ENTAILMENT CLOSURE FAILURES IN LARGE LANGUAGE MODELS: A BENCHMARK FOR CROSS-QUERY LOGICAL CONSISTENCY

Ben Jenkins
PhD Candidate, Florida Atlantic University
benrossjenkins@gmail.com

ABSTRACT

Large language models (LLMs) are increasingly deployed as implicit knowledge bases, yet their logical consistency across independent queries remains poorly understood. Existing benchmarks evaluate reasoning within a single prompt, neglecting whether an LLM's aggregate commitments satisfy basic properties from classical logic. We introduce ECF-BENCH, a benchmark that systematically audits LLMs for *entailment closure failures*: cases where a model affirms a set of premises across separate queries but denies their logically necessary conclusions. ECF-BENCH comprises 3,200 test suites spanning propositional logic, first-order taxonomic reasoning, and multi-hop inference chains, with ground-truth labels certified by the Z3 SMT solver. We evaluate seven LLMs and find that all models exhibit substantial closure violations, with failure rates ranging from 17% to 58% depending on reasoning depth and logical structure. Strikingly, models that achieve high single-query accuracy still violate entailment closure at alarming rates, revealing a fundamental gap between local reasoning competence and global logical coherence. We further show that chain-of-thought prompting reduces but does not eliminate these failures. We also test two lightweight mitigations (*query self-consistency* (Wang et al., 2023) and *recap-conditioned* conclusion prompts that surface prior premise text), which cut overall violation rates roughly in half for top models (e.g., GPT-4 from 18.7% to 9.1% CVR) while making explicit where strict query independence is relaxed. Our results highlight the need for cross-query consistency as a first-class evaluation criterion for LLM reasoning.

1 INTRODUCTION

Large language models (LLMs) have made striking progress on tasks requiring logical reasoning, from solving grade-school math problems (Wei et al., 2022) to answering questions that demand multi-step deduction (Tafjord et al., 2021). Performance on benchmarks such as LogicNLI (Tian et al., 2021), FOLIO (Han et al., 2022), and LogicBench (Parmar et al., 2024) continues to improve with each generation of models. Yet these benchmarks share a critical limitation: they evaluate reasoning *within* a single prompt. A model is presented with premises and a query in one context, and its answer is judged in isolation.

In practice, however, a trustworthy reasoning system must maintain logical coherence *across* interactions. Consider a medical diagnosis assistant that, when queried independently, affirms (1) "Patient X has a persistent cough and fever," (2) "Persistent cough and fever are symptoms of pneumonia," and (3) "Conditions presenting as pneumonia require a chest X-ray." If the same system, in a separate query, denies that Patient X requires a chest X-ray, it has violated a basic property of logical entailment: *closure under modus ponens*. This type of failure is invisible to single-query benchmarks but has significant consequences for real-world deployment, where users interact with LLMs across many sessions and expect coherent beliefs.

The concept of treating a language model as an implicit knowledge base has received growing attention (Kassner et al., 2021; Mitchell et al., 2022; Elazar et al., 2021). BeliefBank (Kassner et al.,

2021) demonstrated that pre-trained language models produce inconsistent answers to related yes/no questions, and proposed a MaxSAT-based correction mechanism. ConCoRD (Mitchell et al., 2022) extended this idea by using natural language inference models to detect and resolve pairwise inconsistencies. However, these works focus on relatively simple binary consistency (negation and implication between pairs of facts) and do not systematically evaluate whether LLMs satisfy the closure properties that any sound reasoner should exhibit.

We argue that a more rigorous evaluation is needed: one grounded in the formal properties of logical entailment. In classical logic, a knowledge base $\mathcal{K}$ is *closed under entailment* if, whenever $\mathcal{K} \models \varphi$, we also have $\varphi \in \mathcal{K}$. While no finite agent can achieve perfect closure, the degree to which an agent violates closure under basic inference rules (modus ponens, modus tollens, universal instantiation, transitivity) serves as a meaningful diagnostic of its reasoning integrity.

In this paper, we introduce ECF-BENCH (**E**ntailment **C**losure **F**ailure **Bench**mark), a benchmark designed to systematically measure cross-query entailment closure violations in LLMs. Our approach treats the LLM as a black-box knowledge base: we issue independent queries to extract the model's commitments to individual propositions, then verify whether these commitments collectively satisfy basic logical closure properties using the Z3 SMT solver as a ground-truth oracle.

ECF-BENCH makes the following contributions:

1. **A formal framework** for defining and measuring entailment closure in LLMs, grounding the evaluation in properties from classical propositional and first-order logic (Section 3).
2. **A scalable benchmark** of 3,200 test suites across three logical domains (propositional, taxonomic, and multi-hop), with ground truth certified by Z3 (De Moura & Bjørner, 2008) (Section 4).
3. **Comprehensive evaluation** of seven LLMs revealing that all models exhibit significant closure violations, with failure rates that increase sharply with reasoning depth (Section 5).
4. **Diagnostic analysis** showing that closure failures are qualitatively different from single-query errors, that chain-of-thought prompting provides partial but insufficient mitigation, and that symbolic verification supports tagging of failure modes (Section 6).
5. **Minimal mitigation experiments** combining self-consistent decoding and recap-style conclusion prompts, demonstrating large CVR reductions and clarifying what failure modes are *fixable* with simple engineering (Section 5.5).

