PuzzleBench-100: A Solver-Verified Diagnostic Puzzle Suite for Closed-World Textual Deduction and Premise-Level Faithfulness

Anonymous ACL submission

Abstract

Large language models can answer deductive puzzles correctly while misreporting the premises or worlds that justify their answers. We introduce PuzzleBench-100, a solver-verified diagnostic puzzle suite for closed-world textual deduction and premise-level stated-support faithfulness. The suite contains 100 manually authored culinary-themed puzzles, each paired with an executable Prolog specification that derives satisfiability, model count, query determinacy, possible answers, false-premise behavior, world reconstruction labels, and minimal premise support. Each puzzle was authored in Polish and released with English plain and narrative variants. We evaluate models under three prompting conditions: status-and-answer prediction, structured support attribution, and full-world reconstruction. Results show that final-answer accuracy substantially overestimates faithful correctness: models often recover the answer while failing to identify minimal support or correct status/-world structure. PuzzleBench-100 is intended as a compact diagnostic collection for studying where answer correctness diverges from solver-grounded faithful reasoning.

1 Introduction

Large language models increasingly perform well on tasks that appear to require reasoning, but a correct final answer does not show whether the answer follows from the stated premises. A model may omit an unavoidable premise, cite a distractor, collapse several satisfying worlds into one, or give a determinate answer when the query is undetermined. These failures are hard to detect with answer-only evaluation.

Chain-of-thought and related inference-time methods improve final-answer accuracy (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023; Yao et al., 2023), but generated rationales need not reflect the premises that actually support the answer (Lanham et al., 2023; Turpin et al., 2023). This motivates diagnostic resources that test whether stated support is formally sufficient, minimal when required, and consistent with the space of possible worlds.

We introduce PuzzleBench-100, a solver-verified diagnostic puzzle suite for closed-world textual deduction and premise-level stated-support faithfulness. It contains 100 manually authored culinary-themed puzzles in a shared Bouffant Manor domain, with 200 English plain/narrative realizations. Each puzzle is paired with an executable Prolog specification that derives satisfiability, model count, query determinacy, possible answers, false-premise behavior, world reconstruction labels, unsatisfiable cores, and minimal sufficient premise sets. The suite is not intended as a leaderboard-scale benchmark, but as a compact diagnostic collection with dense executable metadata.

A central design feature is the separation between world status and query status. A puzzle may have multiple satisfying worlds while the query answer remains fixed, or it may be satisfiable while the query is undetermined. This distinction exposes failures hidden by answer-only evaluation, including over-answering and world collapse.

We evaluate models under three diagnostic prompting conditions: status-and-answer prediction, structured support attribution, and full-world reconstruction. Results show that final-answer accuracy substantially overestimates faithful correctness: models often recover the answer while failing to report correct statuses, false-premise behavior, reconstructed worlds, or minimal premise support.

Contributions. We introduce PuzzleBench-100, a solver-verified suite of 100 closed-world textual deduction puzzles with two language/style realizations each. The suite provides Prolog-derived labels for world and query status, model count, an-

swers, false-premise behavior, minimal support, critical premises, distractors, and unsatisfiable cores. We also define premise-level stated-support faithfulness metrics and a diagnostic protocol for answer, status, support, and world-reconstruction evaluation.

2 Related Work

Reasoning prompts and faithfulness. Chain-of-thought and related inference-time methods improve final-answer accuracy, including zero-shot step-by-step prompting, self-consistency, and tree-structured search (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023; Yao et al., 2023). However, generated rationales need not be faithful to the premises that support the answer (Lanham et al., 2023; Turpin et al., 2023). PuzzleBench-100 follows this line of work, but evaluates stated-support faithfulness at the level of numbered premises: support must be sufficient, minimal when required, and aligned with solver-derived metadata.

