-aware planning with an accurate world model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31900–31931, 2025.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *Advances in Neural Information Processing Systems*, 2025. URL https://limit-of-rlvr.github.io/.

A Appendix

A.1 Inference-time baselines and external transfer

Table 5: Frontier models and inference-time verification versus the trained structural prior (800 questions/split; GPT-4o V2 n=636). **The V3 column of this table is measured on the single-class split and we no longer draw a conclusion from it.** An earlier draft read the verification-first prompt as *degrading* contradiction handling, because V3 there is all-False and a prompt that makes a model answer False less often registers as a loss. On the two-class control the sign reverses: the same prompt scores 0.7767 against 0.7633 without it (§6.2). The external guard likewise catches only contradictions visible in the raw facts, not those requiring derivation.

Setting	Model	Base	V2	V3 (contradiction)
Zero-shot	GPT-4o	0.789	0.818	0.560
Zero-shot	Gemma-3-4B-IT	0.580	0.360	0.439
Zero-shot	GPT-5.2	0.730	0.846	**0.825**
Zero-shot	Claude Opus 5	0.943	0.993	0.679
Verification-first few-shot	GPT-5.2	0.751	0.889	0.800
Verification-first few-shot	Claude Opus 5	0.999	1.000	0.648
Naive LLM + external symbolic guard	GPT-5.2	0.784	0.903	0.648
Naive LLM + external symbolic guard	Claude Opus 5	0.970	1.000	0.648
Fusion-Conflict (Qwen2-1.5B, trained)	—	**1.000**	**1.000**	**1.000**
Fusion-Conflict (Qwen3-8B, trained)	—	1.000‡	1.000‡	1.000‡

Table 6: Zero-shot out-of-distribution transfer. **The result is negative and we report it as such.** $\rho_\top$ is the fraction of items labelled True; a value near 0 or 1 means the classifier has collapsed to a constant, so its accuracy carries no information about reasoning. LogicNLI labels are rebuilt from the preserved `original_label` field after we found the released preparation script had silently labelled all 500 rows `False` (Appendix A.7)—a file on which a constant-False predictor scores 1.000, which is exactly what $\rho_\top$ is here to expose.

	LogicNLI (n=500)		**MNLI-Con** (n=349)		**FOLIO**	**Zebra**
Model	Acc	$\rho_\top$	Acc	$\rho_\top$	(n=811)	(n=626)
Majority-class baseline	*0.750*	—	*0.501*	—	*0.500*	*0.500*
Stage-1 SFT (Qwen2-1.5B)	0.750	0.00	0.510	0.03	0.433	0.470
+ LIRE (Qwen2-1.5B)	0.750	0.00	0.510	0.04	—	—
Fusion-Conflict (Qwen2-1.5B)	0.750	0.00	0.507	0.03	0.438	0.462
Fusion-Conflict (Qwen3-8B)	0.282	0.96	**0.599**	0.58	**0.619**	0.495

A.2 Lean 4 translation, and the withdrawn RLVF-Lean extension

Retained from the earlier draft. The kernel-rewarded RL variant it describes has no checkpoint and no training script, and the training code never calls the Lean reward; it was a feasibility demonstration presented as an extension.

A.3 Phase 2: Scaling the Verifier with a Formal Proof Assistant

The propositional ForwardChain oracle cannot verify quantified reasoning, induction, or mathematical derivations (Davis, 2024a;c;b), motivating a proof assistant as the natural successor. **RLVF-Lean** translates the policy's CoT into Lean 4 (de Moura & Ullrich, 2021) tactic applications and rewards each prefix by the kernel verdict ($+1$ on success, -1 if the tactic fails or introduces inconsistency, 0 if incomplete), converting RLVF's sparse terminal reward into a dense step-grounded signal akin to process reward models (Lightman et al., 2024) but with kernel soundness: every verdict is a deductive check, not a prediction. A hand-crafted pilot achieves $8/8$ kernel agreement at 0.36 s/trace on CPU (Appendix A.19). At benchmark scale we auto-translated a stratified 80-row (187-question) sample of the four test splits (translator specified in Appendix A.13); scored against each query's *classical* polarity—the value derivable *before* the conservative-semantics override, which is why Variant 3 still contains 12 classically-derivable T-cases—the kernel agrees on $104/105$ T-cases (99.0%) and $30/82$ F-cases (36.6%), broken down per split in Appendix A.10.

The T-case result is strong evidence that **Lean soundly verifies the multi-step derivations RLVF rewards**. The asymmetric F-rate is not a translation bug but a semantic distinction: LEMO is *closed-world*—a claim is `F` whenever it is not derivable—whereas Lean is classical and open-world, where `F` requires proving $\neg P$. Variant 2 makes this stark (0%): removing a rule renders a conclusion unreachable, not refutable. Two fixes close the gap—a closed-world axiom schema converting "unreachable" into $\neg$-derivable, or using Lean only for T-verdict step-level reward and the forward-chainer for F-verdicts. Our pipeline implements the latter.