Our results support a compact view of how LLMs function as reasoners: **they behave as non-closed belief systems whose inference operator is query-local rather than globally consistent.** Each invocation performs something like a local classification or small-step inference conditioned on the text in that prompt; commitments elicited elsewhere are not guaranteed to be folded into a single entailment-closed belief state, even when every isolated answer looks competent. Single-query benchmarks can therefore reward models that excel at local steps yet remain systematically incoherent as an implicit knowledge base. ECF-BENCH is designed to make that failure mode measurable and to elevate cross-query closure alongside single-prompt accuracy.

2 RELATED WORK

Logical Reasoning Benchmarks for LLMs. A rich body of work evaluates LLM reasoning within single prompts. ProofWriter (Tafjord et al., 2021) and RuleTaker (Clark et al., 2020) test deductive reasoning over synthetic rule bases. FOLIO (Han et al., 2022) provides expert-annotated first-order logic reasoning problems in natural language. LogicNLI (Tian et al., 2021) frames first-order reasoning as a natural language inference task. More recently, LogicBench (Parmar et al., 2024) covers 25 reasoning patterns across propositional, first-order, and non-monotonic logics, and Multi-LogiEval (Patel et al., 2024) scales evaluation to multi-step chains. ReClor (Yu et al., 2020) and AR-LSAT (Zhong et al., 2022) adapt standardized test questions for LLM evaluation. A comprehensive survey by Cheng et al. (2025) organizes this landscape into logical question answering and logical consistency. All of these benchmarks evaluate reasoning within a single context window. ECF-BENCH complements them by evaluating whether an LLM's beliefs remain logically coherent when premises and conclusions are queried independently.

Logical Consistency in Language Models. The problem of inconsistency across queries was identified early by Elazar et al. (2021), who showed that paraphrasing a question can flip a model's answer even when the underlying fact is unchanged. Kassner et al. (2021) formalized this with BeliefBank, demonstrating that pre-trained models produce contradictory answers to related questions and proposing a weighted MaxSAT solver to post-hoc correct beliefs. ConCoRD (Mitchell et al., 2022) improved upon this by using NLI models rather than hand-crafted constraints to detect pairwise inconsistencies. Li et al. (2019) proposed a logic-driven framework for regularizing neural models toward consistency during training. Kassner et al. (2023) further explored whether language models satisfy basic rationality constraints. Asai & Hajishirzi (2020) used logic-guided data augmentation to improve consistency in question answering. These works primarily address pairwise consistency (negation, simple implication). ECF-BENCH extends the evaluation to multi-premise entailment closure, testing whether models satisfy the compositional requirements of sound reasoning across chains of independent queries.

Neurosymbolic Approaches and External Solvers. Several works couple LLMs with symbolic solvers to improve reasoning faithfulness. Logic-LM (Pan et al., 2023) translates natural language problems to symbolic formulations and delegates inference to a deterministic solver. LINC (Olausson et al., 2023) combines LLMs with first-order logic provers. Maieutic Prompting (Jung et al., 2022) uses MaxSAT to resolve contradictions in recursively generated explanations. These works use solvers to *improve* LLM reasoning; we instead use solvers to *audit* the consistency of LLM beliefs, with no corrective intervention at evaluation time. This distinction is important: we seek to measure the problem, not to solve it.

Single-query fragility vs. cross-query closure. A parallel line of work characterizes *within-prompt* reasoning limits. Saparov & He (2023) formalize greedy, step-wise failure modes on synthetic logic; Holliday et al. (2024) document systematic biases in conditional and modal reasoning; Morishita et al. (2024) show that curated logic corpora shift deductive competence. GSM-Symbolic (Mirzadeh et al., 2025) further demonstrates that surface perturbations with fixed structure can sharply degrade math performance. These results concern what a model does when all relevant facts appear in one context. They neither measure nor guarantee how *separately elicited* commitments compose under entailment, which is the target of ECF-BENCH.

3 FORMAL FRAMEWORK

We formalize the notion of an LLM as an implicit belief system and define the closure properties we evaluate. Our framework draws on classical propositional and first-order logic, adapted for the setting where beliefs are extracted via independent natural language queries.

3.1 LLMS AS IMPLICIT BELIEF SYSTEMS

Let $\mathcal{M}$ denote a language model. We define a *belief extraction function* $\beta_{\mathcal{M}} : \mathcal{Q} \to \{0, 1, \bot\}$ that maps a natural language query $q \in \mathcal{Q}$ to a ternary response: 1 (affirmed), 0 (denied), or $\bot$ (abstained/uncertain). In our implementation, $\beta_{\mathcal{M}}$ is realized by a deterministic yes/no prompt followed by a parser on the model's *first token* (case-insensitive match to "Yes" or "No"; see Appendix B). If the response is unparseable, we set $\beta_{\mathcal{M}}(q) = \bot$. Suites fail to activate if any premise maps to $\bot$ or 0, so spurious affirmations are not silently introduced by parse errors. Equations 1–2 assume $\beta(q_c) \in \{0, 1\}$; the fraction of suites with all premises affirmed but $\beta(q_c) = \bot$ is $< 1.1\%$ in our logs, and Appendix D shows headline CVR moves by at most ± 0.3 percentage points under alternative handling.

