BEYOND CLAUSE COUNT: A STUDY OF PROOF-RELEVANT DIFFICULTY IN LLM SAT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

SAT has recently emerged as a controlled setting for evaluating logical reasoning in large language models (LLMs). However, existing SAT-oriented benchmarks mainly vary surface-level properties such as formula size (e.g., clause count and variable count), phase-transition regime, or question format, but these axes do not directly capture refutational difficulty on UNSAT instances. We propose proof complexity as a theory-grounded lens for diagnosing LLM failures on UNSAT reasoning and instantiate this perspective in a preliminary study using clause-controlled Tseitin formulas together with two auxiliary control families, implication-chain formulas and random 3-CNF instances. Using direct CNF input, we find that aggregate accuracy often hides strong label-specific bias, and that structurally different Tseitin families can remain substantially different in difficulty even when clause budgets are matched. These results suggest that surface statistics such as clause count and variable count are insufficient for understanding LLM performance on satisfiability judgments, and motivate broader evaluations of proof-relevant hardness in formal reasoning benchmarks.

1 INTRODUCTION

Logical reasoning has become a central stress test for large language models (LLMs), motivating a growing ecosystem of benchmarks ranging from synthetic proof tasks to richer first-order and multi-logic evaluations such as ProofWriter, FOLIO, LogicBench, and Multi-LogiEval (Tafjord et al., 2021; Han et al., 2024; Parmar et al., 2024; Patel et al., 2024). At the same time, recent work has shown that final-answer accuracy alone is often insufficient for understanding LLM reasoning quality, while solver-augmented approaches such as LINC, Logic-LM, and SatLM demonstrate that symbolic representations and external provers can substantially improve reliability (Anonymous, 2025; Olausson et al., 2023; Pan et al., 2023; Ye et al., 2023).

SAT has now emerged as a particularly useful controlled setting for evaluating LLM reasoning. Existing SAT-oriented studies already manipulate difficulty through several axes: SATBench varies formula size, Hazra et al. analyze random 3-SAT phase transitions, SATQuest controls instance scale, problem type, and question format, and recent parameterized 2-SAT work emphasizes that surface properties such as wording or clause order can obscure deeper structural effects (Wei et al., 2025; Hazra et al., 2025; Zhao et al., 2025; Es-sebbani et al., 2026). These works show that SAT is a powerful evaluation environment, but they also leave open a key question: for *UNSAT* instances, what notion of difficulty best explains when and why LLMs fail?

This question matters because standard size-based proxies are not the same as refutational hardness. Clause count, variable count, and input length measure how large an instance appears, but not how difficult it is to refute. Likewise, phase-transition regimes and question formats are informative, but they do not directly characterize the proof search required to establish unsatisfiability (Hazra et al., 2025; Zhao et al., 2025; Es-sebbani et al., 2026). We therefore use proof complexity as a theory-grounded diagnostic lens. We do not claim that LLMs internally execute resolution; rather, we ask whether proof-relevant structural distinctions help explain observed model failures on UNSAT reasoning.

This perspective also motivates our use of direct CNF input. Recent SAT-oriented evaluations already include direct-formula settings, and SATBench reports that raw SAT formulas are often easier for LLMs than natural-language puzzle versions, suggesting that verbalization introduces additional

difficulty beyond the underlying logical problem (Wei et al., 2025; Hazra et al., 2025; Zhao et al., 2025). Direct CNF input therefore helps isolate formal reasoning difficulty from natural-language packaging effects.

In this paper, we present a preliminary study that combines one proof-complexity-motivated benchmark family with two auxiliary controls. Our main analysis uses clause-controlled Tseitin formulas derived from multiple graph families, while implication-chain formulas and random 3-CNF instances serve as easy and generic-hard controls. We find that structurally different Tseitin families can remain substantially different in difficulty even under clause-controlled evaluation, providing preliminary support for going beyond clause count and other surface statistics toward more structurally informed notions of difficulty in SAT-oriented LLM evaluation.