A.4 Running example across objectives

Retained from the earlier draft. Three of its four panels show objectives whose claims §6.2 withdraws, so it illustrates what was tried rather than what is claimed.

A.5 Structured two-phase generation

Every trained trace is required to open with an explicit verification phase before any deduction: *"Step 1: Verify facts"*, followed by either "Facts are consistent" and the derivation, or "Conflict detected" and a halt. The prior being installed is the *ordering*—validate the premise set, then deduce—rather than any particular wording, and §6.4 reports how far short of that the trained models fall: the phrase appears on 95% of instances with nothing to detect, so its presence carries almost no information. Appendix A.4 works a contradiction instance through each objective.

The core design imposes a two-phase reasoning structure—*verify, then deduce*—inside the Chain-of-Thought (CoT) generation path; this is a constraint on the trace, not a modification of the neural architecture. Deductive rules cannot be applied until the model has explicitly validated premise consistency (Figure 1); on detected contradiction or missing dependency the model halts. The objectives that train this behaviour are detailed in §6.1.

A.6 Retracted objectives: LIRE and DPO

Retained verbatim from the earlier draft, for the record. The controls in §6.2 and §6.4 withdraw the claims made here: LIRE's invariance gain is an artefact of all-True equivalence labels, and the DPO branch cannot be measured on the generation path.

Why trace-level objectives are not enough. SFT and DPO both act on the reasoning *trace*, and both leave *surface-level pattern matching* intact: either can reach perfect accuracy on the original formulations yet drop sharply ($1.000 \to 0.327$ on De Morgan variants) once an equivalent rule is rephrased. The remaining two objectives act at the *output-distribution* level instead.

Stage 3 (branch): Logical Invariance REgularisation (LIRE). Let $\mathbf{x}$ be a base premise-rule input, $\tilde{\mathbf{x}}$ its logically equivalent reformulation (De Morgan, double-negation, contrapositive), and $y = (y_1, \ldots, y_T)$ the gold trace from the SFT data. LIRE penalises divergence between the next-token distributions under the two inputs, evaluated by *teacher-forcing the same gold* y on both

branches (which aligns the distributions token-wise and avoids variable-length rollout mismatch):

$$\mathcal{L}_{\text{LIRE}} = \mathcal{L}_{\text{CE}}\big(\pi_\theta(\cdot|\mathbf{x}),\, y\big) + \lambda\, \mathcal{L}_{\text{inv}}, \;\; \mathcal{L}_{\text{inv}} = \tfrac{1}{2T}\sum_{t=1}^{T}\big[\text{KL}_t(\mathbf{x}\|\tilde{\mathbf{x}}) + \text{KL}_t(\tilde{\mathbf{x}}\|\mathbf{x})\big], \tag{2}$$

where $\text{KL}_t(\mathbf{x}\|\tilde{\mathbf{x}}) := \text{KL}\big(\pi_\theta(\cdot|\mathbf{x}, y_{<t}) \,\|\, \pi_\theta(\cdot|\tilde{\mathbf{x}}, y_{<t})\big)$. Symmetry avoids the mode-seeking/covering asymmetry of unidirectional KL; the shared y keeps the KL exact rather than approximated. This exactness carries a cost worth naming: teacher-forcing one gold trace on both branches presumes the two formulations admit derivations of the same length. For equivalence classes where they genuinely differ—a contrapositive rewrite introducing an extra negation-elimination step, say—the late-position KL measures "how surprised is the policy, under the reformulated premises, that the *original-form* continuation still holds" rather than comparing two length-matched independent derivations. LIRE still measurably improves invariance (V4-multi $0.741 \rightarrow 0.999$), but it is an approximation with this specific, nameable failure mode.

Limitation of LIRE. LIRE improves invariance (Variant 4 accuracy $0.741 \rightarrow 0.999$) but suppresses *any* distribution shift across rule reformulations, including those that *should* change the answer (Variants 2/3): empirical sensitivity drops from 1.0 to 0.47. The invariance and sensitivity objectives are therefore in tension *within a single loss*, which is what motivates replacing the regulariser with an oracle reward rather than stacking a fourth stage on top of it.