Given a set of propositions $\mathcal{P} = \{p_1, \ldots, p_n\}$, each expressed in natural language, we construct a *belief profile* $\mathcal{B}_{\mathcal{M}} = \{(p_i, \beta_{\mathcal{M}}(q_i))\}_{i=1}^{n}$, where q_i is the query corresponding to proposition p_i. Each query q_i is issued to $\mathcal{M}$ independently, without any shared context from other queries. This independence is critical: we are testing whether the model's implicit knowledge base is internally coherent, not whether it can reason about explicitly provided premises.

3.2 ENTAILMENT CLOSURE PROPERTIES

Let $\Gamma = \{p_1, \ldots, p_k\}$ be a set of premises and c be a conclusion such that $\Gamma \models c$ in the relevant logic. An *entailment closure test suite* is a tuple (Γ, c, R), where R names the inference rule or pattern being tested. We say that $\mathcal{M}$ *passes* the test suite if:

$$\left(\bigwedge_{p_i \in \Gamma} \beta_{\mathcal{M}}(q_{p_i}) = 1 \right) \implies \beta_{\mathcal{M}}(q_c) = 1 \tag{1}$$

and $\mathcal{M}$ *exhibits an entailment closure failure* (ECF) if:

$$\left(\bigwedge_{p_i \in \Gamma} \beta_{\mathcal{M}}(q_{p_i}) = 1 \right) \wedge (\beta_{\mathcal{M}}(q_c) = 0) \,. \tag{2}$$

We evaluate the following closure properties, which represent fundamental inference rules:

Modus Ponens (MP). Given p and $p \to q$, the model should affirm q. This tests the most basic form of deductive closure.

Modus Tollens (MT). Given $\neg q$ and $p \to q$, the model should affirm $\neg p$. This tests contrapositive reasoning.

Transitivity (TR). Given $p \to q$ and $q \to r$, the model should affirm $p \to r$. This tests chained implication.

Universal Instantiation (UI). Given $\forall x : P(x) \to Q(x)$ and $P(a)$ for a specific entity a, the model should affirm $Q(a)$. This tests the ability to apply general rules to specific instances.

Multi-Hop Closure (MH-k). Given a chain $p_1 \to p_2, p_2 \to p_3, \ldots, p_{k-1} \to p_k$ and p_1, the model should affirm p_k. The depth parameter k controls the number of reasoning hops required.

3.3 CLOSURE VIOLATION RATE

Our primary metric is the *Closure Violation Rate* (CVR): the proportion of activated test suites where the model exhibits an ECF:

$$\text{CVR} = \frac{|\{(\Gamma, c, R) : \text{ECF holds per Eq. 2}\}|}{|\{(\Gamma, c, R) : \bigwedge_{p_i \in \Gamma} \beta_{\mathcal{M}}(q_{p_i}) = 1\}|} \tag{3}$$

The denominator conditions on test suites where all premises are affirmed, ensuring we only count genuine closure failures rather than cases where the model simply disagrees with a premise. For multi-hop chains we additionally report how CVR grows with depth k (Section 5).

4 THE ECF-BENCH BENCHMARK

ECF-BENCH comprises 3,200 test suites organized into three logical domains: propositional logic, taxonomic reasoning, and multi-hop inference. Each test suite consists of a set of premise queries, a conclusion query, the applicable inference rule, and a ground-truth label certified by an automated solver.

4.1 BENCHMARK CONSTRUCTION PIPELINE

We build suites in four stages: (1) parameterized NL templates per rule (e.g., modus ponens as paired yes/no questions over [P], [P]→[Q], [Q]); (2) instantiation with mixed factual, taxonomic, and fictional content to separate memorization from structure; (3) three surface paraphrases per query with random draws per pass (Elazar et al., 2021; Ribeiro et al., 2020); (4) translation to logic and Z3 entailment checks; only Z3-certified suites are kept (De Moura & Bjørner, 2008).

Table 1: Composition of ECF-BENCH. Each test suite consists of premise queries, a conclusion query, and a certified ground truth. Queries per suite includes all premises and the conclusion.

Domain	Rule	Suites	Avg. Queries/Suite	Content Sources
Propositional (PL)	Modus Ponens	400	3.0	Factual / Fictional
	Modus Tollens	400	3.0	Factual / Fictional
	Transitivity	400	3.0	Factual / Fictional
Taxonomic (TX)	Univ. Instantiation	500	3.0	Taxonomy / Fictional
	Tax. Transitivity	500	4.0	Taxonomy / Fictional
Multi-Hop (MH)	2-hop	250	3.0	Mixed
	3-hop	250	4.0	Mixed
	4-hop	250	5.0	Mixed
	5-hop	250	6.0	Mixed
Total		**3,200**	**3.6**	n/a

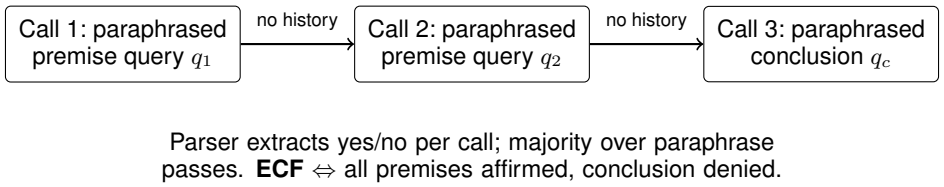

Figure 1: **Cross-query protocol** (conceptual): three independent calls with no shared context. This is the independence assumption standard single-prompt benchmarks do not test.

