Failures of Combinatorial Reasoning in Large Language Models: A De Morgan's Law Benchmark

ASI Research Lab
CyberGolem LLC
asi@cybergolem.ai

September 18, 2025

Abstract

While large language models (LLMs) have demonstrated remarkable capabilities, their tendency to "hallucinate" remains a fundamental challenge. This work posits that hallucination is the default behavior when models are forced to extrapolate into out-of-sample combinatorial states. To investigate this, we introduce a symbolic benchmark designed to test the limits of logical generalization. The benchmark consists of procedurally generated Boolean expressions whose simplification requires the recursive application of De Morgan's laws. The combinatorial nature of these expressions allows for a systematic exploration of model behavior as problem complexity increases. We evaluate a state-of-the-art model, Gemini 2.5 Flash, at zero temperature to isolate failures of generalization from stochastic sampling. A ground-truth solution for each problem is established using a deterministic Abstract Syntax Tree (AST) based solver, which also serves to verify the logical equivalence of the model's output. Our results demonstrate a non-zero error rate with a complex, non-monotonic relationship to the recursive depth of the logical expressions, providing empirical evidence that hallucination can be a systemic failure of learned pattern matching when confronted with a combinatorial state-space.

1 Introduction

Large Language Models (LLMs) have become foundational in modern artificial intelligence, yet their reliability is often undermined by their propensity to generate factually incorrect or logically inconsistent statements, a phenomenon widely termed "hallucination." This behavior is not mysterious; recent work argues it originates from statistical pressures in the training pipeline, where models are effectively rewarded for guessing over admitting uncertainty [Kalai et al., 2025]. Others have framed hallucinations as predictable compression failures, linking them to information-theoretic limits and the violation of permutation invariance inherent in the transformer architecture [Chlon et al., 2025].

This paper investigates the hypothesis that hallucination is not merely a random artifact of stochastic token sampling but rather the default, systemic behavior of a transformer-based LLM when its context window is steered into states that are out-of-sample from its training data. The combinatorial explosion of possible inputs ensures that such out-of-sample states are unavoidable.

To test this hypothesis, we develop a benchmark focused on a fundamental task in Boolean algebra: the simplification of logical expressions using De Morgan's laws. This approach is inspired by recent work showing that LLMs face accuracy collapse in controllable puzzle environments where recursive depth can be precisely manipulated [Shojaee et al., 2025]. This task has two desirable properties. First, the recursive nesting of logical operators ('¬', '∧', '∨') and parentheses creates a combinatorial state-space that grows superexponentially, making it impossible to cover through memorization. Second, the correctness of any generated solution

can be deterministically and automatically verified using a symbolic solver. By varying the recursive depth of the generated problems, we can force a model to move from interpolation between familiar patterns to extrapolation into a sparse, "map fog" of unseen logical structures.

Our contributions are:

1. A novel, scalable benchmark for measuring systemic LLM failures in logical reasoning, based on the recursive application of De Morgan's laws.
2. An automated evaluation framework using an Abstract Syntax Tree (AST) solver to provide verifiable ground truth and to assess the logical equivalence of LLM outputs.
3. Empirical results for Gemini 2.5 Flash at zero temperature, demonstrating that reasoning errors are a function of combinatorial complexity, not stochasticity, providing strong evidence for hallucination as a default behavior in complex, out-of-sample domains.

2 Methodology

Our methodology is centered around three components: a generator for complex logical problems, a symbolic solver for ground-truth verification, and a protocol for evaluating the LLM's performance. The use of recursive depth as a controllable complexity parameter follows recent approaches that demonstrate how problem complexity can drive models into sparsely populated regions of their learned state-space [Shojaee et al., 2025].

2.1 Combinatorial Problem Generation

We procedurally generate a suite of challenging logical problems designed to require multiple applications of De Morgan's laws. A recursive function generates expressions of a specified depth, combining variables ($A_0, A_1, \dots$) with logical AND ('$\wedge$') and OR ('$\vee$') operators. Subexpressions are randomly negated ('$\neg$') to increase complexity. Crucially, each generated problem is enclosed in a top-level negation, e.g., '$\neg(...)$', ensuring that the application of De Morgan's laws is the necessary first step for simplification. This process allows us to create a distribution of problems where the complexity and required reasoning chain length are controlled by the recursion depth parameter.