A.7 Limitations and Future Work

The benchmark is propositional and synthetic; saturation evidences that the structural prior *generalises across backbones*, not that contradictions are universally solved—V4-multi at 1.5B drops to 0.773 after Stage 4 (a capacity bottleneck rather than a structural-prior failure; §6.9), and the Lean F-rate gap on the closed-world / classical semantics divide is genuine. The pipeline also commits to a single *conservative* contradiction semantics; paraconsistent or priority-based resolution strategies (Appendix A.21 ff.) are out of scope. Three further limitations were surfaced in review and we state them explicitly. **(1) External transfer is largely absent.** Zero-shot, every 1.5B checkpoint collapses to a constant-`False` predictor on LogicNLI and MNLI, scoring exactly the majority-class baseline; the 8B checkpoint collapses to constant-`True` on LogicNLI, scoring *below* it; FOLIO at 8B (61.9%) is the only positive signal and ZebraLogic moves at neither scale (§6.8). We therefore do not claim the prior escapes the synthetic template family. We also note a data-preparation defect we found and corrected: the released LogicNLI preparation script silently labelled all 500 rows `False`, on which a constant-`False` predictor scores 1.000; all figures reported here use labels rebuilt from the preserved `original_label` field. **(2) Oracle/label coupling in Phase 1.** The RLVF reward is the same forward-chaining engine that defines the benchmark labels. This is true by construction in the propositional regime and is a property of Phase 1, not a claim that RLVF works detached from a ground-truth source; it is precisely why Phase 2 replaces the oracle with an externally specified Lean 4 kernel (§6.9). **(3) LIRE's length assumption.** Teacher-forcing a shared gold trace presumes length-matched derivations across equivalent formulations, which some rewrites violate (§6.1). Three concrete next steps follow: (i) end-to-end RLVF-Lean training amortised through a persistent LeanDojo server, replacing the current verification-only deployment; (ii) lifting the closed-world override via a CWA axiom schema, or via the hybrid forward-chain/Lean policy of §6.9 that uses Lean only for T-verdicts and the symbolic forward-chainer for F-verdicts; (iii) evaluation on math/proof benchmarks (e.g., MiniF2F, ProofNet) where the Lean kernel is the natural verifier and where stacked-rewrite depth substantially exceeds the 2–5 laws stress-tested in V4-multi.

A.8 Qualitative Analysis: Why Frontier Models Fail

To understand the failure mode of frontier models, we performed an error analysis on GPT-4o using the OpenAI API on a stratified 200-row sample per split (the per-question prediction CSVs are released alongside the benchmark). On Variant 3, where every query has ground-truth `False` under conservative reasoning semantics, GPT-4o produces the correct halt-on-contradiction answer in only 56.0% of the 800 test questions; the remaining 44.0% continue deductive inference despite the explicit contradiction in the premises. A representative failure: given premises $\{A \rightarrow B,\, A,\, \neg B\}$, GPT-4o often surfaces the inconsistency in its verification trace and then proceeds to conclude B from $\{A \rightarrow B, A\}$, stating "Since A is true and A implies B, then B must be true," effectively ignoring its

	BERT		Qwen2		TinyLlama	
Split	Acc	Δ	Acc	Δ	Acc	Δ
base	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant1	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant2 (Essential Deletion)	0.2950	-0.7050	0.2500	-0.7500	0.2500	-0.7500
variant3 (Contradiction)	0.0000	-1.0000	0.0000	-1.0000	0.0000	-1.0000
variant4-multi (Stacked Laws)	1.0000	0.0000	0.6450	-0.3550	0.9925	-0.0075

Table 7: Baseline Performance: Accuracy and deviation (Δ) across structural variants without fusion optimization.

own prior verification of $\neg B$. This suggests that frontier models suffer from a compartmentalization of verification and execution, where the "reasoning momentum" of the deductive steps overrides contradictory evidence. To rule out a vendor-specific artefact, we replicate the analysis on **Gemma-3-4B-IT** (an open-weight frontier model from a different lab, evaluated locally with identical prompts and decoding); it scores *lower* on V3 (0.439) than GPT-4o, continuing past the contradiction in 56.1% of cases, and shows the same qualitative failure pattern. Fusion-Conflict mitigates this by fusing the verification signal directly into the hidden state representation, forcing the model to reconcile the conflict before proceeding.

A.9 Baseline Failure Pattern Across Backbones

A.10 Lean 4 Benchmark-Scale Verification: Per-Split Breakdown

Table 8: Lean 4 verification on LEMO benchmark rows (20 rows per split, stratified; 187 questions). **Polarity is each query's *classical* label**—derivable from the premises *before* the conservative-semantics override of §3—not the per-instance "everything-is-`F`" override applied to Variant 3 elsewhere; this is why Variant 3 still contains 12 classically-derivable T-cases, and why the table tests whether Lean can perform the derivation rather than whether it reproduces the override. Variant 1 draws all-T queries because removing *redundant* rules preserves derivability of the original T-queries, so the stratified 20-row draw contained 42 T- and zero F-queries. The T/F gap is analysed below.

	Ground truth = T			Ground truth = F		
Split	**#T**	**Accept**	**Rate**	**#F**	**Accept**	**Rate**
base	29	28	96.6%	32	12	37.5%
variant1 (redundant removal)	42	42	100.0%	0	0	—
variant2 (essential removal)	22	22	100.0%	20	0	0.0%
variant3 (contradiction)	12	12	100.0%	30	18	60.0%
Overall	**105**	**104**	**99.0%**	**82**	**30**	**36.6%**

A.11 Why we thought plain REINFORCE would suffice, and why it did not

We initially gave three reasons why plain REINFORCE should suffice: a deterministic ± 1 reward has low gradient variance, the EMA baseline already removes the running mean, and the oracle offers no reward-hacking surface. Only the third survives. Without a trust region the EMA baseline climbed to $+0.695$, making a wrong trace $5.6\times$ more costly than a correct one was valuable, and the policy collapsed into non-parsing output at step 250; a KL penalty against the initialisation was required to train the full 600 steps. A deterministic reward bounds bias, not estimator variance. Separately, an outcome-only reward cannot distinguish detecting a contradiction from answering False unconditionally—both earn the same return—so it is degenerate for this capability regardless of the optimiser (Table 4).