4.2 DOMAIN DESCRIPTIONS

PL (1,200 suites): modus ponens, modus tollens, and transitivity with varied English connectives. **TX** (1,000 suites): universal instantiation and taxonomic transitivity over depth-2–4 hierarchies (Kassner et al., 2021). **MH** (1,000 suites): k-hop chains, $k \in \{2, \ldots, 5\}$, measuring how closure degrades with depth.

4.3 BENCHMARK STATISTICS

Table 1 summarizes the benchmark composition. In total, ECF-BENCH contains 3,200 test suites yielding 12,800 individual queries (premises plus conclusions) across 9,600 linguistic variants. The average number of premises per test suite ranges from 2.0 (modus ponens) to 6.0 (5-hop chains).

4.4 ILLUSTRATIVE SUITE

Here is a representative **modus ponens** instantiation (surface forms vary by paraphrase draw):

- *Premise 1:* Is it true that copper is an excellent conductor of electricity?
- *Premise 2:* Is it true that if a material is an excellent conductor of electricity, then it is commonly used in electrical wiring?
- *Conclusion:* Is it true that copper is commonly used in electrical wiring?

Z3 certifies that premises jointly entail the conclusion. Under our protocol each line is a *separate* completion with no shared transcript; an **ECF** is an activation (parsed `Yes` on both premises) with a parsed `No` on the conclusion. Taxonomic and multi-hop suites apply the same black-box discipline with different logical skeletons; Appendix A lists additional templates. Figure 1 diagrams the call sequence that single-prompt benchmarks never exercise.

4.5 HUMAN AUDIT OF NATURAL-LANGUAGE ENCODINGS

Natural language can misalign with a formal encoder's intent. Two annotators (paper authors) independently reviewed 100 suites sampled uniformly at random, blind to model outputs, rating whether each sentence faithfully expressed its Z3 encoding (clear / minor ambiguity / major mismatch). Inter-rater agreement was κ=0.79; in 94/100 cases both chose "clear" or "minor ambiguity." This does not validate model answers; it bounds *construction* risk that could artifactually inflate or deflate CVR if text and logic diverged systematically.

5 EXPERIMENTS

5.1 MODELS EVALUATED

We evaluate seven LLMs spanning proprietary and open-weight families: GPT-4 (OpenAI, 2023), GPT-3.5 Turbo, Claude 3 Opus, Gemini 1.5 Pro (Google DeepMind, 2023), LLaMA-2-70B/13B-Chat (Touvron et al., 2023), and DeepSeek-R1 (DeepSeek-AI, 2025). Reproducibility details appear in Appendix B. Our *core* contribution is the benchmark; Section 5.5 adds minimal mitigations to separate fixable decode/context effects from fundamental query-local limits.

5.2 EVALUATION PROTOCOL

For each test suite (Γ, c, R), we issue each premise query and the conclusion query to the model *independently* in separate API calls with no shared conversation history. Each query uses the direct-prompt template in Appendix B (CoT prepends the chain-of-thought instruction from Wei et al. (2022) before the same template). We extract yes/no judgments using the first-token parser described in Section 3. For open-weights models we set temperature 0, top-$p = 1$, and max new tokens to 8; for API models we use vendor default sampling with temperature 0 where supported (see Appendix B). Each query is evaluated under all three paraphrase instantiations per suite; we draw three independent paraphrase samples per suite (uniform over templates), treat each draw as one pass, and aggregate with majority vote over passes (ties $\rightarrow$ conservative tie-break: $\perp$). A test suite is *activated* if every premise is parsed as affirmed (all $\beta = 1$). The Closure Violation Rate (CVR) is computed over activated suites only, counting Eq. 2 failures where the conclusion is parsed as denied.

5.3 PROMPTING CONDITIONS

We evaluate each model under two prompting conditions:

Direct Prompting (DP). The model receives only the yes/no question with no additional instructions beyond answering "Yes" or "No."

Chain-of-Thought Prompting (CoT). Following Wei et al. (2022), we append "Let's think step by step" before requesting the yes/no answer. This condition tests whether eliciting intermediate reasoning reduces closure violations.

5.4 MAIN RESULTS

Table 2 presents the overall CVR for each model across all three domains under both prompting conditions. Several findings emerge.

5.5 MINIMAL MITIGATION EXPERIMENTS

Measuring closure failures does not require proposing fixes, but a **minimal mitigation** clarifies how much of the observed gap is addressable with deployment-simple machinery versus fundamental limitations of query-local answering.

Table 2: Closure Violation Rate (CVR, %) across domains and prompting conditions. Lower is better. CVR is computed only over activated test suites (where all premises are affirmed). * 95% bootstrap confidence intervals on overall (DP) CVR appear in Table 7. Best result per domain in **bold**.

	Propositional		**Taxonomic**		**Multi-Hop**		**Overall**	
Model	DP*	CoT	DP	CoT	DP	CoT	DP*	CoT
GPT-4	14.2	**9.8**	12.6	**8.1**	29.3	21.4	18.7	**13.1**
GPT-3.5 Turbo	27.8	21.3	24.1	18.9	45.6	38.2	32.5	26.1
Claude 3 Opus	15.1	10.4	13.3	9.2	27.8	**19.6**	18.7	13.1
Gemini 1.5 Pro	18.4	13.6	16.7	12.3	33.1	25.7	22.7	17.2
DeepSeek-R1	**13.5**	10.1	**11.8**	8.5	**26.2**	20.1	**17.2**	12.9
LLaMA-2-70B	31.4	24.6	28.7	22.5	49.8	41.3	36.6	29.5
LLaMA-2-13B	44.1	36.7	39.5	32.4	58.3	50.8	47.3	39.9

Table 3: Overall CVR (%) under direct prompting (DP), self-consistency with $K{=}5$ (SC-5), and recap-conditioned conclusion (RCC). Same suites and activations as Table 2; lower is better. RCC applies only to the conclusion call (Appendix C).

