ConsistPRM: Evaluating and Improving Logical Consistency in LLM Reasoning via Transformation-Invariant Probes

Anonymous authors
Paper under double-blind review

Abstract

We argue that accuracy on isolated logic problems is an incomplete measure of reasoning ability, and propose evaluating logical *consistency* as invariance under truth-preserving transformations. We introduce ConsistBench, a diagnostic suite of 162 base instances with 573 Z3-verified probes spanning entailment, negation, contrapositive, and transitivity transformations. Evaluating six frontier and open-weight models reveals a striking *negation consistency gap*: models achieving 82–95% base accuracy score only 5–14% on negation probes, a failure pattern reproduced under both commutation consistency and probe correctness metrics. We taxonomize three negation failure modes and show, through a pilot ConsistPRM study, that formal Z3-verified step labels substantially outperform heuristic and LLM-judge labels for training consistency-aware process reward models, identifying label quality as the key bottleneck for progress.

1 Introduction

Large language models have demonstrated strong performance on standard logic benchmarks (Saparov & He, 2023; Tafjord et al., 2021), yet isolated accuracy is an incomplete measure of logical competence. A model that correctly answers "If A then B; A holds; does B hold?" but fails on the logically equivalent contrapositive formulation does not reason in a manner consistent with the underlying logic. Such inconsistencies undermine trust in model outputs and limit their reliability in applications where logical soundness, not merely surface-level accuracy, is required.

Behavioral testing methodologies such as CheckList (Ribeiro et al., 2020) have shown that evaluating invariance under systematic perturbations reveals failure modes invisible to aggregate accuracy. Inspired by this paradigm, we propose evaluating LLM reasoning through *transformation invariance*: if a model genuinely understands the logical structure of a problem, its answers should commute with truth-preserving logical transformations.

This paper makes three contributions:

- **ConsistBench**, a diagnostic benchmark of 162 base instances and 573 Z3-verified transformation probes across four transformation families, together with a formal framework that distinguishes *commutation consistency* (self-agreement under transformation) from *probe correctness* (agreement with ground truth).
- **The negation consistency gap**, a cross-model finding in which models scoring 82–95% on base instances achieve only 5–14% on negation probes, driven by three identified failure modes: negation blindness, polarity inversion, and scope confusion.
- **ConsistPRM**, a process reward model for step-level consistency scoring, with a pilot study demonstrating that Z3-verified step labels (+12.4% consistency) substantially outperform heuristic (+2.1%) and LLM-judge labels (+5.8%), identifying formal supervision quality as the critical bottleneck.

2 CONSISTBENCH: CONSISTENCY AS TRANSFORMATION INVARIANCE

2.1 FORMAL FRAMEWORK

Let φ denote a logical reasoning instance and $M(\varphi) \in \{T, F\}$ denote a model's binary judgment. For a logical transformation τ (e.g., negating a premise), let τ^* denote the induced map on truth values. We define two complementary metrics:

Commutation Consistency (CC). A model is τ-consistent with respect to commutation if its answer to the transformed instance agrees with the transformation applied to its original answer:

$$\text{CC}(\tau) = \frac{1}{|\mathcal{D}|} \sum_{\varphi \in \mathcal{D}} \mathbf{1}[M(\tau(\varphi)) = \tau^*(M(\varphi))]\,. \tag{1}$$

This metric captures *self-agreement* regardless of correctness. A model can achieve perfect CC while being systematically wrong, as long as its errors are coherent under τ.

Probe Correctness (PC). A model achieves high probe correctness if its answer to the transformed instance matches the Z3-verified ground truth:

$$\text{PC}(\tau) = \frac{1}{|\mathcal{D}|} \sum_{\varphi \in \mathcal{D}} \mathbf{1}\left[M(\tau(\varphi)) = y^*_{\tau(\varphi)}\right], \tag{2}$$

where $y^*_{\tau(\varphi)}$ is the ground-truth answer to the transformed instance.

Relationship. When a model's base accuracy is high, the two metrics converge. Formally, for a deterministic transformation τ where $\tau^*(T) = F$ and $\tau^*(F) = T$ (as in premise negation), $\text{CC}(\tau) = \text{PC}(\tau)$ whenever $M(\varphi)$ is correct. In general, $\text{PC}(\tau) \leq \text{CC}(\tau) + (1 - \text{Acc}_{\text{base}})$. Under probe correctness, high base accuracy is necessary but not sufficient for strong consistency. Under commutation consistency, however, a model can be coherently wrong: internally consistent yet inaccurate. We report both metrics throughout, as they capture complementary aspects of reasoning quality.

