Beyond Surface Forms: Symbolic Edits as a Test for Logical Reasoning with LLMs

Anonymous ACL submission

Abstract

Logical reasoning with large language models (LLMs) is a critical capability, as it reflects a system's ability to correctly deduce hypotheses from a given context using faithful deductive processes. However, LLM reasoning is often sensitive to small variations in problem formulation, raising questions about whether models truly follow the underlying logical structure. Studying this behavior is challenging because the symbolic components of logical problems such as operators and predicates are difficult to systematically manipulate in natural language. We introduce a tool-driven framework for generating controlled, meaning-preserving edits to logical reasoning problems. Our method operates on symbolic representations of first-order logic and constraint satisfaction tasks, enabling targeted modifications to logical operators and other structural components before translating them back into natural language. Using this framework, we evaluate various LLMs under cumulative and individual operator edits and analyze their behavior in response to these changes. Our quantitative and qualitative analysis show that LLM reasoning behavior under controlled operator edits is inconsistent, regardless of the size or family of the models: models sometimes adapt correctly to structural changes but often fail to track their logical consequences. The results from this automated stress test enables an evaluation of the language models on different dimensions and helps measure the reliability of reasoning.[1]

1 Introduction

Deductive logical reasoning is an important testbed for studying the ability of large language models (LLMs) to perform clear, step-by-step inference while following explicitly specified rules. Recent work reports performance saturation on many existing logical reasoning benchmarks (Qi et al., 2025; Deng et al., 2024). These results have motivated growing scrutiny of benchmark reliability, particularly in mathematical reasoning, where studies ask whether reported gains reflect robust reasoning abilities or instead arise from artifacts such as training-distribution effects, benchmark contamination, or leakage (Mirzadeh et al., 2025; Stolfo et al., 2023).

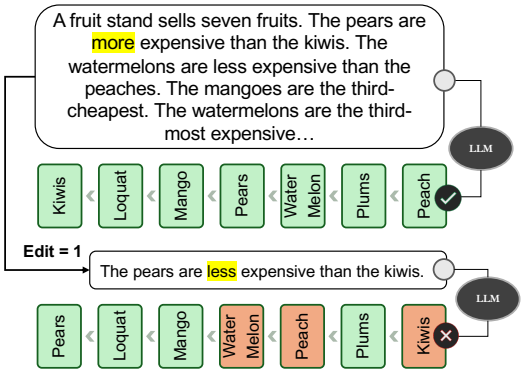

Figure 1: Example from the Deduction task with outputs generated by the `Gemma3-27B` model. Label-preserving edits disrupt the model's reasoning. At edit level 1, the operator changes from *Pears* > *Kiwi* to *Pears* < *Kiwi*, leading the model to incorrectly interpret the final order.

A common paradigm in conducting such studies is to create new versions of problems in existing benchmarks through the replacement of numerical values and entities. While effective for mathematics, where such perturbations often involve simple numerical substitutions (Mirzadeh et al., 2025), logical reasoning depends on structured symbolic components, including operators, predicates, constants, and constraints, that collectively determine the validity of a deduction. Small changes to these components can fundamentally alter both the reasoning process and the outcome. Understanding whether LLMs respond appropriately to such structural changes is therefore critical for evaluating

[1]The code and dataset are available at https://anonymous.4open.science/r/SurfaceForm-BB5E

their capabilities. However, systematically studying this behavior remains difficult. Existing evaluations rely on manually created problem variations, which are costly to produce and difficult to scale (Stolfo et al., 2023). To the best of our knowledge, there is currently no systematic framework for generating controlled modifications to logical reasoning problems that isolate structural changes while preserving semantic validity.

To address this challenge, we introduce a structural analysis framework for logical reasoning that uses tool-driven template generation to produce controlled, meaning-preserving variations of logical deduction problems. The framework decomposes textual reasoning problems into their underlying logical components (using symbolic representations) and enables targeted interventions on specific elements of the logical structure using a tool wrapper. Using this approach, we focus on operator-level edits, allowing us to examine how model performance changes when logical operators are modified while using symbolic systems to track changes in the final answer. Figure 1 shows single variation example of how a change in operator affects the models performance in determining the final order.

We conduct a comprehensive operator-edit analysis along two dimensions: cumulative edits that preserve the final answer, and single edits that can change the ground truth of the problem. We initially evaluate these interventions using final-answer accuracy to capture change in prediction under logically meaningful edits. We measure the performance of several LLMs on deductive logical reasoning and constraint satisfaction tasks, two categories that require reasoning from logical structure rather than external world knowledge. Our experiments stress-test different natural-language reasoning strategies, including standard chain-of-thought prompting (Wei et al., 2022), inference-time scaling (Snell et al., 2024; Wu et al., 2024), and symbolic chain-of-thought reasoning (Xu et al., 2024).

Overall, our study provides a systematic analysis of how LLM reasoning behavior changes under controlled structural interventions. By generating variations of logical reasoning problems, we move beyond traditional final-answer evaluation and examine whether models consistently track the logical consequences of these structural edits. Our results, across recent LLMs with different reasoning strategies, show that model behavior under operator interventions is inconsistent: reasoning changes rapidly under logical modifications and vary unpredictably. Label-changing edits further help us understand the extent to which models fall back to known answers, indicating potential data leakage. In addition, an LLM-based qualitative analysis examines reasoning paths and provides insight into the conditions that cause a drop in reasoning stability. These findings suggest that current language models do not reliably ground their reasoning in the symbolic structure of the problem, underscoring the need for benchmarks that explicitly probe structural reasoning in LLMs.

2 Related Work

Analysis of LLM Behavior Understanding how structural variations in input affect language model behavior has long been a challenging problem (Liao and Vaughan, 2023). Prior work has explored memorization in LLMs through perturbation-based analyses (Xie et al., 2025). While such approaches provide insights into model sensitivity, they do not explicitly control for structural factors across perturbations. In contrast, our work focuses on manipulations at the operator level, enabling a more fine-grained analysis of structural effects.