2 PRELIMINARIES

A *resolution proof* of a clause C from a CNF formula F is a sequence of clauses in which each clause is either an initial clause of F or is obtained from two earlier clauses by one application of the resolution rule

$$(C_1 \vee x), (C_2 \vee \neg x) \;\vdash\; (C_1 \vee C_2).$$

When the derived clause is the empty clause $\perp$, the proof is called a *resolution refutation*. Thus, for an unsatisfiable CNF formula, a resolution refutation is a formal certificate of unsatisfiability.

In this paper, resolution serves as an external proof-theoretic lens for UNSAT reasoning. Intuitively, the length of a resolution refutation measures how difficult it is to derive a contradiction from the clauses of the formula.

2.1 TSEITIN FORMULAS

A *Tseitin formula* $T(G, c)$ is defined from an undirected graph $G = (V, E)$ and a charge function $c : V \to \{0, 1\}$. For each edge $e \in E$, we introduce a Boolean variable x_e. For every vertex $v \in V$, let $E(v)$ denote the set of edges incident to v. The parity condition at v is

$$P_v \;:=\; \left(\sum_{e \in E(v)} x_e \equiv c(v) \right) \bmod 2. \tag{1}$$

Equivalently, P_v is an XOR equation over the variables corresponding to edges incident to v. The Tseitin formula is the conjunction of these parity constraints:

$$T(G, c) \;:=\; \bigwedge_{v \in V} P_v.$$

To express $T(G, c)$ in CNF, each parity equation P_v is translated into its canonical CNF encoding, consisting of $2^{|E(v)|-1}$ clauses, each of width $|E(v)|$. By abuse of notation, we still write $T(G, c)$ for the resulting CNF formula.

A standard fact is that if the total charge $\sum_{v \in V} c(v)$ is odd modulo 2, then $T(G, c)$ is unsatisfiable.

For such formulas, the following result links graph structure to proof complexity.

Lemma 1 (Itsykson et al. (2022))**.** *For any connected graph $G = (V, E)$ of maximum degree at most d, and any odd charge function c, the shortest regular-resolution refutation of $T(G, c)$ has length $2^{\Omega(\mathrm{tw}(G))}$.*

Here $\mathrm{tw}(G)$ is the treewidth of G. Moreover, this lower bound is essentially tight for bounded-degree Tseitin formulas. Prior work gives a regular-resolution upper bound of the form

$$2^{O(\mathrm{tw}(G))} \operatorname{poly}(|V(G)|),$$

and de Colnet and Mengel show that the lower bound essentially matches this upper bound (de Colnet & Mengel, 2023; Itsykson et al., 2022). Thus, in the bounded-degree Tseitin setting, treewidth is not merely a convenient heuristic proxy, but a theorem-backed indicator of regular-resolution complexity.

Table 1: Surface-level size measures do not directly capture refutational hardness.

Formula family	Variable count	Clause count	Refutation length
Contradiction chain	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Tseitin on constant-degree expanders	$\Theta(n)$	$\Theta(n)$	$2^{\Omega(n)}$

3 Proof-Relevant Hardness Metric

As reviewed in Section 2, resolution provides a natural proof-theoretic lens for UNSAT reasoning and is closely connected to proof systems underlying modern SAT solving (Buss & Nordström, 2021; Järvisalo et al., 2012). In particular, for unsatisfiable formulas, resolution-based complexity measures provide a principled way to talk about how hard a formula is to refute, rather than merely how large it appears syntactically. Because an UNSAT judgment fundamentally concerns recognizing that no satisfying assignment exists, proof complexity provides a theory-grounded external way to distinguish formulas that are easy versus hard to refute.

Why replace surface-level size metrics? Clause count and variable count are standard surface-level size proxies, and input length is another closely related one. However, these quantities do not directly characterize how hard a formula is to refute. A simple contrast illustrates this point. The contradiction chain

$$x_1,\ (\neg x_1 \vee x_2),\ (\neg x_2 \vee x_3),\ \ldots,\ (\neg x_{n-1} \vee x_n),\ \neg x_n$$

has $\Theta(n)$ variables, $\Theta(n)$ clauses, and a linear-length resolution refutation obtained by repeated unit propagation. For example, when $n = 3$, the formula

$$x_1, \quad (\neg x_1 \vee x_2), \quad (\neg x_2 \vee x_3), \quad \neg x_3$$

admits the resolution derivation

$$x_1,\ (\neg x_1 \vee x_2)\ \vdash\ x_2, \qquad x_2,\ (\neg x_2 \vee x_3)\ \vdash\ x_3, \qquad x_3,\ \neg x_3\ \vdash\ \bot.$$