Formal and solver-based reasoning benchmarks. ProofWriter, FOLIO, P-FOLIO, and LogicBench evaluate proof construction, first-order reasoning, or systematic logical inference (Tafjord et al., 2021; Han et al., 2024; Parmar et al., 2024). Closest to our setting are puzzle-like and constraint-based evaluations: ZebraLogic studies logic-grid puzzles, SATBench generates SAT-derived natural-language puzzles with solver validation, and FineReason evaluates intermediate state checking in reflective puzzle solving (Lin et al., 2025; Wei et al., 2025; Chen et al., 2025). PuzzleBench-100 instead treats each item as a finite closed-world model-construction puzzle and records satisfiability, model count, query determinacy, false-premise behavior, and premise-level support.

Unlike ZebraLogic and SATBench, which primarily target answer accuracy on grid or SAT-derived puzzles, PuzzleBench-100 treats world/-query status, model count, false-premise behavior, and minimal sufficient support as first-class targets. Unlike FineReason, which evaluates intermediate state checking, PuzzleBench-100 evaluates whether reported support is sufficient, minimal, and consistent with the solver-verified admissible worlds. It is therefore diagnostic not only of answer correctness, but of whether the stated explanation aligns with the finite model structure that makes the answer valid.

Solver use. Solver-aided methods such as Logic-LM, Faithful CoT, and LLM-Modulo use symbolic execution or verification during inference (Pan et al., 2023; Lyu et al., 2023; Kambhampati et al., 2024). In contrast, PuzzleBench-100 does not give the model a solver: Prolog is used only to validate the collection and evaluate outputs after generation. The suite is diagnostic rather than leaderboard-oriented, testing not only final answers but also world status, query status, false-premise behavior, and minimal premise support.

3 PuzzleBench-100

PuzzleBench-100 is a solver-verified diagnostic suite for closed-world textual deduction and premise-level stated-support faithfulness. It contains 100 manually authored culinary puzzles in a shared Bouffant Manor domain, inspired by Tran Anh Hung's film *La passion de Dodin Bouffant*. Each puzzle from PB001 to PB100 is paired with a semi-formal JSON annotation and an executable Prolog validator. Model-facing inputs contain only the closed-world preamble, numbered premises, query, and condition-specific output instructions; solver outputs and support metadata are used only for validation and evaluation.

Each puzzle defines a finite closed world. The preamble specifies the entity set, active domains, exactly-one assumptions, object-location rules, and witness rules. Numbered premises provide the instance-specific evidence from which the solver derives answers, statuses, model counts, false-premise behavior, world assignments, and support labels.

Let P be the numbered premises and A_{pre} the preamble axioms. The solver enumerates admissible worlds under the specified truth regime with respect to $P \cup A_{\text{pre}}$; let $M(P)$ be the resulting model set. The world status is SAT_UNIQUE, SAT_MULTI, or UNSAT depending on whether $|M(P)|$ is 1, greater than 1, or 0. For satisfiable instances, the query answer set is $A(q, P) = \{a(q, m) : m \in M(P)\}$; the query status is QUERY_DETERMINED if this set is singleton and QUERY_UNDETERMINED otherwise. NO_MODEL and NO_SUCH_ENTITY cover inconsistent worlds and out-of-domain queries. This separation exposes answer-only blind spots: a world may be non-unique while the queried answer is fixed, or satisfiable while the query remains undetermined.

Most instances use ALL_TRUE; later diagnos-

tic instances introduce ONE_FALSE, where all premises outside a candidate set must be true and exactly one candidate premise must be false. Falsehood is evaluated at the numbered-premise level, including compound premises with conjunction, disjunction, implication, biconditionality, or witness statements.

The puzzles were authored in Polish first. For each base puzzle, we created a Polish narrative version, validated it against the Prolog specification, and then shortened it into a Polish plain variant that removes literary ornamentation while preserving the logical setup and numbered premises. The English plain and narrative variants used in the reported experiments were produced by controlled translation from the corresponding Polish versions and manually checked for premise-level semantic equivalence. Thus, each experimental puzzle has two English realizations, plain and narrative, with the same formal specification and solver-derived metadata. Difficulty labels were assigned based on proof depth, number of premises, and logical constructions such as negation, implication, disjunction, biconditionality, and false-premise regimes.