In logical deduction and stress testing, prior studies have examined how models behave under changes to premise order and paraphrasing (Chen et al., 2024; Bao and Fu, 2025). Other works in mathematical reasoning have used irrelevant context to demonstrate model performance degradation (Shi et al., 2023; Yang et al., 2025). We follow a similar direction in highlighting current gaps in LLM evaluation, but focus on automated template generation to create controlled surface-form variations grounded in symbolic structure.

Controlled Interventions Recent work in mathematical reasoning has explored controlled interventions to study model behavior. Stolfo et al. (2023) demonstrate the feasibility of causal interventions in mathematical reasoning tasks, although their approach is primarily limited to value-level edits. Similarly, Mirzadeh et al. (2025) examine how variations in numerical values within datasets such as GSM8K influence model performance. While these works provide valuable insights into how LLMs respond to changes in input values or task distributions, they often rely on costly human annotation or dataset construction. In contrast, we propose a neuro-symbolic approach to systemati-

cally generate such variations with minimal manual effort.

Reasoning with Deductive Logic Recent surveys highlight the importance of logical reasoning for LLMs (Cheng et al., 2025; Liu et al., 2025). Prior work has studied symbolic generation (Pan et al., 2023), natural-language reasoning (Wei et al., 2022; Thatikonda et al., 2025), and hybrid approaches that avoid external tools (Xu et al., 2024). We focus on natural-language reasoning to evaluate LLM reasoning behavior rather than translation ability.

3 Methodology

3.1 Controlled Intervention

Given the symbolic translations, we intervene on the representations to introduce controlled variations into the questions, using a theorem prover as a wrapper. The types of interventions depend on the structure of the symbolic form. First-order logic representations consist of predicates, operators, and variables, whereas constraint satisfaction problems comprise variables, operators, and values. We focus on modifying operators in both formulations by using symbolic tools. Because such interventions are difficult to apply reliably in natural language, we operate directly on the symbolic representations, which enable precise identification and tracking of edit locations. We perform sentence interventions in two stages: cumulative edits and individual edits.

3.1.1 Cumulative Edits

In the first stage of intervention, we modify the input questions without altering their underlying semantics or final deductive outcomes. A question may undergo zero or more edits, depending on the availability of operator modifications. Each sentence in a question is evaluated to determine whether a valid modification exists. Validity in this case is with an operator edit, ensuring that the final deductive outcome remains unchanged. For example, if the constraint "the blue book is to the right of the yellow book" is changed to "the blue book is to the left of the yellow book," the final solution must remain invariant. Regardless of whether an edit is applied, the process proceeds to the next sentence, and previously modified statements are not revisited. Subsequent edits operate on the already modified sentences, progressively increasing their divergence from the original structure. As the number of edits increases, the structural similarity to the original question decreases. These interventions enable a systematic assessment of model robustness by testing whether reasoning outcomes remain invariant under meaning-preserving transformations.

Due to the cumulative nature of these edits, we use the number of edits, k, as the dependent variable for measuring performance. Since not all questions admit the same number of valid edits, we define K_i as the maximum number of edits available for question i, and evaluate performance using the most recent available edit state for normalized accuracy with a condition to go to a record available at 0 if an edit is not available for the record:

$$\mathrm{Acc}_{\mathrm{norm}}(k) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[\hat{y}_{i,\tilde{k}_i} = y_i\right],$$

$$\tilde{k}_i = \begin{cases} k, & k \le K_i, \\ 0, & k > K_i \end{cases}$$

where N denote the total number of questions, y_i the ground-truth deductive outcome for input i, and $y_{i,k}$ the model's prediction after k edits.

3.1.2 Individual Edits

In the second stage, we apply single alterations that either preserve or change the final outcome. Here, symbolic formulations are handled one at a time, without accumulating changes, and interventions that require a change in the final output have new ground-truth labels stored for each question. We consider two conditions: label-altering and label-preserving. In both cases, the tool wrapper stores valid symbolic representations that either change the final answer or keep it unchanged. Comparing the two conditions helps us assess the extent of potential data leakage[2] and evaluate how well models adapt to familiar questions with different final answers. Since each data point i can have multiple single-edited variations, performance is measured by aggregating results at the question level:

$$\mathrm{Acc}_c = \frac{1}{|\mathcal{N}_c|}\sum_{i\in\mathcal{N}_c}\left(\frac{1}{|\mathcal{R}_i|}\sum_{r\in\mathcal{R}_i}\mathbf{1}\left[\hat{y}_r^{(c)} = y_r^{(c)}\right]\right),$$

$$y_r^{(c)} = \begin{cases} y_i, & c = \text{preserved_label}, \\ y_r^{\text{new}}, & c = \text{altered_label}. \end{cases}$$

[2]Data leakage refers to the assumed condition in which a model behaves similarly with and without label changes. This cannot be verified with certainty, as the training data used for these models is unavailable.