A.12 Compute Footprint per Stage

Table 9 reports measured wall-clock time per stage, taken from the training logs of the exact checkpoints released with the code (`trained_models/qwen`, `qwen_lire`, `qwen_rlvf` and their Qwen3-8B counterparts), so each row is traceable to a specific run rather than an estimate. The whole pipeline is a few GPU-hours on a single device; the dominant cost at 8B is Stage 4, whose

rollouts are generation-bound rather than gradient-bound. Dataset construction adds no annotation cost: the generator is template-driven and combinatorial (Appendix A.14), so scaling to more base groups means increasing N in the generator, not hiring annotators.

Table 9: Measured wall-clock training time per stage on a single GPU (LoRA + bf16), read from the released training logs. Because the objectives are branches off one base (§6.1), the released model costs only Stage 1 + Stage 4; the DPO and LIRE branches are additional runs used for the comparisons in Tables 3 and 10, not prerequisites. The Qwen3-8B Stage-1 run is a short 180-step LoRA pass. Lean kernel verification (Phase 2) is a separate $\sim$0.36 s/trace CPU cost incurred only during training.

Stage	**Qwen2-1.5B**	**Qwen3-8B**
Stage 1 — Structural SFT (shared base)	4.15 h	0.02 h
Stage 4 — RLVF (final model)	0.94 h	9.69 h
Cost of the released model	**5.09 h**	**9.71 h**
Stage 2 — DPO branch (comparison only)	0.90 h	—
Stage 3 — LIRE branch (comparison only)	1.26 h	4.26 h

A.13 LEMO-TO-LEAN TRANSLATION

Section 6.9 evaluates a Lean 4 kernel against benchmark instances translated automatically from our CSV schema. The translator (`lean_demo/lemo_to_lean.py`) is deliberately small and total on the benchmark's surface grammar; we specify it here so the Lean results are independently reproducible.

1. **Parse.** Each fact, rule, and question is matched against a closed set of natural-language templates: positive facts (*"X is P"*), disjunctive facts (*"X is P or Q"*), negated facts (*"X is not P"*), rules of the form *"If something is P then something is Q"*, and their negated variants. Any instance containing a string outside this grammar is rejected rather than silently mistranslated.
2. **Declare.** Every distinct predicate occurring anywhere in the instance is declared as a Lean `Prop`. The benchmark is propositional, so no quantifier machinery is emitted.
3. **Emit hypotheses.** Facts and rules become hypotheses of the generated theorem. Disjunctive facts are discharged by $\vee$-elimination, producing one branch per disjunct.
4. **Emit theorem and proof skeleton.** For each question the translator emits a theorem statement encoding the expected answer, together with a proof script of chained `have` statements mirroring the steps of the symbolic forward-chaining oracle. The oracle is therefore reused to produce the *proof skeleton*, not merely the label—which is what makes the kernel's verdict a check on the derivation rather than on the answer.
5. **Dispatch.** The generated `.lean` file is compiled by the `lean` kernel and the accept/reject verdict recorded. A worked multi-step example beyond the toy Socrates case is included in the released code alongside the generated `.lean` artefacts.

Two consequences are worth stating. First, the kernel arbitrates *classical* derivability, which is why Table 8 is scored against classical (pre-override) polarities rather than the conservative instance labels. Second, the rejection-on-unparsed policy means kernel agreement is a lower bound restricted to instances the translator fully covers, not an average over silently degraded translations.

A.14 STRUCTURAL VARIANT GENERATION

A.15 PREFERENCE PAIR TEMPLATES (DPO STAGE 2)

For contradiction cases, the preferred reasoning trace y_w correctly detects the inconsistency in the verification step and halts; the rejected trace y_l proceeds with deductive inference.

Preferred (y_w): "Step 1: Detect contradiction between premises. Step 2: Halt reasoning." **Rejected (y_l):** "Step 1: Premises are consistent. Step 2: Continue deductive inference."