In contrast, bounded-degree odd-charge Tseitin formulas under the canonical encoding also have $\Theta(n)$ variables and $\Theta(n)$ clauses, yet their shortest regular-resolution refutations can have length $2^{\Omega(\mathrm{tw}(G))}$ by Lemma 1. In particular, for constant-degree expander families, where $\mathrm{tw}(G) = \Omega(n)$, this yields a lower bound of $2^{\Omega(n)}$. Thus, even among formulas with comparable surface-level size, proof-theoretic difficulty can differ exponentially. This is precisely why we move beyond surface size and seek a proof-relevant structural hardness measure. Table 1 highlights this distinction: formulas with comparable clause and variable counts can still differ exponentially in proof-theoretic difficulty.

This observation is especially relevant in our setting, where clause budgets are intentionally matched across graph families: if formulas with comparable surface-level size can already exhibit exponentially different refutational complexity, then surface size alone is unlikely to explain model behavior on UNSAT instances.

This contrast is only a motivating example. In the actual benchmark, we instantiate the same idea more gradually using three Tseitin graph families with increasing structural complexity: cycles, grids, and random regular graphs.

Why Tseitin formulas? Tseitin formulas are particularly suitable for this purpose for two reasons. First, they provide a clean and controlled SAT/UNSAT construction: satisfiability can be switched simply by changing the parity of the charges while preserving the overall graph structure. Second, they come with rich proof-complexity theory linking structural properties of the underlying graph to refutational hardness (de Colnet & Mengel, 2023; Itsykson et al., 2022). This makes them especially useful for studying whether proof-relevant structural hardness is reflected in LLM behavior.

Why these three graph families? We choose `cycle`, `grid`, and `random_regular` because they provide a simple family-level progression in proof-relevant structure while remaining easy to generate and compare under clause-controlled conditions. Cycle graphs serve as a low-treewidth

Table 2: Family-level structural progression for the Tseitin benchmark. For bounded-degree Tseitin formulas, Lemma 1 implies a regular-resolution lower bound of $2^{\Omega(\mathrm{tw}(G))}$.

Graph family	Treewidth scale	Implied regular-resolution lower bound
Cycle	$\Theta(1)$	$2^{\Theta(1)}$
$k \times k$ grid	$\Theta(k) = \Theta(\sqrt{n})$	$2^{\Theta(\sqrt{n})}$
Fixed-degree random regular	$\Theta(n)$ a.a.s.	$2^{\Omega(n)}$ a.a.s.

baseline, with constant treewidth. Two-dimensional grids provide an intermediate regime with treewidth $\Theta(\sqrt{n})$, capturing formulas that are still locally structured but substantially less trivial. Fixed-degree random regular graphs provide a high-connectivity regime and have linear treewidth asymptotically almost surely, making them a natural source of structurally entangled Tseitin instances. Thus, these three families instantiate a coarse but theoretically motivated family-level progression in proof-relevant structure, from `cycle` to `grid` to `random_regular`. Even when clause budgets are matched, the treewidth-based proxy therefore suggests increasing structural difficulty across these families. Our experiments test whether this family-level progression is reflected in LLM behavior on UNSAT instances.

Remark. The use of treewidth in this paper is specific to bounded-degree Tseitin formulas. The reason is not that treewidth is a universal hardness measure for arbitrary CNF formulas, but rather that for Tseitin formulas we have a direct proof-complexity link from graph structure to regular-resolution hardness via Lemma 1. This theorem-backed connection is what allows us to use underlying graph structure as a proxy for difficulty in the present benchmark.

4 Experimental Setup

Scope of the current study. The current empirical study focuses on a controlled subset of SAT reasoning problems. Our main benchmark consists of clause-controlled Tseitin formulas with approximately matched overall size. We additionally include two auxiliary control families: implication-chain formulas as an easy local-propagation baseline, and random 3-CNF instances near the phase transition as a generic hard-CNF baseline. Concretely, implication-chain formulas are easy because their satisfiability structure is largely exposed by repeated unit propagation, whereas random 3-CNF formulas near the satisfiability phase transition are a classical source of hard SAT instances (Hazra et al., 2025; Mitchell et al., 1992).