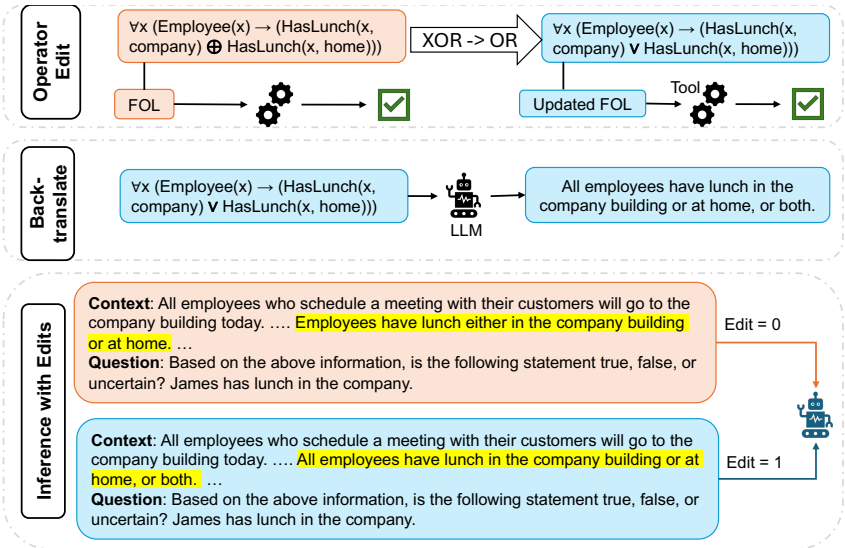

Figure 2: Overview of surface-form edits data extraction. The first step comprises operator edits performed by the tool wrapper, which can be label preserving or altering. In the second step, back-translation by the LLM is conditioned on the valid edited symbolic representation along with the original text. At inference, a second LLM is evaluated to produce outputs that demonstrate the effect of structural changes on LLM performance.

where R_i represents the set of records associated with i, and N_c denotes the set of questions that have valid edits under condition c. The label $y_r^{(c)}$ depends on the condition c to use original label y_i or updated ground-truth label y_r^{new}.

3.2 Back-Translations

Given the altered symbolic forms, the final step is to translate them back into natural-language statements. To reduce complexity and ensure accuracy, we perform back-translation at the sentence level. We use an LLM with a sufficient number of in-context examples to perform back-translation from grounded semantic structures, using the original statements as references. Each translation prompt includes both the altered symbolic form and the original symbolic–text pair, which helps discourage unintended semantic drift. The prompts are designed so that the LLM treats the task as a correction task rather than pure back-translation: we provide the new logical statement and the original sentence, and ask the LLM to revise the sentence so that it matches the new logical statement. For example, to back-translate $blue_book < yellow_book$, we provide the reference sentence 'The blue book is to the right of the yellow book' to encourage the model to preserve the same sentence structure and return 'The blue book is to the left of the yellow book'. After generating all symbolic to natural-language translations, we replace the original statements with their altered counterparts.

Figure 2 shows the complete picture of generation of structural changes.

4 Experiments

4.1 Symbolic Data Extraction

We use three test datasets from (Pan et al., 2023) to evaluate the LLMs: FOLIO (Han et al., 2024), Logical Deduction (Deduction) (Srivastava et al., 2023), and AR-LSAT (Zhong et al., 2022). FOLIO is a first-order logic (FOL) benchmark solved using the Prover9 theorem prover (McCune, 2005); Logical Deduction consists of constraint satisfaction problems (CSPs) solved using a Python-based constraint solver; and AR-LSAT problems are solved using the Z3 solver (De Moura and Bjørner, 2008). For details on translation generation see Appendix A.

4.2 Edits and Translation

Operator edits. For first-order logic representations, such as FOLIO and AR-LSAT, we apply controlled edits to logical operators using the following mappings:

$$\texttt{or} \rightarrow \texttt{and}, \quad \texttt{xor} \leftrightarrow \texttt{or}, \quad \texttt{not} \rightarrow \varnothing$$

These edits introduce structured perturbations that modify the surface form while preserving the final deductive outcome. Additionally, for AR-LSAT, we include the mapping: $\texttt{and} \rightarrow \texttt{or}$. This edit is avoided in FOLIO, because conjunctions are often used to define properties or states of an object. For example, a statement such as

$\text{Game}(x) \wedge \text{LegendOfZelda}(x)$ indicates that *The Legend of Zelda* is a game; replacing $\wedge$ with $\vee$ would make the logical statement semantically ill-formed or substantially alter its meaning. Similarly, we avoid other operator edits that could affect the inherent meaning of the sentence rather than introduce a controlled structural variation.

Edits are applied iteratively. For each question, we scan FOL premises sequentially, apply a single operator substitution, and then pass the modified problem to the theorem prover. To avoid cycling over complementary edits, we proceed to the next premise after each successful modification and continue until no further valid edits can be identified.

For CSP, we apply an analogous procedure over constraints rather than premises, using the following operator swaps:

$$\texttt{lessthan} \leftrightarrow \texttt{greaterthan}, \quad \texttt{equal} \leftrightarrow \texttt{notequal}.$$

An edit is considered valid if the CSP or FOL solver returns a value from the solution set. The number of validated operator edits per question is recorded separately for each symbolic representation.

Back-translations Altered symbolic forms are translated back into natural language using `Gemini 2.5 Flash` with task-specific instructions and five in-context examples per dataset (See Appendix B). The input consists of (i) the original natural-language statement and (ii) the altered symbolic form. The model is instructed to minimally edit the statement to reflect the modified symbolic representation, preserving sentence structure and style. The resulting sentences replace the originals, yielding an expanded evaluation set (see Appendix C) with an additional *edit count* parameter for analysis.

4.3 Reasoning Inference

We evaluate reasoning performance under cumulative edits using Chain-of-Thought (CoT), CoT with test-time scaling (Scaling), and symbolic CoT (SymbCoT) prompting across the `Gemma3` family (1B–27B) (Kamath et al., 2025), `LLaMA3-8B` (Grattafiori et al., 2024), `Phi-4-Mini` (Abouelenin et al., 2025), and `Qwen3-4B` (Team, 2025). For individual edits, we use CoT to analyze model behavior. For prompting details and configurations, see Appendix D.

	k	Gemma3-27B			Qwen3-4B		
		CoT	Scaling	SymbCoT	CoT	Scaling	SymbCoT
FOLIO	0	81.20	82.91	76.07	88.03	87.18	76.92
	1	82.05	85.47	78.63	87.18	88.03	76.92
	2	82.05	84.62	76.07	84.62	84.62	73.50
Deduction	0	84.33	93.00	86.00	96.00	95.33	74.00
	1	85.67	93.67	85.67	95.67	94.33	73.33
	2	84.67	94.33	86.00	95.67	**92.33**	73.67
	3	85.33	93.00	84.67	95.33	**92.67**	71.33
	4	84.67	93.33	85.67	95.67	95.00	73.67
AR-LSAT	0	32.61	36.52	28.70	84.78	36.52	17.83
	1	33.04	37.39	30.43	83.04	35.65	19.13
	2	32.17	**39.57**	27.83	82.17	35.22	17.39
	3	34.35	36.96	28.70	83.48	36.09	16.52

Table 1: Normalized Accuracy (%) versus the number of label-preserving operator edits across datasets for Gemma 27B and Qwen 4B on valid records. Accuracy is computed over the corresponding total count, and **bold** indicates statistically significant performance changes, either gains or drops, relative to the base performance.