Algorithm 1 Structural Variant Generation (Variant 2 & 3)

Require: Canonical generator $\mathsf{Base}(\cdot)$, key rule $r^\star$, contradiction template $\mathsf{Contr}(\cdot)$, instances N
Ensure: Datasets $\mathcal{D}_{base}, \mathcal{D}_{v2}, \mathcal{D}_{v3}$
1: **for** $i = 1$ to N **do**
2: $(F, R, Q, A) \leftarrow \mathsf{Base}(i)$; $\mathcal{D}_{base} \leftarrow \mathcal{D}_{base} \cup \{(F, R, Q, A)\}$
3: $R_{v2} \leftarrow R \setminus \{r^\star\}$; $\mathcal{D}_{v2} \leftarrow \mathcal{D}_{v2} \cup \{(F, R_{v2}, Q, \mathsf{Label}(F, R_{v2}, Q))\}$
4: $F_{v3} \leftarrow F \cup \{\mathsf{Contr}(F, R)\}$; $\mathcal{D}_{v3} \leftarrow \mathcal{D}_{v3} \cup \{(F_{v3}, R, Q, \mathsf{Label}(F_{v3}, R, Q))\}$
5: **end for**

A.16 Stage 1 Hyperparameters

Epochs 3; LR 2×10^{-5}; batch 4; max length 512; AdamW (mixed precision); LoRA r=8, α=16, dropout 0.05.

A.17 Stage-wise objective comparison (legacy figures)

These figures were measured before the benchmark was regenerated on 2026-03-31 and have not been re-verified against the current data; they are retained for continuity with the earlier draft and are not used to support any claim in the main text.

Table 10: Objective comparison over a shared Stage-1 base. **Branches, not a chain**: rows 2 and 3 are each trained from the row-1 checkpoint (§6.1). LIRE buys compositional invariance but *collapses* V2/V3 sensitivity; RLVF, from the same base and drawing half of each batch from the same invariance pairs, reaches oracle-level V2/V3 *and* most of the V4 gain in one run. At 1.5B it retains 0.773 on V4-multi against LIRE's 0.993—a residual we report rather than claim "no loss"; at 8B it matches the LIRE ceiling. Row 1 is the Stage-1 SFT checkpoint, *not* an untrained zero-shot run and with no preference optimisation applied; DPO is a separate branch, evaluated in Table 3.

	Qwen2-1.5B				Qwen3-8B			
Stage	Base	V2	V3	V4-multi	Base	V2	V3	V4-multi
SFT (Stage 1)	1.000	1.000	1.000	0.635	0.333	0.460	0.656	0.006
SFT → LIRE (Stage 3)	0.812	0.590	0.344	**0.993**	0.549	0.503	0.344	**1.000**
SFT → RLVF (St. 4)	**1.000**	**1.000**	**1.000**	0.773	**1.000**	**1.000**	**1.000**	**1.000**

A.18 Ablation Tables

Table 11: Ablation: SFT on a vanilla *Mixed-Aug* corpus *without* the conflict-aware verification preamble. Compared to Fusion-LRA (Conflict-Aware SFT, Table 3, Base = 0.988 on Qwen2), removing the preamble drops Qwen2's base accuracy to 0.5250, isolating the contribution of the structural prior from the contribution of training data alone.

	BERT		Qwen2		TinyLlama	
Split	Acc	Δ	Acc	Δ	Acc	Δ
base	1.0000	0.0000	0.5250	-0.4750	0.5375	-0.4625
variant2 (Essential Deletion)	0.2500	-0.7500	0.4050	-0.1200	0.5325	-0.0050
variant3 (Contradiction)	0.0000	-1.0000	0.0000	-1.0000	0.0000	-1.0000

A.19 Lean Pilot Study (8 hand-crafted traces)

One non-trivial engineering finding: Lean's `sorry` exits with status 0, so the bridge must treat `sorry`/`admit` as -1 rather than trusting the exit code. Production deployment would amortise dispatch via a LeanDojo (Yang et al., 2023) persistent server. Lean complements ForwardChain in expressiveness (quantifiers and induction vs. Horn clauses), granularity (per-tactic verdicts), and compositionality, and is feasible end-to-end given recent neural theorem proving (Yang et al., 2023; Polu & Sutskever, 2020; AlphaProof and AlphaGeometry Teams, 2024).

Before the benchmark-scale Lean evaluation reported in Table 8, we ran a small hand-crafted pilot on the Socrates contradiction example to verify (i) that the Python→Lean bridge faithfully translates a halt-on-contradiction trace into a kernel-checkable theorem, (ii) that valid halt traces

Table 12: Iterative Comparison: Fusion-LRA vs. earlier iterations.

Method	Base Acc	Var 2	Var 3	Rank
Mixed-Aug	0.525	0.405	0.972	3
RA-CoT	0.263	0.593	0.690	2
Fusion-LRA	**0.988**	**0.753**	0.705	**1**

receive reward $+1$ while opportunistic deductive continuations and malformed traces receive -1, and (iii) that per-trace verification latency is compatible with online RL. Table 13 reports kernel agreement on 8 representative traces.

Table 13: Lean 4 step-level reward pilot on the Socrates example. Lean kernel agrees with the expected RLVF-Lean reward on $8/8$ traces, including `sorry`-based admits and syntactic errors. Mean latency 0.36 s/trace on CPU.