2.2 BENCHMARK CONSTRUCTION AND Z3 VERIFICATION

ConsistBench is constructed through a three-stage pipeline. First, we curate 162 base instances from propositional and first-order logic, covering conditional reasoning, syllogisms, and quantified statements. Each instance is formalized as a Z3 (de Moura & Bjørner, 2008) satisfiability query, ensuring machine-verifiable ground truth. Second, for each base instance, we apply one or more transformations from four families (Section 2.3) to produce 573 probe instances. Third, every probe is independently verified through Z3 to confirm that its ground-truth label is correct.

We position ConsistBench as a *diagnostic suite* rather than a comprehensive benchmark. Its 162 base instances are designed to isolate specific consistency failure modes with high signal-to-noise ratio, rather than to provide exhaustive coverage of logical reasoning. Future expansions will include double negation invariance, De Morgan variants, premise reordering and paraphrase invariance, and quantifier-negation interaction cases.

2.3 TRANSFORMATION FAMILIES

We implement four transformation families, each with a well-defined induced map τ^* on truth values:

Entailment. Given a valid argument φ, we construct a probe $\tau(\varphi)$ that adds a logically entailed conclusion as an additional premise. The expected answer is preserved ($\tau^* = \text{id}$). This tests whether models recognize that adding redundant valid information should not change the conclusion.

Negation. We negate a key premise in φ, producing $\tau(\varphi)$. For instances where the original conclusion depends on that premise, the expected answer flips ($\tau^*(T) = F$, $\tau^*(F) = T$). This tests sensitivity to polarity changes in the reasoning chain.

Contrapositive. We replace a conditional "if P then Q" with its contrapositive "if $\neg Q$ then $\neg P$". Since these are logically equivalent, $\tau^* = \text{id}$. This tests whether models treat logically equivalent reformulations identically.

Table 1: Base accuracy, probe correctness (PC), and commutation consistency (CC) across transformation families. The negation gap is consistent across all models and both metrics. CC is reported for the top three models; — indicates CC was not computed.

		Probe Correctness (PC)				Commutation Consistency (CC)			
Model	**Base**	**Ent.**	**Neg.**	**Contra.**	**Trans.**	**Ent.**	**Neg.**	**Contra.**	**Trans.**
GPT-4o	0.95	0.91	0.12	0.78	0.88	0.93	0.15	0.80	0.90
Claude 3.5 Sonnet	0.94	0.89	0.14	0.76	0.85	0.91	0.17	0.78	0.87
Llama-3.1-70B	0.91	0.84	0.08	0.69	0.79	0.87	0.11	0.72	0.81
Qwen2.5-72B	0.92	0.86	0.11	0.72	0.82	—	—	—	—
Mistral-Large-2	0.89	0.82	0.09	0.65	0.76	—	—	—	—
Llama-3.1-8B	0.82	0.71	0.05	0.52	0.63	—	—	—	—

Transitivity. Given premises $A \to B$ and $B \to C$, we construct a probe asking about $A \to C$ directly. Since the transitive closure preserves validity, $\tau^* = \text{id}$ for valid chains. This tests whether models maintain coherence across multi-hop inferences.

3 Experiments

3.1 Cross-Model Consistency Evaluation

We evaluate six models: GPT-4o, Claude 3.5 Sonnet, Llama-3.1-70B, Qwen2.5-72B, Mistral-Large-2, and Llama-3.1-8B. All models are prompted with zero-shot chain-of-thought (Wei et al., 2022) and asked to respond with a final "True" or "False" judgment. We extract answers by parsing the last line of the model response for a clear Boolean token. Instances where no answer can be extracted are marked incorrect. Table 1 reports both probe correctness (PC) and commutation consistency (CC).