5 Results and Discussion

5.1 Cumulative Edits

As the maximum number of valid edits (k) varies across logical questions, records without k valid edits fall back to their inference at baseline ($k = 0$), ensuring that accuracy is normalized with respect to the total number of questions. Thus, the accuracy at k is compared with the baseline ($k = 0$). Table 1 shows a representative snippet of these results, indicating a minimal impact, with most configurations remaining stable across edit levels. These preliminary findings suggest that operator edits neither systematically harm nor improve model reasoning. Additionally, across the three reasoning methods, SymbCoT often under-performs relative to the other prompting strategies. The full set of results is presented in the Appendix E.

We observe that this minimal impact arises because transitions in record-level predictions tend to balance out in the final performance, as shown in Figure 3. Although the large Gemma model shows stable performance in terms of accuracy, this metric does not capture the full extent of the model's behavior. Rather than exhibiting uniform gains or drops in final accuracy, model behavior varies at the instance level as inputs undergo successive transformations. To better examine the LLMs, we analyze reasoning stability using flip rate.

Flip Rate We measure the extent to which predictions change after edits, even when the underlying semantics are preserved. Since models behave inconsistently under edit-induced changes, we ana-

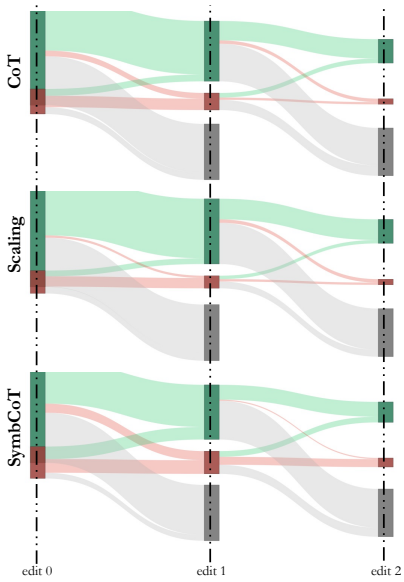

Figure 3: Transitions for FOLIO with the `Gemma3-27B` model across three forms of reasoning. Green indicates correct predictions, red indicates incorrect predictions, and grey denotes records with no further edits. Despite the insignificant change in accuracy reported in Table 1, records continue to transition between correct and incorrect predictions across edit levels.

lyze flips in both directions: from correct to incorrect and from incorrect to correct. We compute the overall flip rate as the number of instances whose correctness changes, divided by the total number of instances:

$$\text{FlipRate} = \frac{n_{\text{I}\to\text{C}} + n_{\text{C}\to\text{I}}}{n_{\text{I}\to\text{C}} + n_{\text{C}\to\text{I}} + n_{\text{I}\to\text{I}} + n_{\text{C}\to\text{C}}},$$

where $n_{\text{I}\to\text{C}}$ denotes the number of inferences from a given LLM that change from incorrect to correct, $n_{\text{C}\to\text{I}}$ denotes the number that change from correct to incorrect, and $n_{\text{I}\to\text{I}}$ and $n_{\text{C}\to\text{C}}$ denote inferences whose correctness remains unchanged.

To measure significance, we define the null hypothesis as follows: a model's reasoning outcome remains unchanged after editing. The alternative hypothesis states that a model exhibits a significant transition in correctness with respect to the original question. We treat changes greater than (5%) as practically meaningful and apply a Bonferroni correction when estimating (p)-values to account for multiple models for a given edit.

Table 2 reports the proportion of instances whose inference correctness changes after editing. We compare these results against the accuracy trends reported earlier. To better interpret the results, we analyze them by model size and dataset type.

Smaller Gemma models, particularly Gemma 1B, are highly unstable across datasets when using CoT prompting. A notable finding is that inference-time scaling substantially stabilizes smaller models on FOLIO and AR-LSAT, suggesting that this form of reasoning can benefit lower-capacity models. Among the 4B–12B models, Qwen shows lower flip rates on FOLIO and Deduction, whereas Gemma 4B shows lower flip rates on AR-LSAT. However, Qwen models also achieve the highest accuracy on AR-LSAT (Table 1), indicating that strong final-answer performance can still coexist with reasoning instability.

Gemma 27B, in contrast, shows higher stability when its overall performance is high, and lower stability when its overall accuracy is very low, as in AR-LSAT. This contrasts with the behavior of Qwen, even though the two models, despite differing substantially in size, show similar performance trends. Across the full analysis, SymbCoT exhibits mixed behavior, but overall tends to produce relatively high flip rates across datasets. This is likely due to error propagation introduced across the multiple prompting stages required to derive the final answer.

Qualitative study We conduct a qualitative study to examine the reasoning patterns produced at each edit level. We use coding criteria that assess the overall correctness of the reasoning path for logical deductive reasoning. The qualitative coding is performed across four levels, two of which evaluate the reasoning process and two of which evaluate the final answer. The criteria are designed to answer the following questions:

- *Premise Fidelity (PF):* Does the reasoning use only the information provided in the original question?
- *Inference Validity (IV):* Do the reasoning steps logically follow from the given premises and from prior reasoning steps?
- *Answer Completion (AC):* Does the reasoning include the necessary intermediate steps needed to support the conclusion?
- *Answer Alignment (AL):* Is the final answer supported by the reasoning?