Trace ID	Category	Expected r	Lean r	Time (s)
T1-halt-minimal	correct halt	$+1$	$+1$	0.36
T2-halt-direct	correct halt	$+1$	$+1$	0.38
T3-halt-contradiction-tactic	correct halt	$+1$	$+1$	0.36
T4-naive-stops-early	naïve continue	-1	-1	0.37
T5-naive-wrong-goal	naïve continue	-1	-1	0.35
T6-naive-empty (`sorry`)	naïve continue	-1	-1	0.36
T7-syntax-error	malformed	-1	-1	0.36
T8-misspelled-lemma	malformed	-1	-1	0.36
Agreement with oracle				**8/8 (100%)**

A.20 Lean 4 Feasibility Demo for RLVF-Lean (Phase 2)

The following Lean 4 script encodes the Socrates contradiction example from Figure 1 as a verified theorem. It accompanies Section 6.9 (*Phase 2: Scaling the Verifier with a Formal Proof Assistant*) and demonstrates that the step-level reward signal defined by the step-level reward of §6.9 can be sourced from the Lean kernel with no additional model training. The complete source is included in `lean_demo/SocratesContradiction.lean`; a minimal excerpt is shown below.

```
axiom Person   : Type
axiom Socrates : Person
axiom Man      : Person -> Prop
axiom Mortal   : Person -> Prop
axiom men_are_mortal : forall x : Person, Man x -> Mortal x

-- The halt-on-contradiction behavior that RLVF-Lean is rewarded
-- for discovering. Lean accepts the proof iff the policy has
-- genuinely identified both sides of the contradiction.
theorem socrates_contradiction
    (h_man        : Man Socrates)
    (h_not_mortal : Not (Mortal Socrates)) : False := by
  have h_mortal : Mortal Socrates := men_are_mortal Socrates h_man
  exact absurd h_mortal h_not_mortal
```

A Python bridge (`lean_demo/lean_verifier_bridge.py`) synthesizes one such script per policy rollout and returns the kernel's accept/reject verdict as the step-level reward r_t of §6.9. The same theorem additionally refutes the naive "continue deducing" policy: under the contradictory premise set, no total proof of an arbitrary query exists without first deriving `False`, so any rollout that claims `True` receives reward -1. This preserves the conservative reasoning semantics of Section 3 at the level of the Lean kernel itself.

A.21 BASE EXAMPLE: COMPLEX DILEMMA REASONING STRUCTURE

We begin with a comprehensive example that establishes the core reasoning pattern, drawing on a structure analogous to the *Paradox of the Court*[1], a classic logical dilemma where multiple possible paths lead to the same conclusion, despite their apparent differences.

- **Facts:**
  - Anne is green or blue
- **Rules:**
  - Rule 1: If someone is green then they are cold. $\forall x\,(\text{Green}(x) \rightarrow \text{Cold}(x))$
  - Rule 2: If someone is blue then they are cold. $\forall x\,(\text{Blue}(x) \rightarrow \text{Cold}(x))$
  - Rule 3: If someone is cold then they are rough. $\forall x\,(\text{Cold}(x) \rightarrow \text{Rough}(x))$
  - Rule 4: If someone is rough then they are young. $\forall x\,(\text{Rough}(x) \rightarrow \text{Young}(x))$
  - Rule 5: If someone is young then they are cold. $\forall x\,(\text{Young}(x) \rightarrow \text{Cold}(x))$
  - Rule 6: If someone is young then they are nice. $\forall x\,(\text{Young}(x) \rightarrow \text{Nice}(x))$
- **Questions:**
  - Q1: Anne is cold. True/False? [Answer: T]
  - Q2: Anne is rough. True/False? [Answer: T]
  - Q3: Anne is young. True/False? [Answer: T]
  - Q4: Anne is nice. True/False? [Answer: T]

The fact “Anne is green or blue" combined with Rules 1 and 2 creates a classic dilemma: both possibilities lead to the same conclusion. This dilemma reasoning yields “Anne is cold." Rule 3 then derives “Anne is rough" from cold, Rule 4 derives “Anne is young" from rough, and Rule 6 derives “Anne is nice" from young. Rule 5 creates a circular reinforcement but doesn’t alter the conclusions.

The logical structure can be represented as:

$$(G_a \vee B_a) \wedge (G_a \rightarrow C_a) \wedge (B_a \rightarrow C_a) \vdash C_a$$

- G_a: Green(Anne)
- B_a: Blue(Anne)
- C_a: Cold(Anne)

A.22 VARIATION 1: RULE REDUCTION WITH SAME CONCLUSIONS

This variation demonstrates that removing redundant rules preserves the AI’s ability to reach the same conclusions, illustrating that fewer rules can be equally effective.