Several patterns emerge. First, all models exhibit a dramatic gap between base accuracy and negation probe correctness. GPT-4o scores 0.95 on base instances but only 0.12 on negation probes. Second, CC and PC track closely for the three highest-accuracy models (difference ≤ 0.03 across transformations), confirming that when base accuracy is high, the two metrics are nearly interchangeable. Third, the consistency hierarchy across transformations is stable: entailment $>$ transitivity $>$ contrapositive $\gg$ negation, suggesting that the difficulty ordering is intrinsic to the transformation type rather than model-specific.

3.2 The Negation Consistency Gap

The most striking finding is the near-total failure on negation probes. Even GPT-4o, which achieves 0.95 base accuracy, drops to 0.12 on negation PC and 0.15 on negation CC. This gap is not attributable to a single model family or architecture: it persists across proprietary (GPT-4o, Claude 3.5) and open-weight (Llama-3.1, Qwen2.5, Mistral) models, and scales with model size (Llama-3.1-8B at 0.05 vs. 70B at 0.08). The universality of this failure suggests a fundamental limitation in how current LLMs process negation during multi-step reasoning, consistent with earlier findings on negation sensitivity in masked language models (Kassner & Schütze, 2020; Hosseini et al., 2021).

3.3 Negation Sanity Checks

Because the near-zero negation scores are extreme, we conducted several checks to rule out evaluation artifacts.

Worked examples. Consider a base instance: "All birds can fly. Tweety is a bird. Can Tweety fly?" (Answer: True). The negation probe negates the first premise: "Not all birds can fly. Tweety is a bird. Can Tweety fly?" (Answer: False). In Z3, the base is encoded as ; the probe replaces this with , and the solver confirms that Tweety's ability to fly is no longer guaranteed. A second example: "If it rains, the ground is wet. It is raining. Is the ground wet?" (True). Negation probe: "If it rains, the ground is wet. It is *not* raining. Is the ground wet?" (False, since the consequent is not independently established). GPT-4o answers "True" to both, exemplifying negation blindness.

Answer distribution. Base instances have a balanced label distribution: 54% True and 46% False. Negation probes correctly invert this to 46% True and 54% False. However, model predictions on negation probes are heavily skewed: GPT-4o predicts True for 89% of negation probes and False for only 11%. This confirms that models default to affirming the original conclusion regardless of the negated premise.

Answer extraction. We use strict regular-expression matching on the final response line, accepting only unambiguous "True" or "False" tokens. Manual inspection of 50 randomly sampled responses confirmed 100% extraction accuracy; no instances were misclassified due to parsing errors.

Hand audit. Two annotators independently verified a random sample of 40 negation probes (base-probe pairs) against their Z3 encodings. Inter-annotator agreement was 97.5% (39/40), and the single disagreement was resolved as a correct Z3 label upon review. We estimate the Z3 verification error rate at $< 2.5\%$.

3.4 Error Taxonomy

We manually analyzed 100 GPT-4o failures on negation probes and identified three systematic failure modes:

Negation Blindness (47%). The model generates a reasoning chain that entirely ignores the negated premise, proceeding as if the original (non-negated) premise were stated. The chain-of-thought may explicitly quote the premise but omit the negation word. This is the dominant failure mode and is consistent with findings that chain-of-thought explanations can be unfaithful to the model's actual reasoning process (Turpin et al., 2024).

Polarity Inversion (31%). The model correctly identifies that a premise has been negated and flips its final answer, but does not update the intermediate reasoning steps. The reasoning chain derives the same intermediate conclusions as the base instance, then appends "but since the premise is negated, the answer is False" without verifying whether the flipped conclusion is logically warranted.

Scope Confusion (22%). The model misapplies the scope of negation, particularly in instances involving quantifiers. When "All X are Y" is negated to "Not all X are Y" (i.e., $\exists x : X(x) \wedge \neg Y(x)$), the model interprets the negation as "No X are Y" ($\forall x : X(x) \rightarrow \neg Y(x)$), applying negation at the wrong scope level.

4 ConsistPRM: Process Reward Models for Consistency

4.1 Approach

Process reward models (PRMs) assign scores to individual reasoning steps rather than only to final answers (Lightman et al., 2024; Uesato et al., 2022), enabling fine-grained identification of where reasoning goes wrong. We adapt this framework to consistency evaluation by training a PRM to score whether each step in a chain-of-thought correctly handles a logical transformation.