We run the coding procedure using the `Gemini 2.5 Flash` model with a curated prompt (see Appendix G) designed to produce structured outputs. We apply this procedure to a subset of records across all datasets, selecting records that contain the maximum number of edits. Each criterion receives a binary score, and the overall reasoning score is computed as the mean of these criterion-

	Method	Transition	Gemma3-1B	Gemma3-4B	Gemma3-12B	Gemma3-27B	Llama3-8B	Qwen3-4B	Phi4Mini-4B
FOLIO	CoT	0→1	**41.18**	19.12	19.12	16.18	**23.53**	10.29	**29.41**
		0→2	23.08	19.23	**34.62**	26.92	26.92	23.08	**46.15**
	Scaling	0→1	2.94	13.24	2.94	10.29	14.71	7.35	16.18
		0→2	3.85	15.38	30.77	23.08	19.23	11.54	26.92
	SymbCoT	0→1	**29.41**	**33.82**	**33.82**	**27.94**	8.82	14.71	22.06
		0→2	30.77	23.08	**46.15**	23.08	15.38	**46.15**	19.23
Deduction	CoT	0→1	**19.35**	**31.45**	9.68	9.68	**23.39**	4.03	**22.58**
		0→2	**23.36**	**33.64**	14.95	12.15	**26.17**	2.80	**28.97**
		0→3	20.48	**39.76**	9.64	8.43	**25.30**	2.41	20.48
		0→4	**28.00**	**30.00**	8.00	6.00	22.00	2.00	**26.00**
	Scaling	0→1	12.90	16.13	2.42	4.84	8.06	4.03	11.29
		0→2	12.15	18.69	4.67	5.61	3.74	10.28	8.41
		0→3	19.28	18.07	0.00	0.00	3.61	9.64	3.61
		0→4	**28.00**	22.00	0.00	2.00	8.00	6.00	6.00
	SymbCoT	0→1	8.87	12.10	12.10	4.03	**20.16**	14.52	17.74
		0→2	5.61	17.76	**20.56**	7.48	**20.56**	19.63	**26.17**
		0→3	7.23	**25.30**	15.66	4.82	**26.51**	19.28	**30.12**
		0→4	10.00	22.00	20.00	2.00	**26.00**	18.00	24.00
LSAT	CoT	0→1	**15.31**	**13.27**	**23.47**	**21.43**	**29.59**	**18.37**	**17.35**
		0→2	**23.21**	**14.29**	**23.21**	**23.21**	**19.64**	**21.43**	**26.79**
		0→3	**15.38**	11.54	**42.31**	**23.08**	**15.38**	11.54	**26.92**
	Scaling	0→1	8.16	8.16	**10.20**	**14.29**	**10.20**	**12.24**	**13.27**
		0→2	5.36	5.36	7.14	**16.07**	**12.50**	**16.07**	**25.00**
		0→3	11.54	0.00	**23.08**	3.85	7.69	11.54	**15.38**
	SymbCoT	0→1	**15.46**	**15.31**	**32.65**	**32.65**	**10.20**	9.18	**22.45**
		0→2	**25.00**	**14.29**	**37.50**	**28.57**	**12.50**	8.93	**19.64**
		0→3	**15.38**	7.69	**23.08**	**38.46**	0.00	11.54	**15.38**

Table 2: Flip Rate: Total Percentage of records that transition from correct to incorrect and from incorrect to correct as surface forms change from edit 0 to later edits. Higher values indicate greater instability across edits. Values in **bold** indicate significance of transitions across all models.

Edits	PF	IV	AC	AL
0	83.33	75.00	91.67	87.50
1	79.17	70.83	79.17	75.00
2	70.83	66.67	75.00	83.33
3	75.00	58.33	70.83	62.50

Table 3: Qualitative scores (%) by the number of label-preserving edits for the Qwen3-4B model on the AR-LSAT dataset. The results show a decline in reasoning metrics as the number of edits increases.

level scores. To assess whether these codes are reliable, we compute the correlation between final-answer accuracy and the overall reasoning score. The two are highly correlated, with a score of 0.9, providing some confidence in analyzing the individual criterion scores.

For brevity, we examine one AR-LSAT example using the `Qwen3-4B` model, where accuracy is high but stability is questionable. Table 3 summarizes qualitative reasoning robustness under label-preserving edits. We report the overall reasoning percentage for each edit level. We observe that answer alignment, which measures whether the final answer is supported by the reasoning, decreases as question variation increases, even though accuracy does not decrease. This points to the need to examine reasoning paths more deeply rather than relying only on accuracy scores.

The model is also prompted to provide a justification whenever it assigns a score of 0 for any criterion. We analyze negatively connoted words in these justifications across all models and identify several recurring patterns that may indicate reasoning issues in LLMs. Among the frequently observed relevant tokens, we find *distorts*, *misinterprets*, and *invents* for premise fidelity; *invalid*, *unsupported*, and *leap* for inference validity; *fails*, *skips*, and *incomplete* for answer completion; and *flawed* and *contradicts* for answer alignment. We provide keyword list, representative justifications, and additional scores in Appendix H.

5.2 Individual Edits

So far, we have focused on edits that do not change the final answer, which raises the question of whether models rely on pattern matching to recover the final answer. Individual edits, together with explicit tracking of the final answer, allow us to investigate this behavior. Since each record can have multiple edited instances, we use grouped accu-

	Condition	Gemma3-1B	Gemma3-4B	Gemma3-12B	Gemma3-27B	Llama3-8B	Qwen3-4B	Phi4-4B
FOLIO	Original	−5.59	2.38	5.93	3.55	5.29	−0.12	−0.47
	Altered	**-27.33**	**-29.33**	−16.00	**-23.33**	−21.33	**-22.67**	−4.67
Deduction	Original	−4.00	−2.91	3.40	2.79	−4.05	−0.68	3.13
	Altered	2.87	−0.32	**-7.70**	**-12.95**	−8.65	−0.32	−8.23

Table 4: Difference in model accuracy(%) under operator mutation with and without alteration. **Bold** values indicate statistically significant differences. Negative values represent drops in accuracy, whereas positive values represent gains. The original condition refers to the existing label, while the altered condition refers to the tool-edited label.