For premise-level support, the solver derives one or more minimal sufficient premise sets for the requested target. Let $\mathcal{S}_P$ be the set of such sets:

$$C_P = \cap \mathcal{S}_P, \quad R_P = \cup \mathcal{S}_P, \quad D_P = P \setminus R_P.$$

Here, C_P is the critical set, R_P the relevant-support universe, and D_P the distractor set. Model outputs are evaluated against both reported `used_premises` and claimed `minimal_support`.

The construction pipeline maps natural-language premises to restricted JSON annotations, compiles them into Prolog validators, and derives statuses, answers, model counts, false-premise choices, world assignments, underdetermined slots, unsatisfiable cores, minimal supports, critical premises, and distractors. Instances were revised or rejected whenever the natural-language reading, JSON annotation, and Prolog-derived labels disagreed; the final suite contains only aligned instances.

Table 1 summarizes the solver-validated composition of the suite. The design deliberately crosses world status, query status, truth regime, difficulty, entity count, premise count, and distractor count; for example, PB043 is SAT_MULTI but QUERY_DETERMINED, with two satisfying worlds that agree on the queried cook.

Dimension	**Breakdown**	**Count**
Base puzzles	–	100
Realizations	EN × plain/narrative	200
Truth regime	ALL_TRUE	67
	ONE_FALSE	33
World status	SAT_UNIQUE	59
	SAT_MULTI	35
	UNSAT	6
Query status	QUERY_DETERMINED	68
	QUERY_UNDETERMINED	22
	NO_MODEL	10
Difficulty	Easy	19
	Medium	35
	Medium-hard	16
	Hard	30
Entities	3 / 4	52 / 48
Premises	3–5	38
	6–8	32
	9+	30
Depth	2–4	42
	5–6	37
	7–11	21
Distractors	0	50
	1–2	25
	3+	25

Table 1: Summary statistics for PuzzleBench-100 over 100 base puzzles.

4 Metrics

PuzzleBench-100 evaluates answer correctness, status correctness, and stated-support faithfulness. All metrics are computed deterministically from parsed outputs and solver metadata; no LLM judge is used.

For each instance i, the solver provides the gold answer set A_i, world status w_i, query status q_i, model count c_i where requested, false-premise set F_i where applicable, and minimal sufficient premise sets $\mathcal{S}_i$. From these, we define:

$$C_i = \bigcap_{S \in \mathcal{S}_i} S, \quad R_i = \bigcup_{S \in \mathcal{S}_i} S, \quad D_i = P_i \setminus R_i.$$

Here, C_i is the critical set, R_i the relevant-support universe, and D_i the distractor set.

Answer and status correctness. We report exact Answer Accuracy, World Status Accuracy, Query Status Accuracy, Model-Count Accuracy where requested, and False-Premise Accuracy where applicable. For undetermined queries, the answer is correct only if the full possible-answer set is reported.

Premise-level faithfulness. Let $\hat{S}_i$ be the model-reported `minimal_support`. Critical Premise Coverage (CPC) measures recall over unavoidable premises:

$$\mathrm{CPC}_i = \begin{cases} |C_i \cap \hat{S}_i|/|C_i|, & |C_i| > 0, \\ 1, & |C_i| = 0. \end{cases}$$

Sufficient Faithful Correctness (SFC) requires the answer and statuses to be correct and $\hat{S}_i$ to contain some gold minimal set, i.e. $\exists S \in \mathcal{S}_i : S \subseteq \hat{S}_i$. Minimal Faithful Correctness (MFC) is stricter and requires exact match with one gold minimal set, i.e. $\exists S \in \mathcal{S}_i : \hat{S}_i = S$.