Models and decoding. Unless otherwise noted, all experiments use the same direct CNF prompt with temperature 0.0, maximum generation length 512 tokens, and a request timeout of 300 seconds. Each model receives the raw DIMACS CNF formula as input and is instructed to output exactly one label, either `SAT` or `UNSAT`, without explanation. Outputs other than exactly `SAT` or `UNSAT` are counted as incorrect. We do not explicitly set top-p, so it follows the provider default.

Main benchmark. Our main benchmark consists of clause-controlled Tseitin formulas generated from three graph families: `cycle`, `grid`, and `random_regular`. These families instantiate the family-level structural progression motivated in Section 3: cycles provide a low-treewidth baseline, grids provide an intermediate regime, and random regular graphs provide a more entangled high-connectivity regime.

Labels and verification. Ground-truth labels were determined by construction: even-charge instances were labeled SAT and odd-charge instances were labeled UNSAT. We additionally verified all formulas with the Kissat solver as a sanity check. The current paper focuses on model predictions rather than solver-level hardness regression.

Evaluation metrics. We report overall accuracy together with label-conditional accuracy on SAT and UNSAT instances. Because the current study is centered on refutational difficulty, we also break down UNSAT accuracy by graph family. This allows us to test whether graph-structural differences

Table 3: Accuracy (%) on the clause-controlled medium-profile Tseitin benchmark, broken down by label. Since the benchmark is label-balanced, always predicting `SAT` or always predicting `UNSAT` yields 50.00% overall accuracy.

Model	Overall	SAT	UNSAT
Always SAT	50.00	100.00	0.00
Always UNSAT	50.00	0.00	100.00
deepseek/deepseek-r1	73.44	71.88	75.00
deepseek/deepseek-chat	54.69	15.63	93.75
anthropic/claude-3.5-sonnet	60.94	100.00	21.88
google/gemini-2.0-flash-001	57.81	93.75	21.88
openai/gpt-4.1-mini	43.75	56.25	31.25

Table 4: UNSAT accuracy (%) by graph family under matched clause-budget regimes. Cycle and Grid columns are computed over 8 UNSAT instances each, whereas Random-Regular is computed over 16 UNSAT instances.

Model	Cycle	Grid	Random-Regular
deepseek/deepseek-r1	100.00	87.50	56.25
deepseek/deepseek-chat	100.00	100.00	87.50
anthropic/claude-3.5-sonnet	75.00	12.50	0.00
google/gemini-2.0-flash-001	25.00	62.50	0.00
openai/gpt-4.1-mini	87.50	37.50	0.00

induce systematic performance differences even when clause budgets are matched and other size measures are only approximately controlled.

4.1 Results

Result 1: Overall accuracy hides strong label asymmetries. Table 3 shows that aggregate accuracy alone is insufficient to characterize model behavior on satisfiability judgments. Several models exhibit strong label-specific biases that are only weakly reflected in overall accuracy. For example, `deepseek/deepseek-chat` attains 93.75% on UNSAT instances but only 15.63% on SAT instances, indicating a strong bias in the opposite direction. In contrast, `deepseek/deepseek-r1` is comparatively more balanced across labels and also achieves the highest overall performance among the tested models. These results show that class-conditional evaluation is necessary: a model may appear competitive in aggregate accuracy while still failing systematically on one label. A different but equally striking asymmetry appears for `anthropic/claude-3.5-sonnet`, which achieves 100.0% on SAT instances but only 21.88% on UNSAT instances.