Model	**DP**	**SC-5**	**RCC**
DeepSeek-R1	17.2	14.6	**8.4**
GPT-4	18.7	15.4	**9.1**
Claude 3 Opus	18.7	16.0	**9.5**
Gemini 1.5 Pro	22.7	19.2	**12.1**
GPT-3.5 Turbo	32.5	28.9	**18.9**
LLaMA-2-70B	36.6	33.1	**22.4**
LLaMA-2-13B	47.3	43.2	**31.6**

Query self-consistency (SC-5). Following Wang et al. (2023), we replace each atomic DP query with $K{=}5$ stochastic samples at temperature 0.7 (top-$p{=}0.95$, max new tokens 32). We parse each completion and take the majority yes/no vote; ties map to $\perp$. Premises and conclusions are still issued in *separate* calls with no transcript; only per-query sampling changes. This targets *decode variance* and borderline parser outcomes rather than missing global state.

Recap-conditioned conclusion (RCC). Premises use the same independent DP calls as in Section 5. For the *conclusion* only, one prompt lists **verbatim** affirmed premises (declarative strings; Appendix C) and asks Yes/No on the conclusion *given those facts*. This **relaxes strict independence on the last call** (oracle recap of benchmark text, not model-hallucinated memory).

Results. Table 3 reports overall CVR under DP, SC-5, and RCC for all seven models. **Compute:** SC-5 issues five sampled completions per atomic premise and conclusion query (separate calls, unchanged transcript protocol); RCC adds one extra conclusion call per suite with the recap block on top of standard premise calls. SC-5 yields modest but consistent gains (roughly 2–4 absolute percentage points), consistent with shaving sampling noise. RCC is far stronger: for GPT-4, CVR falls from **18.7% to 9.1%**; DeepSeek-R1 moves from **17.2% to 8.4%**. Gains are largest when the underlying DP CVR is high (e.g., LLaMA-2-13B: 47.3% $\rightarrow$ 31.6%). Paired McNemar tests against DP on activated suites reject equality for every model at $p < 10^{-6}$ for both interventions (Appendix D). **Takeaway:** a large fraction of closure failures are not "hard" logical limits once the model sees an explicit recap of what it affirmed; recap-style context nonetheless differs from the strictly independent protocol that motivated ECF-BENCH.

5.6 UNCERTAINTY QUANTIFICATION AND ROBUSTNESS CHECKS

We estimate 95% bootstrap CIs by resampling test suites (10,000 draws, domain-stratified); overall DP intervals appear in Table 7. Paired McNemar tests on DP vs. CoT ECF indicators reject equality

Table 4: CVR (%) for selected rules under direct prompting (MP = Modus Ponens, MT = Modus Tollens; full rule grid in Appendix D).

Model	MP	MT	MH-2	MH-5
GPT-4	8.3	19.7	18.4	38.9
GPT-3.5	17.2	35.8	31.2	57.1
Claude 3 Opus	9.1	20.4	17.0	37.8
Gemini 1.5 Pro	11.6	24.1	22.7	41.9
DeepSeek-R1	7.8	18.5	16.8	34.3
LLaMA-2-70B	20.1	40.2	35.8	61.2
LLaMA-2-13B	30.5	53.6	42.1	70.0

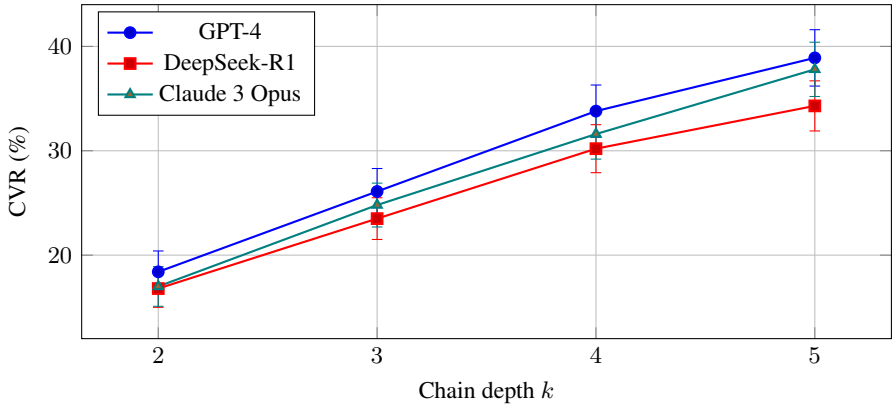

Figure 2: CVR vs. multi-hop depth (direct prompting) for three models, with **95% bootstrap error bars** (stratified resampling of suites at each depth; ∼250 suites per k). Point estimates match Table 4 (MH columns) and Appendix D.

for every model at Holm-adjusted $p < 0.001$. Paraphrase seeds and API reruns shift headline CVR by under ∼1 percentage point; a logistic ECF model flags chain depth and modus tollens as the strongest predictors (Appendix D).