Support errors and full correctness. We report Missing Critical Premise when $C_i \setminus \hat{S}_i \neq \emptyset$, Non-Minimal Support when $\hat{S}_i$ strictly contains a gold minimal set, and Distractor Inclusion Rate:

$$\mathrm{DIR}_i = \begin{cases} |D_i \cap \hat{S}_i|/|\hat{S}_i|, & |\hat{S}_i| > 0, \\ 0, & |\hat{S}_i| = 0. \end{cases}$$

For full-world prompts, we also report World Reconstruction Accuracy, World-Collapse Rate, and Over-Answering Rate. Full Correctness requires all requested components—answer, statuses, false-premise behavior, model count, world reconstruction, and minimal support—to be correct.

5 Evaluation Protocol

The goal is to test whether models that recover the final answer also report correct status labels, premise support, and finite-world structure. Model-facing inputs contain only task instruction, closed-world preamble, the narrative or plain puzzle version, numbered premises, query, and condition-specific output instructions; JSON annotations, Prolog specifications, solver outputs, gold labels, and support metadata are never shown.

We compare three prompting conditions of increasing difficulty. L1 asks for JSON with `world_status`, `query_status`, `answer`, and `false_premise` where applicable. L2 adds `deduction_steps`, `open_alternatives`, `used_premises`, and `minimal_support`. L3 further asks for `full_world_reconstruction`, `possible_worlds` or `invariant_assignments`, and `underdetermined_slots`.

Outputs are parsed deterministically. If a model returned malformed JSON, the same prompt was retried up to three times; outputs that remained unparsable were marked as INVALID_JSON. Premise references are normalized to numbered premise IDs, and support-level metrics are computed from the model-reported `minimal_support` field and solver-derived support metadata. No LLM judge is used for the metrics.

We assign deterministic, non-exclusive error labels from parsed outputs and solver metadata, separating answer-level failures from status errors, support-attribution errors, and full-world reconstruction errors.

Full prompt templates, output schemas, parsing rules, and the malformed-output policy are included in the supplementary artifact. Response latency and output-token length were not evaluated, as our focus is diagnostic correctness rather than efficiency.

Inference setup. All model calls were executed through OpenRouter (OpenRouter, 2026) with temperature set to 0. Tool use was disabled by setting `tool_choice` to `none`, so the evaluation focused on model reasoning rather than external solvers, code execution, search, calculators, or provider-side tools. Outputs were limited to 10k tokens, including at least 2k reasoning tokens where supported. Prolog validators were used only offline, after generation, to validate the collection and score outputs. Full OpenRouter model identifiers are provided in the supplementary artifact.

For each model, we submitted the same model-facing prompt for each puzzle variant and prompting level. The prompt contained only the task instruction, closed-world preamble, numbered premises, query, and required JSON output schema; solver outputs, Prolog code, gold labels, model counts, false-premise labels, and support metadata were never included. Each query was evaluated in isolation.

We evaluated Claude Sonnet 4.5, DeepSeek V4 Pro, Gemini 3.1 Flash Lite, Gemma 4 31B IT, Kimi K2 Thinking, GPT-5.2, and Qwen 3.5 27B. These models were selected because they are publicly positioned as strong reasoning or instruction-following systems and together cover a broad range of model families and providers. Evaluations were conducted in May 2026.

5.1 Error Taxonomy

Table 2 lists the deterministic, non-exclusive error labels used for support and world-level analysis.

Error	Meaning
INVALIDFORMAT	Required structured fields are missing or unparsable.
ANSWERERROR	Final answer differs from the gold answer in a determined instance.
WORLDSTATUSERROR	Predicted world status is incorrect.
QUERYSTATUSERROR	Predicted query status is incorrect.
OVERANSWERING	The model gives an answer for an undetermined query.
WORLDCOLLAPSE	The gold world is SAT_MULTI, but the model reports only one world.
NONECOLLAPSE	A gold answer exists, but the model returns none.
NOSUCHENTITYERROR	The model incorrectly reports an out-of-domain query.
INVALIDMINIMALSUPPORT	minimal_support matches no solver-derived minimal sufficient set.
MISSINGCRITICALPREMISE	A required critical premise is omitted.
NONMINIMALSUPPORT	Extra non-critical support is included.
DISTRACTORINCLUDED	A solver-derived distractor is cited.
INVALIDWORLDBUILT	The reconstructed world violates constraints or premises.