Model	FOLIO (%)	Deduction (%)
Gemma3-1B	**36.23**	**24.00**
Gemma3-4B	**30.43**	13.14
Gemma3-12B	**28.99**	17.14
Gemma3-27B	**27.54**	13.14
Llama3-8B	**42.03**	16.00
Qwen3-4B	15.94	3.43
Phi4Mini-4B	**23.19**	**18.86**

Table 5: Memorisation rates by model and dataset. Bold values indicate statistically significant memorisation at $p < 0.05$ using a 10% threshold.

racy to measure the effect of single edits, both with and without changes to the final answer. Specifically, we measure the difference in performance between edited records and the model's original performance on the corresponding unedited records. For a fair comparison, we also apply single edits that do not change the final answer and compare model behavior across these two conditions. We assess statistical significance using a pairwise test.

Table 4 reports the performance changes between edit level 0 and the grouped single-edit conditions.[3] For deduction tasks with premise–hypothesis structures, such as FOLIO, performance consistently drops when the label changes, with large declines observed across high-performing models. Although the larger Gemma models generally achieve high accuracy on logical deduction, their performance decreases when labels are changed through small operator-level edits. This pattern may indicate that the model has been exposed to the test dataset, or that its high original performance depends on surface-level familiarity rather than robust reasoning. We explore this further by analyzing model prediction behavior.

Prediction Behavior We measure how often a model falls back to the original answer instead of predicting the edited answer. Table 5 reports this behavior. We define the ideal case as one in which the model does not output the original answer after the label-changing edit. The alternative hypothesis is that the model outputs the original answer for more than 10% of edited records.

Except for Qwen, all models show some level of memorization on FOLIO. Since Qwen previously showed a significant performance drop under label-changing edits, its behavior may be attributed to reasoning difficulty rather than memorization. The Logical Deduction dataset also shows evidence of memorization among lower-performing models; however, as model size increases, the memorization rate decreases. Under a stricter 5% threshold, all models except Qwen show signs of memorization.

6 Conclusion

Understanding how LLM reasoning varies with changes to surface form is key to assessing the extent to which models follow logical structure. In this work, we present a tool-driven framework that generates controlled, meaning-preserving edits by operating on symbolic representations of logical problems. This enables precise operator-level interventions while preserving the original deductive outcome. Using this framework, we analyze LLM behavior under symbolically grounded variations of existing records. Our results show that models reason inconsistently under these edits and may exhibit signs of data leakage in evaluation sets. These findings suggest that current LLMs do not reliably ground their reasoning in symbolic structure, highlighting the need for evaluation metrics beyond standard accuracy.

Overall, our evaluation provides a controlled way to stress-test potential data leakage and assess robustness to structural variation. Future work could use stability under such variations as a training or evaluation signal, helping reduce reliance on pattern matching. The proposed template-generation framework may also support the construction of more diverse benchmarks through symbolic editing, exposing models to broader reasoning patterns during training and evaluation.

[3] We exempt AR-LSAT because only 22 records undergo label-changing edits (too few for reliable estimation).

Limitations

Our work is subject to the following limitations.

- **Tool dependency and translation coverage.** The methodology is contingent on the availability of translations supported by the tool used. When translations are unavailable for a given problem, the analysis cannot be performed as-is, since automated translation forms the foundation of our approach. Our study uses a multiple pass at translation for logical deduction and ar-lsat datasets. We rely on availability of parsable translations for datasets such as FOLIO, which limits the number of translations. Furthermore, even where translations are available, semantic errors may remain latent; these may not surface during tool-based inference if they do not affect the underlying reasoning chain.

- **Back-translation fidelity.** The back-translation from first-order logic to natural language is constructed to be as semantically transparent as possible, with explicit constraints provided to the language models. Although a manual inspection by the authors revealed no errors, the non-deterministic nature of language models (we use Gemini 2.5 Flash) means that translation errors may have been introduced. As our work is evaluation-based and measures model reasoning ability after back-translation with at least one model solving these back-translated questions correctly, we expect any such errors to have a minimal effect on the overall results.

- **Reproducibility with proprietary models.** Our study makes use of proprietary language models, which are non-deterministic and subject to undisclosed version updates, limiting exact replication of results. To mitigate this, we report results for open-source models with explicit version identifiers, which can be used to reproduce our findings.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Jun-Kun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, et al. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *CoRR*, abs/2503.01743.

Qiming Bao and Xiaoxuan Fu. 2025. Less is more for multi-step logical reasoning of LLM generalisation under rule removal, paraphrasing, and compression. *CoRR*, abs/2512.06393.

Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise order matters in reasoning with large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 6596–6620. PMLR / OpenReview.net.

Fengxiang Cheng, Haoxuan Li, Fenrong Liu, Robert van Rooij, Kun Zhang, and Zhouchen Lin. 2025. Empowering llms with logical reasoning: A comprehensive survey. *arXiv preprint arXiv:2502.15652*.

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

Shujie Deng, Honghua Dong, and Xujie Si. 2024. Enhancing and evaluating logical reasoning abilities of large language models. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana

Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. 2025. Gemma 3 technical report. *CoRR*, abs/2503.19786.

Q. Vera Liao and Jennifer Wortman Vaughan. 2023. AI transparency in the age of llms: A human-centered research roadmap. *CoRR*, abs/2306.01941.

Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. 2025. Logical reasoning in large language models: A survey. *arXiv preprint arXiv:2502.09100*.

William McCune. 2005. Release of Prover9. In *Mile high conference on quasigroups, loops and nonassociative systems, Denver, Colorado*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 3806–3824, Singapore.