Result 2: Clause-controlled Tseitin instances still differ substantially in UNSAT difficulty. Table 4 examines UNSAT accuracy across graph families under matched clause budgets. Several models show a clear degradation from `cycle` to `grid` to `random_regular`. For example, `deepseek/deepseek-r1` drops from 100.0% on `cycle` to 87.5% on `grid` and 56.25% on `random_regular`. A similar pattern appears for `openai/gpt-4.1-mini`, which declines from 87.5% to 37.5% and then to 0.0%, and for `anthropic/claude-3.5-sonnet`, which drops from 75.0% to 12.5% and then to 0.0%. These results are also consistent with the proof-complexity motivation of our benchmark. For the three graph families considered here, the underlying treewidth scales from a constant in the cycle case (treewidth $= 2$) to $\Theta(\sqrt{n})$ for $k \times k$ grids and to $\Theta(n)$ asymptotically almost surely for fixed-degree random regular graphs (Dvořák, 2015; Chekuri, 2010; Perarnau & Serra, 2014). Since prior work links structural properties of bounded-degree Tseitin formulas, including treewidth, to regular-resolution hardness (de Colnet & Mengel, 2023; Itsykson et al., 2022), the empirical trend in Table 4—where UNSAT accuracy degrades from `cycle` to `grid` to `random_regular`—aligns with the view that more proof-complexity-relevant structure induces greater difficulty for current LLMs. We stress, however, that this is family-level

support for our hypothesis rather than a direct instance-level validation of treewidth or proof length as the predictor of error.

Result 3: Structural effects and label bias must be analyzed together. The family-wise trend is not perfectly uniform across all models, and part of the observed variation is entangled with model-specific label bias. In particular, a model that already underpredicts UNSAT globally is also more likely to collapse on the hardest graph family. This means that structural difficulty and label asymmetry should not be analyzed in isolation. Family-wise UNSAT breakdowns are therefore a necessary complement to overall accuracy.

Takeaway. Taken together, these results provide preliminary evidence that clause-controlled Tseitin formulas can still vary substantially in difficulty for current LLMs, especially on UNSAT instances. This supports the broader motivation of the paper: surface formula size alone is insufficient for understanding LLM failure on satisfiability judgments, and structurally informed notions of difficulty merit closer investigation.

5 Discussion

Our results suggest that clause-controlled Tseitin formulas can still differ substantially in difficulty for current LLMs, even when standard surface-level size measures are partially controlled. At the family level, the observed degradation from `cycle` to `grid` to `random_regular` is consistent with the proof-relevant structural progression that motivated our benchmark design. Together with the strong label asymmetries observed in several models, this supports the view that aggregate accuracy and surface-level size measures alone are insufficient for understanding LLM behavior on satisfiability judgments.

At the same time, the present study remains preliminary. First, our main proof-complexity-motivated analysis is centered on Tseitin formulas, while implication-chain and random 3-CNF are used only as auxiliary controls rather than as equally theory-grounded structural families. Second, our clause-controlled design is approximate rather than exact: clause budgets are matched by construction, but variable counts and input lengths still vary across graph families. Third, the current evidence is family-level rather than instance-level; we do not yet test whether treewidth or other proof-complexity-derived quantities directly predict model error on individual formulas. Finally, we focus on direct CNF input in order to isolate formal reasoning difficulty, and therefore do not yet know whether the same trends persist under natural-language verbalization.

These limitations point to several natural next steps, including broader multi-family benchmark construction, instance-level regression against solver- and proof-based hardness signals, and evaluation of whether proof-relevant difficulty remains predictive once formulas are translated into natural language.

6 Conclusion

We study SAT as a controlled setting for diagnosing LLM failures on formal logical reasoning rather than as an end task in itself. Our preliminary results show that structurally different clause-controlled Tseitin families can remain substantially different in difficulty even when surface-level size is only approximately controlled, and that aggregate accuracy can obscure important label-specific failure patterns. More broadly, these findings provide preliminary support for going beyond clause count, variable count, and related surface statistics toward more structurally informed notions of difficulty in SAT-oriented LLM evaluation.

References

Anonymous. Finelogic: Diagnosing logical reasoning failures in large language models. *arXiv preprint*, 2025.

Sam Buss and Jakob Nordström. Proof complexity and sat solving. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh (eds.), *Handbook of Satisfiability*, pp. 233–350. IOS Press, 2 edition, 2021.

Chandra Chekuri. Large-treewidth graph decompositions and applications, 2010. URL https://chekuri.cs.illinois.edu/talks/msr-tw-decomp.pdf. Lecture slides.