Table 2: Deterministic error labels used for support and world-level analysis.

5.2 Release Plan and Reproducibility

For review, we provide an anonymized supplementary archive containing the PB001–PB100 JSON records, plain and narrative puzzle realizations, prompt templates, Prolog validators, model outputs, and scripts for deterministic parsing, scoring, and summary generation. The artifact is intended to support both result reproduction and further evaluation: any puzzle can be re-run with an additional model using the provided prompts and scripts, and its output can be scored against the same solver-derived labels.

6 Results and Analysis

We report completed runs for all seven models across three prompting levels and two style variants, for a total of $7 \times 3 \times 2 \times 100 = 4{,}200$ evaluated instances. Each model-condition pair contains 100 processed instances, so all values in Tables 3–5 are percentages. The results are not intended as a leaderboard; they show how final-answer accuracy diverges from status correctness, support faithfulness, and finite-world reconstruction.

L1: answer/status prediction. Table 3 shows that L1 is the least difficult condition, but it already reveals a gap between answer accuracy and full diagnostic correctness. The strongest answer score reaches 84% in both style variants, while full correctness reaches only 72% on plain variants and 76% on narrative variants. Thus, even when models recover the final answer, they may still fail on status labels, false-premise behavior, or other required structured fields.

Narrative variants are not uniformly harder. In L1, every model improves in full correctness on narrative inputs, suggesting that additional scenario context can sometimes provide useful scaffolding rather than only distraction.

L2: support attribution. Table 4 shows a sharper drop when models must report premise-level support. Answer accuracy remains relatively high, reaching 76% on plain variants and 82% on narrative variants. Full correctness, however, drops to 42% and 41%, respectively. This widening gap supports the central motivation of PuzzleBench-100: correct answers do not imply solver-faithful stated support.

The support metrics reveal different model profiles. Claude obtains the strongest sufficient faithful correctness in both styles, whereas GPT-5.2 obtains the strongest full correctness. Sufficient premise recovery and complete structured correctness are therefore related but distinct diagnostic targets.

L3: full-world reconstruction. Table 5 shows that full-world reconstruction is harder still. Models can reconstruct much of the finite world structure while failing full diagnostic correctness. Claude reaches 91% world-reconstruction accuracy in both styles, and GPT-5.2 reaches 90% exact world-set match in both styles. Nevertheless, full correctness remains much lower: 35% on plain variants and 44% on narrative variants.

This separates world reconstruction from faithful deductive correctness. A model may approximate the admissible world set while still failing minimal support, model count, query status, or another requested component.

Main finding. Across all three levels, final-answer accuracy substantially overestimates full diagnostic correctness. The gap is modest in L1 but becomes large in L2 and L3: in L3, answer

	Plain			Narrative		
Model	Ans.	Full	FalsePrem.	Ans.	Full	FalsePrem.
Claude Sonnet 4.5	80	**72**	90	**84**	**76**	92
DeepSeek V4 Pro	73	70	86	78	71	86
Gemini 3.1 Flash Lite	**84**	70	89	82	73	**93**
Gemma 4 31B IT	75	**72**	81	80	73	89
Kimi K2 Thinking	73	68	91	81	74	90
GPT-5.2	71	65	**94**	79	**76**	**93**
Qwen 3.5 27B	69	64	74	73	66	76

Table 3: L1 results for status-and-answer prediction. All runs contain 100 instances; values are percentages. Best values within each style variant are bolded.