- **Facts:**
  - Anne is green or blue
- **Rules:**
  - Rule 1: If someone is green then they are cold. $\forall x\,(\text{Green}(x) \rightarrow \text{Cold}(x))$
  - Rule 2: If someone is blue then they are cold. $\forall x\,(\text{Blue}(x) \rightarrow \text{Cold}(x))$
  - Rule 3: If someone is cold then they are rough. $\forall x\,(\text{Cold}(x) \rightarrow \text{Rough}(x))$
  - Rule 4: If someone is rough then they are young. $\forall x\,(\text{Rough}(x) \rightarrow \text{Young}(x))$
  - Rule 6: If someone is young then they are nice. $\forall x\,(\text{Young}(x) \rightarrow \text{Nice}(x))$
- **Questions:**
  - Q1: Anne is cold. True/False? [Answer: T]

[1]The Paradox of the Court involves a contract between the teacher Protagoras and his student Euathlus, where the student only pays for lessons if he wins a court case. When Protagoras sues Euathlus for the fee, a paradox arises: if Euathlus wins, he owes nothing, but if he loses, he still avoids payment, creating a logical contradiction about the outcome.

  - Q2: Anne is rough. True/False? [Answer: T]
  - Q3: Anne is young. True/False? [Answer: T]
  - Q4: Anne is nice. True/False? [Answer: T]

The reasoning proceeds identically to the base case: the dilemma from Rules 1-2 yields "Anne is cold," Rule 3 yields rough, Rule 4 yields young, and Rule 6 yields nice. The removal of Rule 5 has no impact on the conclusions, demonstrating its redundancy.

Key Insight: AI systems that recognize this redundancy can simplify their reasoning processes without sacrificing accuracy, embodying the "less is more" principle.

The simplified logical structure becomes:

$$(G_a \vee B_a) \wedge (G_a \to C_a) \wedge (B_a \to C_a) \vdash C_a$$

$$C_a \wedge (C_a \to R_a) \vdash R_a$$

$$R_a \wedge (R_a \to Y_a) \vdash Y_a$$

$$Y_a \wedge (Y_a \to N_a) \vdash N_a$$

- $G_a = \text{Green}(Anne)$
- $B_a = \text{Blue}(Anne)$
- $C_a = \text{Cold}(Anne)$
- $R_a = \text{Rough}(Anne)$
- $Y_a = \text{Young}(Anne)$
- $N_a = \text{Nice}(Anne)$

A.23 Variation 2: Rule Equivalence with Different Conclusions

This variation replaces multiple rules with logically equivalent fewer rules, but interestingly leads to different conclusions due to the modified rule interactions.

- **Facts:**
  - Anne is green or blue
- **Rules:** (*This variant deliberately combines two perturbations: a logical-equivalence rewrite of Rules 1+2 into a single Rule A,* ***and*** *the deletion of Rules 5 and 6. Rule 6 is essential for deriving "nice"; we discuss the two perturbations separately below.*)
  - Rule A: If someone is green or blue then they are cold. $\forall x\,((\text{Green}(x) \vee \text{Blue}(x)) \to \text{Cold}(x))$ *(replaces Rules 1+2)*
  - Rule 3: If someone is cold then they are rough. $\forall x\,(\text{Cold}(x) \to \text{Rough}(x))$
  - Rule 4: If someone is rough then they are young. $\forall x\,(\text{Rough}(x) \to \text{Young}(x))$
  - *Rule 5 (Young → Cold) – removed (redundant cycle)*
  - *Rule 6 (Young → Nice) – removed (essential for "nice"; this is the cause of Q4=F, not the Rule A rewrite)*
- **Questions:**
  - Q1: Anne is cold. True/False? [Answer: T]
  - Q2: Anne is rough. True/False? [Answer: T]
  - Q3: Anne is young. True/False? [Answer: T]
  - Q4: Anne is nice. True/False? [Answer: F]

Rule A directly captures the dilemma of Rules 1 and 2, leading to “Anne is cold" with equivalent logical force—this is the equivalence rewrite, and it preserves Q1–Q3. The change in Q4 (“nice” becomes F) is caused entirely by the *deletion of Rule 6*, not by the equivalence rewrite. We made both edits in this single illustrative variant to keep the example compact, but they are structurally independent: Variant 4 in the main benchmark applies the rewrite alone, and Variant 2 applies the essential-rule deletion alone.

Key Insight: Logical equivalence of a sub-rule set (here Rules 1+2 vs. Rule A) is a local property; it does not by itself change derivability. Macro-level derivability changes are caused by which essential rules remain, as isolated cleanly in Variant 2.

The logical equivalence can be shown as:

$$(\forall x\,(G_x \to C_x) \land \forall x\,(B_x \to C_x)) \equiv \forall x\,((G_x \lor B_x) \to C_x)$$

- $G_x = \mathrm{Green}(x)$
- $B_x = \mathrm{Blue}(x)$
- $C_x = \mathrm{Cold}(x)$

However, the missing Rule 6 prevents the derivation of Nice(Anne), showing that local equivalence doesn’t preserve global derivability.

A.24 Variation 3: Rule Interference with Contradictory Conclusions

This variation adds distracting and potentially contradictory rules, testing the AI’s ability to address conflicts and maintain coherent reasoning.