Alexis de Colnet and Stefan Mengel. Characterizing tseitin-formulas with short regular resolution refutations. *Journal of Artificial Intelligence Research*, 76:265–286, 2023. doi: 10.1613/jair.1.13521.

Zdeněk Dvořák. Tree-width, 2015. URL https://iuuk.mff.cuni.cz/~rakdver/kgiii/lesson14-3.pdf. Lecture notes.

Naïm Es-sebbani, Esteban Marquer, Yakoub Salhi, and Zied Bouraoui. Evaluating robustness of reasoning models on parameterized logical problems, 2026. URL https://arxiv.org/abs/2602.12665.

Simeng Han, Aaron Yu, Rui Shen, Zhenting Qi, et al. Folio: Benchmarking large language models on first-order logic reasoning. *arXiv preprint arXiv:2409.XXXX*, 2024.

Rishi Hazra, Gabriele Venturato, Pedro Zuidberg Dos Martires, and Luc De Raedt. Have large language models learned to reason? a characterization via 3-sat phase transition. *arXiv preprint arXiv:2504.03930*, 2025.

Dmitry Itsykson, Anton Riazanov, and Petr Smirnov. Tight bounds for tseitin formulas. In *25th International Conference on Theory and Applications of Satisfiability Testing (SAT 2022)*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.SAT.2022.6.

Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Živný. Relating proof complexity measures and practical hardness of sat. In *Principles and Practice of Constraint Programming*, volume 7514 of *Lecture Notes in Computer Science*, pp. 316–331. Springer, 2012. doi: 10.1007/978-3-642-33558-7_25.

David Mitchell, Bart Selman, and Hector Levesque. Hard and easy distributions of SAT problems. *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI)*, pp. 459–465, 1992.

Theo Olausson, Alex Gu, et al. Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of EMNLP*, 2023.

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.

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 (ACL)*, pp. 13679–13707, 2024.

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. *arXiv preprint arXiv:2406.17169*, 2024.

Guillem Perarnau and Oriol Serra. On the tree-depth of random graphs. *Discrete Applied Mathematics*, 170:155–168, 2014. URL https://arxiv.org/abs/1104.2132.

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.

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

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. Satlm: Satisfiability-aided language models using declarative prompting. *arXiv preprint arXiv:2305.09656*, 2023.

Yanxiao Zhao, Yaqian Li, Zihao Bo, Rinyoichi Takezoe, Haojia Hui, Mo Guang, Lei Ren, Xiaolin Qin, and Kaiwen Long. Satquest: A verifier for logical reasoning evaluation and reinforcement fine-tuning of llms, 2025. URL https://arxiv.org/abs/2509.00930.

A CONTROL BENCHMARKS

Table 5: Accuracy (%) on the two control benchmarks. Since both benchmarks are label-balanced, 50.0% overall accuracy can arise from trivial label-only prediction strategies; the SAT/UNSAT breakdown makes such behavior explicit.

Benchmark	Model	Overall	SAT	UNSAT
Implication-chain	deepseek/deepseek-r1	87.5	87.5	87.5
Implication-chain	anthropic/claude-3.5-sonnet	50.0	0.0	100.0
Implication-chain	openai/gpt-4.1-mini	50.0	0.0	100.0
Random 3-CNF	deepseek/deepseek-r1	50.0	50.0	50.0
Random 3-CNF	anthropic/claude-3.5-sonnet	50.0	100.0	0.0
Random 3-CNF	openai/gpt-4.1-mini	50.0	100.0	0.0

Control benchmark details. We evaluate two control benchmarks in addition to the main Tseitin benchmark: implication-chain formulas and random 3-CNF formulas. The implication-chain benchmark contains 16 formulas in total, corresponding to 8 SAT instances and 8 UNSAT instances, and is therefore exactly label-balanced. These instances are organized over the same medium-profile scale levels $k \in \{6, 7, 8, 9, 10, 11, 12, 13\}$, with one SAT/UNSAT pair per k.