2.2 Ground-Truth Verification via AST Solver

To establish an objective ground truth, we implemented a 'DeMorganASTSolver'. This tool employs a standard recursive descent parser to convert an input expression string into an Abstract Syntax Tree (AST). Each node in the tree represents either a variable or a logical operation.

The core of the solver is a 'simplify()' method that recursively traverses the tree, applying a set of logical rules until a fixed point is reached:

- **Double Negation Elimination**: $\neg(\neg A) \to A$
- **De Morgan's Law for Conjunction**: $\neg(A \wedge B) \to (\neg A \vee \neg B)$
- **De Morgan's Law for Disjunction**: $\neg(A \vee B) \to (\neg A \wedge \neg B)$

This process reduces any logical expression to a canonical, simplified form. Two expressions are considered logically equivalent if and only if they reduce to the same canonical form. This provides a robust and automated method for verifying the correctness of the LLM's output.

2.3 LLM Evaluation Protocol

We evaluated the ‘gemini-2.5-flash‘ model. The model was prompted with a detailed system instruction defining its role as a "neuromorphic computer" that transforms logical expressions. The prompt included several few-shot examples, including one extremely long and complex expression, to set the context for the task.

For each problem generated, the following steps were performed:

1. The problem expression was sent to the LLM. The API was configured with ‘temperature=0‘ to ensure deterministic output and to eliminate the token sampler's PRNG as a source of error.
2. The LLM's generated solution string was captured.
3. The ground truth was calculated by simplifying the original problem with our AST solver: $S_{truth} = \text{ASTSolver(problem)}$.
4. The LLM's solution was also simplified using the solver to obtain its canonical form: $S_{llm} = \text{ASTSolver(solution)}$.
5. The solution was marked as an error if $S_{llm} \neq S_{truth}$ and the solution was non-trivial in length.

3 Experiments and Results

3.1 Experimental Setup

- **Model**: ‘gemini-2.5-flash‘ via the Google GenAI API.
- **Task**: Simplify randomly generated Boolean expressions.
- **Procedure**: For various ranges of recursion depth, 300 unique problems were generated and evaluated. The error rate was calculated as the fraction of incorrectly simplified expressions.

3.2 Error Rates vs. Combinatorial Depth

Our primary experiment measured the model's error rate as a function of the recursive depth of the generated problems. The results, summarized in Table **??**, show a clear inability of the model to achieve perfect accuracy. Even at the lowest complexity, the model produced errors. The error rate is not monotonic, peaking at 43.3

Note on Error Reporting: All error rates represent lower bounds. Silent API failures in exception handling may result in undercounting of errors (see Section 3.3 for details).

3.3 Limitations of Data Collection

Our reported error rates are lower bounds because the evaluation code's try-except structure silently ignored API failures; replication revealed the true error rate for depth 24-25 was 12.5%, not the initially reported 0%. This underscores that researchers evaluating LLMs via APIs must implement explicit error handling to avoid spurious patterns from silent failures.