Chengwen Qi, Ren Ma, Bowen Li, He Du, Binyuan Hui, Jinwang Wu, Yuanjun Laili, and Conghui He. 2025. Large language models meet symbolic provers for logical reasoning evaluation. In *The Thirteenth International Conference on Learning Representations*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 31210–31227. PMLR.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Trans. Mach. Learn. Res.*, 2023.

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schölkopf, and Mrinmaya Sachan. 2023. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 545–561. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Ramya Keerthy Thatikonda, Wray Buntine, and Ehsan Shareghi. 2025. Logical reasoning with outcome reward models for test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 26113–26123.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. An empirical analysis of compute-optimal inference for problem-solving with language models. *CoRR*, abs/2408.00724.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. 2025. On memorization of large language models in logical reasoning. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL 2025, Mumbai, India, December 20-24, 2025*, pages 2742–2785. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13326–13365. Association for Computational Linguistics.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Yang Wang, and Liangming Pan. 2025. How is LLM reasoning distracted by irrelevant context? an analysis using a controlled benchmark. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 13329–13347. Association for Computational Linguistics.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Yining Chen, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2022. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, Findings of ACL, pages 2306–2319. Association for Computational Linguistics.

A Symbolic Translations

We extract symbolic translations based on the dataset and the availability of existing translations. For Logical Deduction and AR-LSAT, symbolic representations are generated following the two-shot prompting setup of Pan et al. (2023), adapted to a structured JSON format. This design allows us to enforce formatting constraints and guide the model toward producing well-formed outputs. Unless otherwise specified, all symbolic representations are generated using `Gemini 2.5 Flash` with a temperature of (0.1) under batched inference. The final dataset statistics are summarized in Table 6.

- **FOLIO:** FOLIO provides ground-truth symbolic representations that do not always follow the tool format. We first run these through Prover9 and manually edit a set of formulations to correct annotation errors. These errors are identified by inspecting tool feedback. The corrected symbolic forms are retained for approximately 10% of the records.
- **Logical Deduction:** Logical Deduction is relatively easy to translate, with the model achieving a 99% success rate on the first generation pass. We run the model twice with a temperature of (0.1) to obtain all 300 translations suitable for execution using a Python-based CSP tool.
- **AR-LSAT:** AR-LSAT is more challenging to translate into a form suitable for Z3. We initially use a temperature of (0.1) and then increase decoding randomness. We find that test-time scaling with a temperature of (0.7), an incremental scaling factor of 4, and up to 8 samples yields a larger number of valid translations. This incremental scaling strategy reduces inference cost by 12.7% compared to directly generating 8 samples for every record. Using `Gemini 2.5 Flash`, we extract translations for approximately 88% of the records. For the remaining records, we use `Gemini 3 Flash Preview` with diverse decoding strategies by varying temperature, top-(k), and top-(p) values. We note that self-correction does not improve translation quality over this inference-only approach with diverse decoding strategies. Overall, we obtain 224 translated records. Among these, we identify 6 records with incorrect final labels, which the authors manually correct by solving the questions and verifying the answers with multiple LLMs. These corrections are reported in Table 7 and are used as the ground truth for all results.

Dataset	Tool	Valid
FOLIO	Prover9	117
Deduction	Python	300
AR-LSAT	Z3	224

Table 6: Statistics of datasets showing the number of records with valid translations

ID	Given	Corrected
`ar_lsat_201612_3-G_1_3`	B	D
`ar_lsat_201612_3-G_1_5`	E	B
`ar_lsat_201612_3-G_2_6`	C	D
`ar_lsat_201612_3-G_2_8`	A	B
`ar_lsat_201612_3-G_3_12`	C	A
`ar_lsat_201612_3-G_3_17`	D	E

Table 7: Corrections to AR-LSAT answer labels. The *Given* column shows the original label, and the *Corrected* column shows the revised label.

B Back-Translations

We apply back-translation at the sentence level by passing the original natural-language sentence as the incorrect statement and prompting the model to generate a corrected natural-language sentence without deviating from the semantic structure of the question. Figures 4, 5, and 6 provide examples of the prompts used for AR-LSAT, Logical Deduction, and FOLIO, respectively.

The Logical Deduction dataset involves simple alterations; however, the semantic variation induced by these changes is not easily programmable. For example, changing ‘less than” to ‘greater than” may correspond to different interpretations, such as older vs. newer or left vs. right, making it difficult to generalize such transformations.

C Edit Stats

After applying cumulative and single edits, we perform back-translation of the logical forms using the corresponding natural-language text provided with each logical form. In some cases, the natural-language sentence associated with a logical form does not appear as a standalone sentence in the original question. This occurs, for example, in AR-LSAT, where a statement such as “there are four objects” may be embedded within a longer sentence rather than appearing independently. In other cases, the LLM performing back-translation fails to generate a valid replacement, leaving the final question unchanged.

After eliminating cases that do not support the required back-translate-and-replace procedure, fewer

Dataset	Cuml	Ind-original	Ind-altered
FOLIO	94	68	50
Deduction	364	123	157
AR-LSAT	180	-	-

Table 8: Total number of edits available for each dataset under two edit settings. Individual edits include both label-preserving edits and label-altering edits. AR-LSAT does not generate enough records to support a meaningful individual-edit analysis.

records remain. For cumulative edits, if a translation is unavailable at edit level (k), all subsequent edits are discarded, and the record is retained only up to edit level ($k-1$). The final counts of edited records are reported in Table 8.

D Reasoning Prompts

We use the following system prompt for CoT and scaled CoT to obtain structured outputs: 'You are a reasoning assistant that reasons step by step, and puts your final answer within \boxed{}.' The user instruction explicitly prevents the use of external knowledge: 'Solve the given logical deductive reasoning problem using only the information provided in the question. Do not use or rely on any external knowledge.'