- **Facts:**
  - Anne is green or blue
  - Anne is not cold or not nice
- **Rules:**
  - Rule 1: If someone is green then they are cold. $\forall x\,(\mathrm{Green}(x) \to \mathrm{Cold}(x))$
  - Rule 2: If someone is blue then they are cold. $\forall x\,(\mathrm{Blue}(x) \to \mathrm{Cold}(x))$
  - Rule 3: If someone is cold then they are rough. $\forall x\,(\mathrm{Cold}(x) \to \mathrm{Rough}(x))$
  - Rule 4: If someone is rough then they are young. $\forall x\,(\mathrm{Rough}(x) \to \mathrm{Young}(x))$
  - Rule 5: If someone is young then they are cold. $\forall x\,(\mathrm{Young}(x) \to \mathrm{Cold}(x))$
  - Rule 6: If someone is young then they are nice. $\forall x\,(\mathrm{Young}(x) \to \mathrm{Nice}(x))$
- **Questions:**
  - Q1: Anne is cold. True/False? [Answer: F]
  - Q2: Anne is rough. True/False? [Answer: F]
  - Q3: Anne is young. True/False? [Answer: F]
  - Q4: Anne is nice. True/False? [Answer: F]

A.25 Reasoning Process and Analysis

This variation introduces *contradictory interference*, which challenges the AI’s ability to address conflicts within the logical structure. The fact “Anne is green or blue," combined with the original chain of reasoning (Rules 1-6), leads to the conclusions that Anne is cold, rough, young, and nice. However, the fact “Anne is not cold or not nice" introduces a conflict, as it implies that being nice would mean Anne is not cold. This creates a contradiction that different AI systems might address differently:

Contradiction Handling Strategy. In the experiments presented in this paper we adopt a *conservative* contradiction-handling strategy. Once a contradiction is detected in the premise set, the reasoning process halts and no further deductions are performed. As a result, all queries associated with that instance are labeled `False`.

Variations	Cold	Rough	Young	Nice
Base Example	T	T	T	T
Rule Reduction	T	T	T	T
Rule Equivalence	T	T	T	F
Rule Interference	F	F	F	F

Table 14: Comparison of conclusions across variations

Split	BERT		Qwen2		TinyLlama	
	Acc	Δ	Acc	Δ	Acc	Δ
base	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant1	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant2	0.2950	-0.7050	0.2500	-0.7500	0.2500	-0.7500
variant3	0.0000	-1.0000	0.0000	-1.0000	0.0000	-1.0000
variant4-contrapositive	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant4-double-negation	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant4-implication	1.0000	0.0000	0.9525	-0.0475	1.0000	0.0000
variant4-de-morgan	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant4-identity	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
variant4-commutativity	0.9925	-0.0075	1.0000	0.0000	1.0000	0.0000
variant4-multi (2–5 laws)	1.0000	0.0000	0.6450	-0.3550	0.9925	-0.0075

Table 15: Accuracy and deviation from base (Δ) for all models across structural variants.

Alternative reasoning strategies such as priority-based resolution or paraconsistent reasoning are possible, but they are outside the scope of the current work and are left for future investigation.

- **Conservative approach**: Detect contradiction and withhold conclusions
- **Priority-based approach**: Apply rule priorities or specificity heuristics
- **Paraconsistent approach**: Accept some contradictions and continue reasoning

In this case, a conservative reasoning system would recognize the contradiction and potentially reject all derived conclusions, resulting in *false* for all questions.

The addition of interfering rules not only tests the AI's ability to ignore distractions but also its capacity for contradiction detection and resolution.

The contradiction can be formally represented as:

$$(C_a \wedge N_a) \wedge (N_a \rightarrow \neg C_a) \vdash \bot$$

- $C_a = \mathrm{Cold}(Anne)$
- $N_a = \mathrm{Nice}(Anne)$
- $\bot$ represents a value that is always false.

A.26 Comparative Analysis

Table 14 provides a clear comparison of how different modifications to the rule set lead to distinct conclusion patterns, highlighting the sensitivity of reasoning systems to structural changes. These variations demonstrate how even minor adjustments to the logical framework can significantly impact the reasoning process and the final outcomes.

Table 15 reports accuracy (Acc) and deviation from the base condition ($\Delta = \mathrm{Acc}_{\text{variant}} - \mathrm{Acc}_{\text{base}}$) for BERT, Qwen2, and TinyLlama across all structural variants. All models achieve Acc = 1.0000 on the base split and exhibit no degradation under redundant rule removal (Variant 1; $\Delta = 0$). By contrast, removing an essential rule (Variant 2) yields a substantial drop (BERT: 0.2950, $\Delta = -0.7050$; Qwen2/TinyLlama: 0.2500, $\Delta = -0.7500$), indicating strong sensitivity to missing inferential links. Injecting explicit contradictory facts (Variant 3) reduces accuracy to 0.0000 for all models ($\Delta = -1.0000$), suggesting that the models do not reliably revise conclusions in the presence of inconsistency.