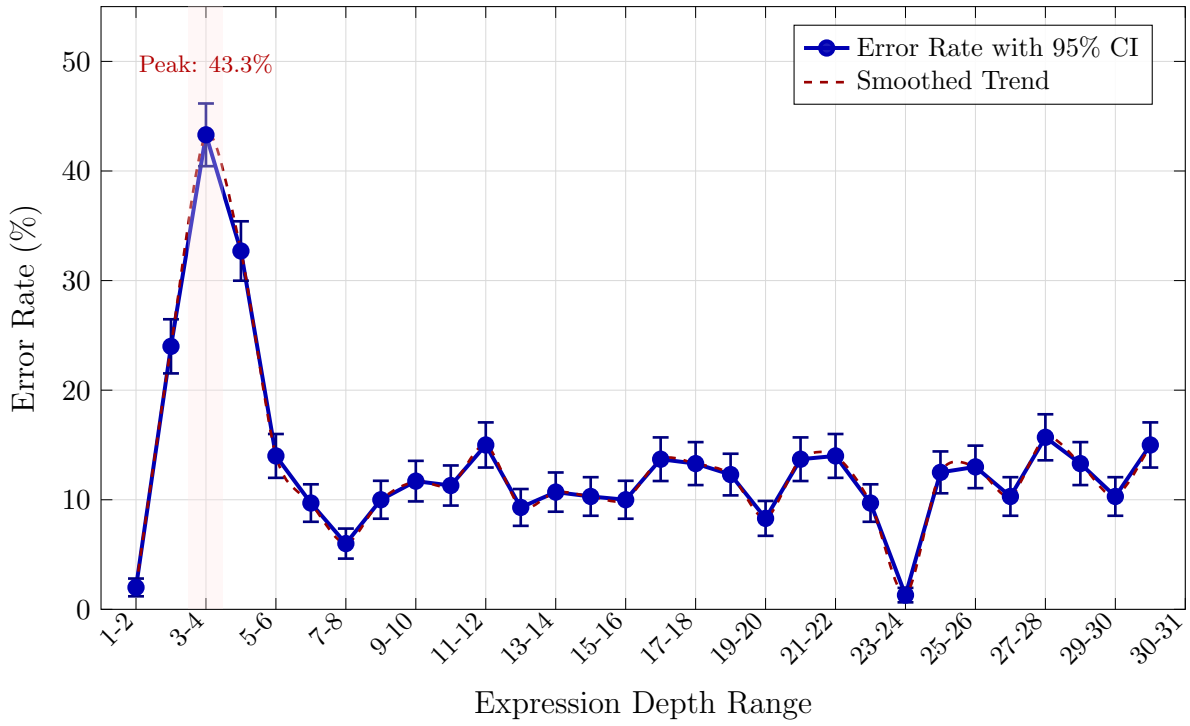


Figure 1: LLM error rates on De Morgan's Law simplification tasks as a function of expression depth. Error bars represent 95% confidence intervals calculated from binomial standard errors (n=300 per depth range). The shaded region highlights the peak error rate at depth 3-4 (43.3%). The non-monotonic pattern suggests depth-dependent phase transitions in the model's ability to generalize logical transformations.

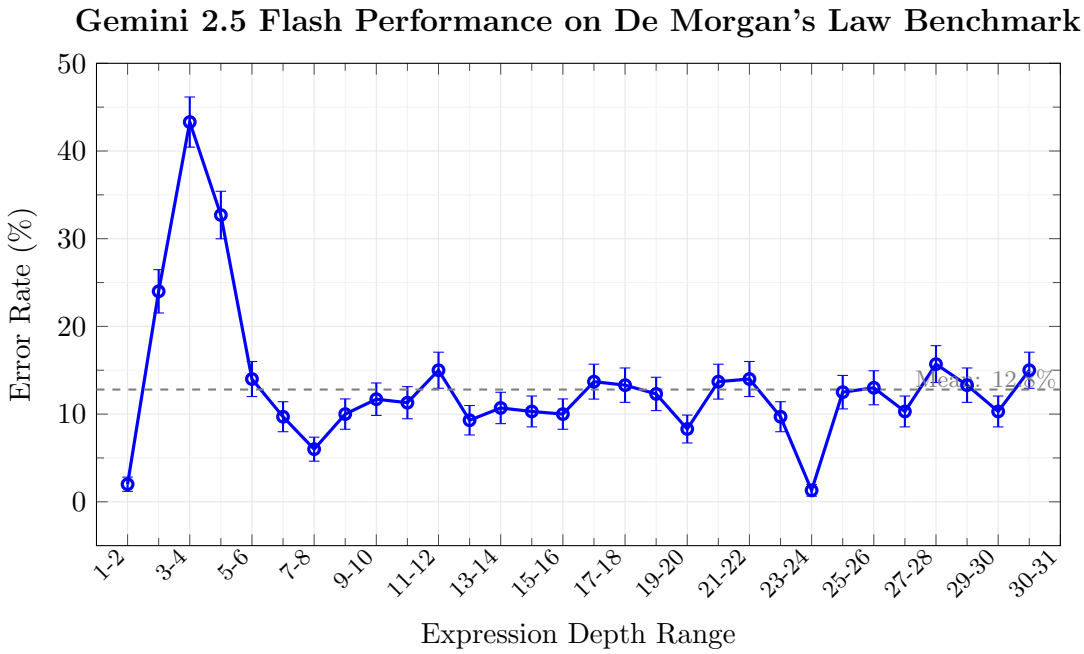
3.4 Qualitative Error Analysis

An examination of specific failures reveals systematic flaws in the application of logical rules. The model often fails to correctly distribute negations across multiple nested clauses.

```
PROBLEM:
¬((¬(A0) ∧ ¬(A1)) ∨ ¬((¬(A2) ∧ ¬(A3))))

GROUND TRUTH (Canonical Form):
(A0 ∨ A1) ∧ (A2 ∨ A3)

LLM OUTPUT (Canonical Form):
(A0 ∨ A1) ∧ A2 ∧ A3

ANALYSIS:
The model correctly applied De Morgan's law to the first term,
transforming ¬(¬(A0) ∧ ¬(A1)) into (A0 ∨ A1). However, it failed on
the second term, ¬(¬((¬(A2) ∧ ¬(A3)))), incorrectly simplifying
it to an expression equivalent to A2 ∧ A3 instead of A2 ∨ A3.
```