	Plain						Narrative					
Model	Ans.	Full	FalsePrem.	CPC	SFC	MFC	Ans.	Full	FalsePrem.	CPC	SFC	MFC
Claude Sonnet 4.5	75	38	**91**	52	**60**	48	**82**	38	**93**	51	**68**	**52**
DeepSeek V4 Pro	56	34	68	43	45	41	60	39	67	46	49	45
Gemini 3.1 Flash Lite	**76**	33	88	51	48	43	78	38	92	**57**	55	48
Gemma 4 31B IT	68	32	81	46	50	41	71	30	79	40	52	44
Kimi K2 Thinking	56	30	70	37	43	35	64	33	72	41	48	39
GPT-5.2	70	**42**	84	**53**	56	**50**	73	**41**	83	52	50	46
Qwen 3.5 27B	64	32	69	44	49	37	61	33	65	45	50	41

Table 4: L2 results for structured support attribution. All runs contain 100 instances; values are percentages. CPC = Critical Premise Coverage, SFC = Sufficient Faithful Correctness, and MFC = Minimal Faithful Correctness. Best values within each style variant are bolded.

	Plain							Narrative						
Model	Ans.	Full	FalseP.	CPC	MFC	WRecon	WSet	Ans.	Full	FalseP.	CPC	MFC	WRecon	WSet
Claude	79	29	**93**	56	51	**91**	86	**83**	26	**93**	57	49	**91**	82
DeepSeek	57	24	68	42	38	60	69	57	26	70	45	42	59	70
Gemini	**81**	24	88	**57**	47	87	71	73	29	89	**61**	49	87	69
Gemma	72	29	82	52	43	77	80	66	22	74	41	44	70	74
Kimi	57	23	64	36	30	61	62	52	19	68	35	29	62	63
GPT-5.2	71	**35**	89	54	**53**	86	**90**	78	**44**	89	57	**55**	86	**90**
Qwen	63	25	71	46	39	68	67	59	23	66	42	37	63	62

Table 5: L3 results for full-world reconstruction. All runs contain 100 instances; values are percentages. FalseP. = False-Premise Accuracy, CPC = Critical Premise Coverage, MFC = Minimal Faithful Correctness, WRecon = World Reconstruction Accuracy, and WSet = exact world-set match. Best values within each style variant are bolded.

accuracy remains above 80% for the strongest models, while full correctness remains below 45%. PuzzleBench-100 therefore exposes failures hidden by answer-only evaluation.

6.1 Diagnostic Error Analysis

The aggregate results are supported by representative item-level failures. Table 6 shows three cases illustrating why answer-only scoring is insufficient.

PB082 shows that errors are not limited to overconfident answering. The gold structure has one model, a determined answer, and one false premise; however, Claude in the narrative L1 condition reports a multi-world, underdetermined interpretation and returns neither the answer nor the false premise. In higher-level prompts, the same model recovers the answer and world status, but still includes a distractor and invalid support. This shows that status prediction, answer recovery, false-premise detection, and support faithfulness can fail independently.

PB043 illustrates the need to separate world status from query status. The item has two satisfying worlds, but both yield the same answer. GPT-5.2 can recover the answer while still misreporting the world/query structure or omitting critical support in structured conditions. Such cases would be scored as correct by answer-only evaluation, although the model misrepresents why the answer is determined.

PB015 shows a support-only failure. Qwen pre-

Item / model	Gold structure	Model behavior	Interpretation
PB082 / Claude / narr.	SAT_UNIQUE; QUERY_DETERMINED; answer: Rita; false premise: p11	L1 reports SAT_MULTI, QUERY_UNDETERMINED, no answer, and no false premise.	Over-uncertainty: the model fails to commit to a solver-unique world and loses both the answer and the false-premise label.
PB043 / GPT-5.2	SAT_MULTI; QUERY_DETERMINED; answer: John	Gives the correct answer, but misreports world/query structure or critical support depending on condition.	World/query decoupling error: the answer is correct, but the model misrepresents the admissible world space or its support.
PB015 / Qwen	SAT_UNIQUE; QUERY_DETERMINED; answer: Sebastiao	Preserves answer and status labels, but reports invalid or non-minimal support in L2/L3.	Support-only failure: the final answer is correct, but the stated premise support is not solver-faithful.