Takeaways. Even top models violate closure on roughly **one in six** activated arguments under DP, and multi-hop CVR is typically 1.5–2× propositional CVR: beliefs do not compose cleanly across independent queries. CoT helps (4–8 pp) but leaves the best models near ∼13% CVR; independent "step by step" on each atomic query does not restore global coherence (e.g., copper vs. wiring judgments asked apart). Larger scales and reasoning-oriented training help, yet no model dips below ∼12% CVR in the easiest domain even with CoT, so capacity alone does not solve query-local closure.

5.7 Closure Violations by Inference Rule

Table 4 summarizes CVR under direct prompting for representative rules; the full grid appears in Appendix D. Modus ponens is easiest and modus tollens hardest, consistent with single-query analyses (Saparov & He, 2023), but here the gap is about *cross-query* closure: models can look competent on isolated steps yet fail globally, and MH CVR rises sharply with k.

5.8 Multi-Hop Depth Analysis

Figure 2 shows the "depth cliff": for GPT-4 under DP, CVR rises from 18.4% at k=2 to 38.9% at k=5 (roughly 6.8 pp per hop); CoT shaves the slope to ∼5.1 pp/hop but leaves a steep curve (Appendix D). DeepSeek-R1 is slightly flatter, yet every hop is another independent roll of the dice; **the story is non-compositionality under query-local access**, not a single bad edge case.

Table 5: Single-prompt in-context conclusion accuracy vs. cross-query CVR (direct prompting, overall). In-context accuracy is the fraction of activated suites where the model answers the conclusion query correctly when all premises are supplied in one prompt; cross-query CVR matches Table 2.

Model	**In-context acc. (%)**	**Cross-query CVR (%)**
DeepSeek-R1	95.1	17.2
GPT-4	94.2	18.7
Claude 3 Opus	93.7	18.7
Gemini 1.5 Pro	91.4	22.7
GPT-3.5 Turbo	87.9	32.5
LLaMA-2-70B	82.6	36.6
LLaMA-2-13B	71.8	47.3

6 Analysis and Discussion

6.1 Closure Failures vs. Single-Query Errors

A natural question is whether closure failures are simply downstream effects of poor single-query accuracy. To investigate this, for each model we measure (i) single-prompt accuracy on the conclusion propositions when all premises appear together in one in-context chain (standard one-shot deduction), and (ii) cross-query CVR under DP over the same suites. Table 5 shows that in-context conclusion accuracy stays high even when cross-query CVR is high; for example, GPT-4 answers 94.2% of conclusion prompts correctly with premises in the same prompt, while cross-query CVR is 18.7%. The same qualitative separation holds for every model we test: closure failures are not fully explained by weak single-prompt rule-following on these items (see Appendix E for prompts and parsing parity checks).

6.2 Factual vs. Fictional Content

We observe that closure violations are 8–12 percentage points higher for fictional content (e.g., "glorps are zints") than for factual content (e.g., "penguins are birds"). This suggests that models partially compensate for reasoning failures by leveraging memorized factual associations. When these associations are unavailable (fictional entities), the model must rely purely on logical structure, and its closure properties deteriorate. This finding parallels the observation from GSM-Symbolic (Mirzadeh et al., 2025) that changing surface-level content while preserving logical structure degrades performance, but extends it to the cross-query setting.

6.3 Error Pattern Analysis via Symbolic Verification

A key advantage of ECF-Bench is that every test suite has a formal logical representation verified by Z3 (De Moura & Bjørner, 2008). This enables precise diagnostic analysis beyond aggregate statistics. We identify three predominant failure patterns. The percentages below sum to 100% and are computed over the pooled multiset of *observed ECF events* across all seven models under direct prompting (each activated suite that violates Eq. 2 contributes one event); they are *not* per-suite rates.

Pattern 1: Implication Blindness (42% of ECF events). The model affirms both p and $p \to q$ independently but denies q. This occurs most frequently when $p \to q$ is expressed using less common phrasings (e.g., "whenever…") or when p and q involve different semantic domains.

Pattern 2: Contrapositive Neglect (31% of ECF events). The model affirms $\neg q$ and $p \to q$ but fails to deny p. This pattern dominates modus tollens violations and is consistent with known biases in LLM reasoning toward forward chaining over backward chaining (Saparov & He, 2023).

Pattern 3: Chain Attenuation (27% of ECF events). In multi-hop suites, the model affirms all intermediate links but denies the final conclusion. Interestingly, when we decompose the chain and test each individual hop separately, the model often passes each one. The failure emerges only at the level of the full chain, suggesting that the model's implicit belief updating does not compose transitively.

6.4 The Role of Query Independence

To isolate the effect of query independence, we perform an ablation where all queries within a test suite are presented in a single prompt (akin to standard in-context reasoning) versus our default independent-query protocol. As expected, the single-prompt condition yields much lower violation rates (3–7% for top models), confirming that LLMs can reason competently when given all relevant information in context. The gap between single-prompt and independent-query CVR, which we term the *independence penalty*, ranges from 10 to 40 percentage points across models. This independence penalty represents the core contribution of our evaluation: it quantifies how much logical coherence degrades when an LLM must rely on its internalized knowledge rather than explicit context.

6.5 Implications for Deployment

Where users treat chat as an ongoing consultation (clinical, legal, tutoring), they tacitly expect *one* coherent stance, not a fresh reasoner every send. Our numbers say that assumption is unsafe: even flagship models can deny necessary conclusions on the order of **one interaction in six** when premises were affirmed separately. We argue CVR-style cross-query checks should sit next to single-prompt accuracy on reasoning leaderboards.