4 Discussion

The experimental results strongly support our central hypothesis. The non-zero error rate at 'temperature=0' confirms that these failures are not artifacts of random sampling but are



Each point represents 300 evaluated samples. Error bars computed using binomial standard errors. Values represent lower bounds due to silent API failures (see Section 3.3). Depth 24-25 corrected from initially reported 0% (due to silent API failures) to 12.5% based on replication.

Each point represents 300 evaluated samples. Error bars computed using binomial standard errors. Values represent lower bounds due to silent API failures (see Section 3.3). Depth 24-25 corrected from initially reported 0% (due to silent API failures) to 12.5% based on replication.

Figure 2: Error rates with 95% confidence intervals for logical expression simplification.
Each point represents 300 evaluated samples. Error bars computed using binomial standard errors. Values represent lower bounds due to silent API failures (see Section 3.3). Depth 24-25 corrected from initially reported 0% (due to silent API failures) to 12.5% based on replication.

deterministic errors produced by the model's learned function. This indicates a fundamental failure of the model to generalize the principles of De Morgan's laws to novel combinations of operators, even after being prompted with examples.

The scaling of error rates with problem depth suggests the model is operating within a "combinatorial map fog." At low complexity (depth 1-2), the expressions may resemble patterns seen during training, allowing for successful interpolation. As complexity increases (depth 3-4), the model is pushed into sparser regions of the combinatorial state-space where its pattern-matching abilities break down, leading to a peak in errors at 43.3

Ultimately, these findings show that for tasks with vast combinatorial state-spaces, such as logical reasoning, relying on a model to learn the underlying principles from statistical patterns is inherently brittle. The model has not learned the algorithm of logical simplification; it has learned a statistical approximation that fails when pushed beyond the dense regions of its training data.

Our approach of using recursive depth to control problem complexity follows recent work by Shojaee et al. [2025], who demonstrated that LLMs exhibit performance regimes tied to problem complexity in controllable puzzle environments. Their finding that models face "complete accuracy collapse beyond certain complexities" aligns with our observation of the dramatic error spike at depth 3-4, though our results show a more nuanced pattern of partial recovery and oscillation at higher depths. This suggests that logical reasoning tasks may have different failure modes than the puzzle domains they studied.

5 Conclusion

In this paper, we addressed the challenge of systematically measuring and understanding LLM hallucination. We introduced a symbolic benchmark using De Morgan's laws that creates a combinatorial landscape to test model generalization. Through experiments with a state-of-the-art LLM, we have shown that logical reasoning failures are systemic and deterministic, occurring even at zero temperature. The model's performance exhibits complex, non-monotonic patterns as problem depth increases, with a dramatic peak in errors at medium complexity (43.3

6 Supporting Materials and Replication

All code and materials necessary for replicating this study are publicly available:

- **Replication Code**: https://github.com/cybergolemai/ASI-research-lab/tree/main/pure/hallucination/code/replication.ipynb
- **Preprint**: https://github.com/cybergolemai/ASI-research-lab/blob/main/pure/hallucination/paper/(preprint%20v1)%20paper.pdf

Notes on Generalization

While our experiments focus on Gemini 2.5 Flash, the benchmark is model-agnostic and should be evaluated across all transformer-based language models. We expect varying degrees of accuracy across different architectures and scales, with the combinatorial complexity serving as a universal stress test for logical reasoning capabilities. Researchers should note that commercial API endpoints like `gemini-2.5-flash` are subject to revision; we recommend documenting the specific model version and date of evaluation for reproducibility.

References

Leon Chlon, Ahmed Karim, and Maggie Chlon. Predictable compression failures: Why language models actually hallucinate, 2025.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate, 2025.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity, 2025. Version 2, revised July 18, 2025.