Table 6: Representative errors illustrating over-uncertainty, world/query decoupling, and support-only failure.

serves the correct answer and status labels, but its L2/L3 outputs contain invalid or non-minimal support. Correct answers and world reconstruction do not guarantee premise-level faithfulness: a model may solve the puzzle extensionally while failing to report a solver-valid justification.

7 Discussion

PuzzleBench-100 is best understood as a compact diagnostic suite rather than a leaderboard-scale benchmark. Its value lies in dense executable metadata: model counts, query determinacy, false-premise behavior, minimal sufficient supports, critical premises, distractors, and world-reconstruction checks.

The completed results support this diagnostic framing. The strongest models often remain above 80% answer accuracy, but full correctness falls below 45% in the most demanding L3 condition. This gap is not explained solely by world reconstruction failures: some models achieve high reconstruction or world-set scores while still failing full correctness. The bottleneck is therefore jointly satisfying answer, status, false-premise, support, and world-level requirements.

The paired plain-narrative variants show that linguistic elaboration is not uniformly harmful: for some models, narrative context provides useful scenario scaffolding, while for others it increases the burden of filtering formally relevant premises. Although the Bouffant Manor domain is intentionally narrow, the exposed failures are structural properties of closed-world reasoning rather than domain artefacts, and should transfer to finite-model tasks with similar satisfiability and support-attribution requirements.

8 Conclusion

We presented PuzzleBench-100, a compact solver-verified diagnostic suite for closed-world textual deduction and premise-level stated-support faithfulness. By pairing natural-language puzzles with executable Prolog specifications, the suite provides solver-derived labels for answer/world status, model count, query determinacy, false-premise behavior, minimal support, and world reconstruction. The results show that answer-only evaluation substantially overestimates faithful reasoning: models may recover the correct final answer while misreporting status labels, collapsing admissible worlds, over-answering underdetermined queries, or citing insufficient support. PuzzleBench-100 exposes these gaps by testing whether a model's answer, stated support, and reconstructed world are jointly consistent with the finite solver-verified space of admissible worlds.

9 Limitations

PuzzleBench-100 is a diagnostic puzzle suite rather than a leaderboard-scale corpus, with 100 base puzzles and four language/style realizations per puzzle. The fixed Bouffant Manor domain controls confounds but limits ecological breadth; models repeatedly exposed to the schema could learn domain patterns rather than general reasoning strategies, although we vary reasoning structure, status class, reliability regime, and false-premise behavior.

All puzzles were authored by one author, creating a construct-validity risk. However, final labels are solver-derived from executable Prolog validators, and puzzle acceptance required three-way alignment between the natural-language reading, semi-formal JSON annotation, and executable validator; any disagreement triggered revision or re-

jection, making the solver the de facto second annotator for all structural labels.

Finally, the suite evaluates stated-support faithfulness, not latent model cognition.

References

Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Chaoqun Liu, Lidong Bing, Deli Zhao, Anh Tuan Luu, and Yu Rong. 2025. FineReason: Evaluating and improving LLMs' deliberate reasoning through reflective puzzle solving. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, Martin Riddell, Wenfei Zhou, Yujie Qiao, Yilun Zhao, Semih Yavuz, Ye Liu, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Dragomir Radev, Rex Ying, and Arman Cohan. 2024. P-FOLIO: Evaluating and improving logical reasoning with abundant human-written reasoning chains. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16553–16565, Miami, Florida, USA. Association for Computational Linguistics.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, and Jared Kaplan. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

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

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*.

OpenRouter. 2026. OpenRouter API Documentation. https://openrouter.ai/docs. Accessed: 2026-05-24.

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

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. 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)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

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

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*.

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

Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. 2025. SATBench: Benchmarking LLMs' logical reasoning via automated puzzle generation from SAT formulas. *arXiv preprint arXiv:2505.14615*.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*.

A Supplementary Artifact

The anonymized supplementary artifact includes the full L1–L3 prompt templates, translation prompt, Prolog validators, puzzles with JSON records, paired plain and narrative puzzle realizations, evaluation scripts, model outputs, and instructions for running additional evaluations.