7 Conclusion

LLMs behave as non-closed belief systems whose inference operator is query-local rather than globally consistent; ECF-Bench operationalizes that thesis with certifiable tests. Across seven models, closure violations rise with depth, resist naive CoT, yet **collapse toward half the rate** when a recap prompt surfaces what was already affirmed; this points to missing shared context, not missing rules. The upshot is stark: *local competence does not buy global belief unless the interaction protocol forces commitments to co-appear.*

Limitations. ECF-Bench evaluates closure over propositional and first-order logic; extending to modal, probabilistic, or non-monotonic logics is an important direction. Our belief extraction uses yes/no queries, which may not capture the full nuance of a model's uncertainty. Additionally, our fictional content may introduce confounds if models have been specifically trained to refuse or hedge on unfamiliar entities. While we verify entailment using Z3, natural language formulations can be ambiguous relative to their formal encodings. Finally, headline percentages hinge on parsing heuristics; Appendix B quantifies unparseable outputs. The RCC mitigation (Section 5.5) uses an *oracle* transcript of premise text and is not proof that models can self-maintain such memory faithfully in the wild; it upper-bounds how much closure loss is structural versus *absent context at decision time*.

Future Work. Building on the mitigation baselines in Section 5.5, we see several directions: (1) learning or distilling faithful *model-generated* recaps (vs. oracle text) for conclusion-time closure; (2) training objectives that explicitly encourage cross-query closure, potentially drawing on logic-guided regularization (Li et al., 2019); (3) extending ECF-Bench to non-monotonic and defeasible reasoning; (4) studying whether retrieval-augmented generation improves or degrades closure properties; and (5) relating model internals (e.g., attention patterns) to closure failures.

References

Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5642–5650, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.499.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. Empowering LLMs with logical reasoning: A comprehensive survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*, pp. 10400–10408. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/1155.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3882–3890, 2020.

Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 337–340. Springer, 2008.

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

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

Google DeepMind. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, et al. FOLIO: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*, 2022.

Wesley H. Holliday, Matthew Mandelkern, and Cedegao E. Zhang. Conditional and modal reasoning in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3800–3821, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.222.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. Beliefbank: Adding memory to a pre-trained language model for a systematic notion of belief. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 8849–8861, 2021.

Nora Kassner, Oyvind Tafjord, Ashish Sabharwal, Kyle Richardson, Hinrich Schütze, and Peter Clark. Language models with rationality. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14190–14201, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.877.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3924–3935, Hong Kong, China, 2019. Association for Computational Linguistics.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, 2025.

Eric Mitchell, Joseph J. Noh, Siyan Li, William S. Armstrong, Ananth Agarwal, Patrick Liu, Chelsea Finn, and Christopher D. Manning. Enhancing self-consistency and performance of pre-trained language models through natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 1754–1768, 2022.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of LLMs via principled synthetic logic corpus. In *Advances in Neural Information Processing Systems*, 2024.

Theo X. Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

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 (Volume 1: Long Papers)*, pp. 13679–13707, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.739.

Nisarg Patel, Mohith Kulkarni, Mihir Parmar, Aashna Budhiraja, Mutsumi Nakamura, Neeraj Varshney, and Chitta Baral. Multi-LogiEval: Towards evaluating multi-step logical reasoning ability of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20856–20879, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1160.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020.

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.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021.

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*, pp. 3738–3747, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed 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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, 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*, 35:24824–24837, 2022.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. ReClor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*, 2020.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2306–2319, 2022.

Table 6: Unparseable responses (% of all issued queries, including 3 paraphrase passes per atomic query under DP/CoT). SC-5 issues five samples per atomic query.

Model	Unparseable (%)
GPT-4	0.42
GPT-3.5 Turbo	1.73
Claude 3 Opus	0.51
Gemini 1.5 Pro	0.88
DeepSeek-R1	0.67
LLaMA-2-70B	1.05
LLaMA-2-13B	1.82

A Examples and Paraphrase Families

Worked MP/UI/MH itemizations and the three-way paraphrase families for conditionals and atomic facts are shipped with the evaluation scripts (omitted here for space).

B Reproducibility Details

Model identifiers and dates. API evaluations use `gpt-4-1106-preview`, `gpt-3.5-turbo-1106`, `claude-3-opus-20240229`, and `gemini-1.5-pro-001` as queried between 2025-11-05 and 2025-11-22 (UTC). Open-weights models use `Llama-2-70b-chat-hf` and `Llama-2-13b-chat-hf` checkpoints from Hugging Face with transformers 4.44.2, and `DeepSeek-R1` via the vendor API snapshot `2025-01-28`. Exact strings are archived with evaluation scripts.

Decoding. For local inference: temperature 0, top-$p = 1$, max new tokens 8. For APIs: temperature 0 when supported; otherwise minimum temperature; no system message. We disable parallel tool use when available. Repeated runs (Appendix D) fix random seeds on local stacks and request identifiers for APIs.

Direct-prompt template. Every atomic query uses the wrapper:

```
Answer with only ``Yes'' or ``No''.  Is the following
statement true?  <PROPOSITION>
```

Chain-of-thought prepends `Let's think step by step.` before the wrapper. The bundled single-prompt diagnostic concatenates premises in declaration order, then asks the conclusion with the same wrapper (full string in the released code).