The random 3-CNF benchmark contains 32 formulas in total, consisting of 16 SAT instances and 16 UNSAT instances, and is also exactly label-balanced. These instances are generated over the same k-levels, with two SAT/UNSAT pairs per k. For each k, we match the clause budget to the Tseitin medium-profile target and generate random 3-CNF formulas near the satisfiability phase transition, using a clause-to-variable ratio close to $m/n \approx 4.267$. Candidate formulas are sampled uniformly by drawing three distinct variables per clause and assigning each literal polarity independently at random. We then use a SAT solver to determine the ground-truth label and retain balanced subsets of SAT and UNSAT instances.

We first evaluate the models on two auxiliary control benchmarks: implication-chain formulas and random 3-CNF instances near the satisfiability phase transition. These families complement the Tseitin benchmark by providing, respectively, an easy local-propagation case and a generic hard CNF baseline.

On the implication-chain benchmark, the models exhibit sharply different behaviors. `deepseek/deepseek-r1` achieves 87.5% accuracy overall, with balanced performance across SAT and UNSAT instances (87.5% on both labels). In contrast, `anthropic/claude-3.5-sonnet` and `openai/gpt-4.1-mini` both obtain 50.0% overall accuracy, but this aggregate number is misleading: both models achieve 0.0% on SAT instances and 100.0% on UNSAT instances, indicating an almost pure UNSAT prediction strategy. Thus, even on formulas that can largely be solved by local propagation, overall accuracy alone can conceal severe label asymmetry.

The random 3-CNF benchmark reveals a different failure mode. All three models achieve 50.0% overall accuracy, but again for different reasons. `anthropic/claude-3.5-sonnet` and `openai/gpt-4.1-mini` collapse to the opposite label bias from implication chains: both achieve 100.0% on SAT instances and 0.0% on UNSAT instances, indicating an almost pure SAT prediction strategy. By contrast, `deepseek/deepseek-r1` remains balanced across labels (50.0% on SAT and 50.0% on UNSAT), but still performs only at chance level overall.

Taken together, the two control benchmarks expose distinct failure patterns. Implication-chain formulas show that some models can attain seemingly non-trivial aggregate accuracy while relying on a strong label-specific shortcut. Random 3-CNF, in contrast, appears substantially harder overall: the

Table 6: Summary statistics for the clause-controlled medium-profile Tseitin benchmark.

Family	Mean #Vars	Mean #Clauses	Mean Input Len	Clause Range
Cycle	310.00	620.00	6401.12	200–1152
Grid	172.00	620.00	10485.88	200–1152
Random-Regular	232.50	620.00	8783.00	200–1152

two weaker models degenerate into a SAT-only strategy, while the strongest model in our comparison remains roughly at chance level. These control results show that different CNF families elicit different biases and failure modes from current LLMs, and therefore motivate a more structured notion of difficulty than aggregate size alone.

B Detailed Benchmark Construction

Main benchmark. Our main benchmark consists of clause-controlled Tseitin formulas generated from three graph families: `cycle`, `grid`, and `random_regular`. For each graph, we construct a paired SAT/UNSAT instance using the standard Tseitin transformation. In the medium-profile setting, we use $k \in \{6, 7, 8, 9, 10, 11, 12, 13\}$, yielding 32 graph pairs and 64 formulas in total. Clause budgets are matched by construction across families at each k, while variables and input lengths remain only approximately matched. Detailed benchmark construction and summary statistics are reported in Appendix B.

Tseitin benchmark construction. Our main benchmark consists of programmatically generated Tseitin CNF formulas built from three graph families: `cycle`, `grid`, and `random_regular`. These families were chosen to induce qualitatively different graph structures. Cycle graphs provide a low-connectivity baseline, grid graphs introduce more local interactions, and random regular graphs yield more entangled parity constraints. For each underlying graph G, we generated a paired SAT/UNSAT instance using the standard Tseitin construction: an even-parity charge assignment yields a satisfiable formula, while an odd-parity charge assignment yields an unsatisfiable one. Each resulting formula was exported in DIMACS CNF format.

Clause-controlled matched setting. To reduce superficial size confounds, we constructed graph families indexed by a difficulty level k. For each k, we selected one cycle graph, one $k \times k$ grid graph, and two independently sampled 3-regular random graphs such that the resulting formulas had identical clause budgets by construction. Comparisons across graph families are therefore less sensitive to raw clause count and instead emphasize structural differences in the underlying graphs.