We fine-tune a Llama-3.1-8B model as the PRM backbone on 1,200 annotated chain-of-thought traces (200 per transformation type, balanced across correct and incorrect traces). Each step receives a binary label indicating whether the logical operation at that step is valid. We compare three sources of step-level labels: (1) *heuristic labels* derived from pattern matching on negation keywords and conclusion tokens; (2) *LLM-judge labels* obtained by prompting GPT-4o to assess each step; and (3) *Z3-verified labels* obtained by encoding each intermediate claim as a Z3 query and checking satisfiability.

4.2 Results: Label Quality as the Bottleneck

Table 2 reveals a clear hierarchy aligned with label formality. Heuristic labels, which rely on surface-level keyword matching, achieve only 0.23 recall on negation steps and yield a negligible +2.1% improvement in downstream negation consistency when used for best-of-8 re-ranking. LLM-judge labels improve recall to 0.41 but still miss the majority of subtle negation errors, particularly scope

Table 2: ConsistPRM performance by label source. Z3-verified labels substantially outperform alternatives, particularly on negation probe recall. Consistency Δ measures improvement on negation PC when using the PRM to re-rank candidate reasoning chains (best-of-8 sampling).

Label Source	Neg. Recall	Overall F1	Consistency Δ
Heuristic (pattern matching)	0.23	0.61	+2.1%
LLM-as-judge (GPT-4o)	0.41	0.68	+5.8%
Z3-verified step labels	**0.67**	**0.79**	**+12.4%**

confusion cases where the surface form appears reasonable. Z3-verified labels achieve 0.67 negation recall and +12.4% consistency improvement, demonstrating that formal verification provides supervision signal that neither heuristic patterns nor LLM judges can replicate.

The central message is that the bottleneck for consistency-aware training is not model capacity or training methodology, but *formal supervision quality*. Scaling up heuristic labels yields diminishing returns because the labels themselves fail to capture the logical structure of negation errors. This points to a need for scalable formal verification pipelines as a prerequisite for training robust reasoning models, a direction complementary to existing work on outcome-based (Cobbe et al., 2021) and process-based (Lightman et al., 2024) reward models.

5 Related Work

Logic and reasoning evaluation. Formal reasoning benchmarks such as ProofWriter (Tafjord et al., 2021) and the systematic analysis of Saparov & He (2023) evaluate whether models can derive conclusions from premises, while LogiQA-style benchmarks (Joshi et al., 2023) test multiple-choice logical reasoning. Pan et al. (2023) augment LLMs with symbolic solvers for faithful reasoning. These benchmarks measure accuracy on individual instances; ConsistBench complements them by evaluating whether accuracy is *stable* under logical transformations.

Consistency and robustness. Self-consistency decoding (Wang et al., 2023) improves accuracy by marginalizing over sampled reasoning paths, but measures agreement across stochastic samples rather than across logically related instances. CheckList (Ribeiro et al., 2020) pioneered behavioral testing via perturbation invariance for NLP tasks; we adapt this philosophy to logical reasoning with formally verified transformations. Kassner & Schütze (2020) demonstrated that pretrained language models are insensitive to negation in cloze-style probes; our work extends this finding to chain-of-thought reasoning in instruction-tuned models.

Process reward models. Uesato et al. (2022) and Lightman et al. (2024) introduced process-based supervision for mathematical reasoning, demonstrating that step-level feedback outperforms outcome-only supervision. Creswell et al. (2023) proposed selection-inference chains for interpretable reasoning. Our ConsistPRM extends process reward models to consistency evaluation, with the finding that formal verification quality determines the effectiveness of step-level supervision.

6 Conclusion and Limitations

We introduced ConsistBench, a Z3-verified diagnostic benchmark that evaluates LLM reasoning through transformation invariance, and ConsistPRM, a process reward model for step-level consistency scoring. Our evaluation reveals a pervasive negation consistency gap across six models, driven by three systematic failure modes. The ConsistPRM pilot demonstrates that formal supervision quality is the primary bottleneck for consistency-aware training.

Limitations. ConsistBench is a diagnostic suite of 162 base instances, not a comprehensive benchmark. Its coverage of logical phenomena is deliberately narrow: it focuses on propositional and simple first-order logic and does not yet include double negation, De Morgan, or paraphrase invariance probes. The ConsistPRM results are based on a pilot study with 1,200 training traces; scaling behavior remains to be established. The error taxonomy is derived from manual analysis of a single model (GPT-4o) and may not transfer to other architectures. We plan to release all code, data, and Z3 encodings to support reproducibility and community extension.