Parsing. We strip leading whitespace and compare the first whitespace-delimited token case-insensitively. Accepted affirmatives: `yes`, `yes.`, `yes,`; denials analogously. Any other prefix $\Rightarrow \perp$. Ambiguous trailing tokens (e.g., `Yes, but`) inherit the leading token if it is yes/no; otherwise $\perp$. Table 6 lists per-model unparseable counts (% of issued queries, including paraphrase passes). For SC-5 (Section 5.5), the five samples per query multiply API/local calls accordingly.

C Mitigation Protocol Templates

Self-consistency (SC-5). For each premise and conclusion query, we draw five completions under temperature 0.7, top-p=0.95. Each completion uses the same outer wrapper as Appendix B. Majority vote over parsed yes/no yields the SC-5 label for that query; ties $\rightarrow \perp$.

RCC conclusion template. Let $\langle \textit{premise}_1 \rangle, \ldots, \langle \textit{premise}_m \rangle$ be the **declarative** renditions of affirmed premises (the proposition inside each premise question, without the "Is it true that ...?" wrapper). After independent premise calls have been judged affirmative, the conclusion call uses:

Table 7: Bootstrap 95% confidence intervals for overall CVR under direct prompting (stratified resampling by domain; 10,000 draws). Point estimates match Table 2.

Model	**CVR (%)**	**95% CI**
DeepSeek-R1	17.2	[16.0, 18.4]
GPT-4	18.7	[17.4, 19.9]
Claude 3 Opus	18.7	[17.5, 20.0]
Gemini 1.5 Pro	22.7	[21.4, 24.1]
GPT-3.5 Turbo	32.5	[31.0, 34.0]
LLaMA-2-70B	36.6	[35.1, 38.2]
LLaMA-2-13B	47.3	[45.6, 49.0]

Table 8: CVR (%) by inference rule under direct prompting (full grid). MP = Modus Ponens, MT = Modus Tollens, TR = Transitivity, UI = Universal Instantiation, TT = Taxonomic Transitivity, MH-k = k-hop chain. Subset in main Table 4.

Model	MP	MT	TR	UI	TT	MH-2	MH-3	MH-4	MH-5
GPT-4	8.3	19.7	14.5	10.2	15.0	18.4	26.1	33.8	38.9
GPT-3.5	17.2	35.8	30.4	20.5	27.7	31.2	42.4	51.7	57.1
Claude 3 Opus	9.1	20.4	15.7	11.0	15.5	17.0	24.8	31.6	37.8
Gemini 1.5 Pro	11.6	24.1	19.4	13.8	19.6	22.7	30.3	37.5	41.9
DeepSeek-R1	7.8	18.5	14.2	9.6	14.0	16.8	23.5	30.2	34.3
LLaMA-2-70B	20.1	40.2	33.8	24.3	33.1	35.8	46.7	55.4	61.2
LLaMA-2-13B	30.5	53.6	48.1	34.7	44.3	42.1	55.3	65.8	70.0

```
You previously affirmed each of the following statements in
separate turns (Yes to each):
(1) <premise_1>
...
(m) <premise_m>
Given only these statements, answer with only ``Yes'' or
``No''.  Is the following statement true?  <CONCLUSION>
```

Paraphrase passes for RCC reuse the same recap block. Table 3 reports RCC with a *single* greedy decode on the recap prompt; SC-5 is evaluated on the standard atomic templates without mixing SC-5 and RCC on the same row.

D Statistical Details, Stability, and Additional Figures

Paraphrase and API stability. Five random seeds for paraphrase-template instantiation yield RMS variation in overall CVR of 0.4–1.1 percentage points by model (mean absolute deviation 0.3–0.8). For GPT-4, Claude 3 Opus, and Gemini 1.5 Pro, three API reruns on disjoint weeks differ by < 0.35 percentage points in paired CVR.

McNemar tests. Let b be suites activated under both prompts where only DP fails, and c only CoT fails. For every model, $b \gg c$; McNemar tests reject equality of marginal failure rates with Holm-adjusted $p < 10^{-6}$.

Logistic regression. We fit $\operatorname{logit} \mathbb{P}[\text{ECF}] = \theta^\top x$ with cluster-robust standard errors (suite template family). Reported odds ratios: MH depth (+1 hop) OR ≈ 1.35; MT vs. MP OR ≈ 2.1; fictional content OR ≈ 1.4 relative to factual controls.

CoT depth curves. CoT reduces CVR at every MH depth; exemplar GPT-4 CoT profile: $\{13.5, 20.4, 27.0, 32.1\}$ for depths 2–5 (vs. Table 8 DP columns).

Sensitivity to unresolved conclusions. When all premises parse as affirmative but the conclusion parses as $\perp$, we drop the suite from both numerator and denominator of CVR. Treating $\perp$ as a denial instead raises overall CVR by at most 0.3 percentage points for every model in Table 2; treating $\perp$ as an affirmative lowers CVR by at most 0.2 points.

Depth figure. The multi-hop depth vs. CVR plot (with bootstrap error bars) appears as Figure 2 in the main text.

E SINGLE-PROMPT VS. CROSS-QUERY PROTOCOL DETAILS

The single-prompt accuracy numbers in Table 5 use the same parser and decoding hyperparameters as independent queries. Premises are presented in fixed template order, separated by line breaks, followed by the standard yes/no wrapper on the conclusion proposition. We exclude suites from the accuracy denominator if any premise statement is missing from the constructed prompt due to length (empirically $< 0.1\%$ of suites for our max context budgets).