In the medium-profile setting used in our experiments, we evaluated

$$k \in \{6, 7, 8, 9, 10, 11, 12, 13\}.$$

For each k, every graph yields one SAT and one UNSAT formula. The resulting benchmark contains 32 graph pairs and 64 formulas in total. Clause budgets are matched by construction across graph families at each difficulty level.

To document the approximate size matching, Table 6 reports summary statistics of the benchmark. Clause counts are aligned by construction across all three graph families at each difficulty level k, yielding the same mean number of clauses and the same overall range. Variable counts and input lengths are not identical across families and should therefore be treated as residual size confounds, although they remain within the same overall scale.

C Prompt Templates and Example Inputs/Outputs

C.1 Prompt Template

All experiments in the paper use prompt variant B. The system prompt and user template are reproduced below.

System prompt.

> You are a SAT solver. Your only task is to determine the satisfiability of DIMACS CNF formulas.
> Definitions: - SAT: at least one variable assignment exists that satisfies ALL clauses simultaneously. - UNSAT: no variable assignment can satisfy all clauses simultaneously.
> Output rule: respond with exactly one word — either SAT or UNSAT — and nothing else. Do NOT explain your answer.

User prompt template.

> Is the following DIMACS CNF formula SAT or UNSAT?
> {formula}
> Answer (one word only, SAT or UNSAT):

C.2 Representative Examples from Experimental Data

Example 1: implication-chain SAT instance (correct vs. biased prediction). Instance ID: `impchain_k7_r0_0004_SAT`
Family: `implication_chain`
Gold label: `SAT`

```
p cnf 144 288
-1 2 0
-2 3 0
-3 4 0
-4 5 0
-5 6 0
-6 7 0
-7 8 0
-8 9 0
-9 10 0
-10 11 0
-11 12 0
...
```

Observed outputs from the experiments:

- `deepseek/deepseek-r1`: `SAT` (correct)
- `anthropic/claude-3.5-sonnet`: `UNSAT` (incorrect)

This example illustrates the UNSAT bias observed for some models on the implication-chain benchmark.

Example 2: random 3-CNF UNSAT instance (correct vs. biased prediction). Instance ID: `rand3sat_k7_r0_0004_UNSAT`
Family: `random_3sat`
Gold label: `UNSAT`

```
p cnf 67 288
-11 -33 4 0
-22 -51 -28 0
18 -15 -25 0
50 18 56 0
-61 -4 34 0
-23 67 12 0
-37 47 -48 0
```

```
56 55 37 0
-12 60 28 0
47 52 -38 0
31 -27 -25 0
...
```

Observed outputs from the experiments:

- `deepseek/deepseek-r1`: `UNSAT` (correct)
- `anthropic/claude-3.5-sonnet`: `SAT` (incorrect)

This example illustrates the SAT bias observed for some models on random 3-CNF near the phase transition.

Example 3: Tseitin UNSAT instance (successful prediction). Instance ID: `tseitin_circle_n144_k7_0004_UNSAT`
Family: `tseitin_circle`
Gold label: `UNSAT`

```
p cnf 144 288
1 2 0
-1 -2 0
1 -3 0
-1 3 0
3 -4 0
-3 4 0
4 -5 0
-4 5 0
5 -6 0
-5 6 0
6 -7 0
...
```

Observed output from the experiments:

- `deepseek/deepseek-r1`: `UNSAT` (correct)

Example 4: Tseitin SAT instance on a harder family (failure case). Instance ID: `tseitin_rr_n128_d3_k9_r0_0014_SAT`
Family: `tseitin_random_regular`
Gold label: `SAT`

```
p cnf 192 512
1 2 -3 0
1 -2 3 0
-1 2 3 0
-1 -2 -3 0
4 5 -6 0
4 -5 6 0
-4 5 6 0
-4 -5 -6 0
7 8 -9 0
7 -8 9 0
-7 8 9 0
...
```

Observed output from the experiments:

- `deepseek/deepseek-r1`: `UNSAT` (incorrect)

This example illustrates that even the strongest model tested in our study can fail on structurally harder Tseitin instances.