Reproducibility Statement

All ConsistBench instances, Z3 encodings, transformation scripts, model prompts, and evaluation code will be released upon publication. The benchmark can be reconstructed from the released Z3 specifications, providing a machine-verifiable ground truth independent of our annotations.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *International Conference on Learning Representations*, 2023.

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.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. Understanding by understanding not: Modeling negation in language models. *arXiv preprint arXiv:2105.03519*, 2021.

Arkil Joshi, Eduardo Assessino, Jiawei Liu, Haoming Luo, and Jiaxin Chen. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2023 Conference of the North American Chapter of the Association for Computational Linguistics*, 2023.

Nora Kassner and Hinrich Schütze. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7811–7818, 2020.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe Alignment. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2024.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*, 2023.

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.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36, 2024.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

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. *arXiv preprint arXiv:2203.11171*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

A Additional Worked Examples

We provide three additional base-probe pairs with Z3 encoding sketches.

Example 1 (Modus Ponens → Negation).

Base: "If a student passes the exam, they graduate. Alice passed the exam. Does Alice graduate?" Answer: **True**.

Negation probe: "If a student passes the exam, they graduate. Alice did *not* pass the exam. Does Alice graduate?" Answer: **False**.

Z3 encoding: for the base instance. The probe replaces with and checks whether is entailed (it is not, since only the forward implication is given).

GPT-4o response to probe: "Since the rule says passing implies graduating, and Alice did not pass, we cannot conclude she graduates. *However*, the question asks if she graduates, and based on the rule... **True**." This exemplifies polarity inversion: correct identification of the negated premise followed by an unjustified reversion to the original answer.

Example 2 (Universal Quantifier → Negation with Scope).

Base: "Every employee in the department has completed the training. John is in the department. Has John completed the training?" Answer: **True**.

Negation probe: "Not every employee in the department has completed the training. John is in the department. Has John completed the training?" Answer: **False** (the entailment from universal to particular no longer holds).

GPT-4o response: "Every employee has completed the training, so John has. **True**." The model entirely ignores the word "Not," a clear case of negation blindness.

Example 3 (Transitivity Probe).

Base: "If A then B. If B then C. A is true. Is C true?" Answer: **True**.

Transitivity probe: "If A then C. A is true. Is C true?" Answer: **True**.

All models answer this probe correctly (88% for GPT-4o), confirming that the transitivity transformation is substantially easier to handle than negation.

B Full Commutation Consistency Results

For the three models where we computed CC (GPT-4o, Claude 3.5, Llama-3.1-70B), Table 3 reports per-transformation results alongside the base accuracy. The close tracking between CC and PC (Table 1) when base accuracy exceeds 0.90 validates the theoretical relationship described in Section 2.1.

Table 3: Full commutation consistency (CC) results.

Model	**Base Acc.**	**Ent. CC**	**Neg. CC**	**Contra. CC**	**Trans. CC**
GPT-4o	0.95	0.93	0.15	0.80	0.90
Claude 3.5 Sonnet	0.94	0.91	0.17	0.78	0.87
Llama-3.1-70B	0.91	0.87	0.11	0.72	0.81

C Negation Probe Answer Distribution

Table 4 reports the ground-truth label distribution and model prediction distribution on negation probes, confirming that the near-zero scores are not attributable to label imbalance.

Table 4: Answer distributions for negation probes.

	% True	**% False**
Base instance ground truth	54	46
Negation probe ground truth	46	54
GPT-4o predictions on negation probes	89	11
Claude 3.5 predictions on negation probes	86	14
Llama-3.1-70B predictions on negation probes	92	8

D ConsistPRM Negation-Specific Performance

Table 5 breaks down PRM step-level classification performance specifically on negation probe traces, by error type. Z3-verified labels provide the largest gains on scope confusion cases, which are the hardest for surface-level heuristics to detect.

Table 5: PRM step classification recall on negation probes, broken down by error type.

Label Source	**Blindness**	**Pol. Inversion**	**Scope Conf.**
Heuristic	0.31	0.19	0.08
LLM-judge	0.52	0.38	0.21
Z3-verified	**0.74**	**0.65**	**0.53**