For Symbolic CoT, we use the prompts from the original paper to perform the three stages of inference. The only modification is that we ask the model to generate JSON output, which allows us to extract the required inference results more easily. Apart from this formatting change, the examples and instructions remain the same.

The configurations for the models used to generate inferences are detailed in Appendix I.

E Cumulative Performance

We present the performance of other open-source models on cumulative edits in Table 9. Performance is measured at each edit level with respect to the base question. If an edit is not available for a given input, we fall back to the base question, ensuring that the total number of evaluated records remains constant across edit levels. The number of records affected by each edit level is reported as *Count* in the table.

F Single Edits

We present the accuracy results for all single edits in Table 11. Accuracy differences are computed as the difference between base accuracy and edited-record accuracy. Labels are either preserved or changed using the tool wrapper. The number of base records represents the number of unique records with valid edits among the total available translations.

Label-changing edits are fewer than label-preserving edits, since extracting a completely new label from a given question often requires substantially different reasoning. This new reasoning requirement may also explain the drastic drop in accuracy under altered labels. Overall, this provides a useful way to evaluate existing datasets that have reached near-saturated performance and to test whether models are still reasoning correctly under controlled structural changes.

G Qualitative Analysis- Prompt

In the qualitative analysis, we examine model reasoning patterns in a generated CoT setting. For logical deduction tasks, the reasoning process typically proceeds in three steps: first, the model identifies the premises provided in the question; second, it derives intermediate inferences from these premises when necessary; and third, it checks whether one of the answer options is supported by the reasoning. We also examine the model's final answer claim, especially the answer placed in ($\square$), as requested in the prompt.

Based on this observed reasoning process, we divide the qualitative analysis into four components. We define the coding criteria as Premise Fidelity (PF), Inference Validity (IV), Answer Completion (AC), and Answer Alignment (AL). Each code is designed to measure a specific aspect of the reasoning process. These criteria are described in more detail in the evaluation prompt shown in Figure 7. The LLM evaluator receives both the original question and the generated reasoning path as input.

This analysis depends on the verification capability of the evaluator LLM. Therefore, the results should be interpreted as estimated judgments rather than conclusive findings, and further study is needed to validate these observations.

H Qualitative Analysis- Findings

Table 10 presents the full set of LLM-evaluated qualitative results for records with the maximum number of edits in each dataset. Smaller models generally show low premise fidelity and inference validity. Models such as Gemma3-1B also exhibit very low answer completeness, indicating missing

	Edits	Gemma3 1B			Gemma3 4B			Gemma3 12B			Llama3 8B			Phi4mini			Count
		CoT	Scaling	SymbCoT	CoT	Scaling	SymbCoT	CoT	Scaling	SymbCoT	CoT	Scaling	SymbCoT	CoT	Scaling	SymbCoT	
FOLIO	0	47.01	41.03	35.90	71.79	74.36	51.28	80.34	89.74	71.79	58.12	70.94	44.44	68.38	72.65	24.79	117
	1	43.59	41.03	41.03	74.36	75.21	55.56	79.49	89.74	74.36	59.83	70.94	42.74	68.38	71.79	22.22	68
	2	47.01	41.88	39.32	69.23	74.36	52.99	81.20	89.74	70.09	60.68	70.09	44.44	64.96	**66.67**	25.64	26
Logical Deduction	0	21.67	26.33	6.33	37.00	39.33	50.00	82.00	91.33	76.00	36.33	39.33	42.00	48.00	58.67	38.00	300
	1	20.67	26.33	8.00	35.33	39.33	49.00	81.33	91.67	77.00	35.00	39.00	44.00	48.67	59.67	37.33	124
	2	19.33	28.00	7.67	36.33	39.33	52.33	80.00	91.00	78.00	36.00	40.00	44.67	47.67	61.67	33.00	107
	3	21.67	27.67	7.67	35.33	40.33	52.67	83.33	91.33	77.67	35.67	39.33	44.00	48.33	61.33	35.67	83
	4	21.67	27.00	6.67	37.33	39.67	50.67	83.33	91.33	78.00	35.67	39.33	44.00	47.67	61.67	38.00	50
	5	**24.33**	28.67	6.67	36.67	38.67	50.67	82.67	91.33	77.67	36.00	39.33	41.67	49.00	**62.67**	39.33	41
AR-LSAT	0	22.61	20.00	12.17	15.22	17.83	13.91	23.91	32.61	26.09	17.83	20.87	13.91	14.35	19.13	16.09	230
	1	21.30	22.61	10.87	14.78	18.70	14.35	23.48	33.48	22.61	15.65	20.00	13.04	14.35	20.43	16.52	98
	2	23.04	20.43	11.30	16.09	18.26	13.91	23.48	33.48	23.04	17.83	21.30	13.91	15.22	21.30	16.52	56
	3	22.61	21.30	12.17	15.65	17.83	14.35	21.74	33.48	25.22	16.96	20.87	13.91	14.78	19.57	16.96	26

Table 9: Backfilled accuracy (%) versus the number of label-preserving operator edits across datasets and models. We report results for Gemma 1B, Gemma 4B, Gemma 12B, Llama 8B, and Phi4mini. **Bold** values indicate statistically significant records. Count denotes the number of records corresponding to each edit level.

or incomplete final answers, as well as low answer alignment, which is correlated with answer completion. Cases with high answer alignment but low answer completeness suggest that the model may rely on incorrect assumptions to arrive at the final answer. The opposite pattern suggests that the model changes its final decision despite partially complete reasoning. All models except Qwen3-4B show reasoning issues on AR-LSAT, which is also reflected in their accuracy.

We also examine individual tokens appearing in the justifications for each code, with an emphasis on negative tokens. We generate justifications only for criteria assigned a score of (0), enabling targeted negative-token analysis. Table 12 reports the top 20 tokens for each code across all models and edits. We highlight tokens that provide additional insight into common reasoning failures. Representative justifications based on selected keywords are provided below.

- **distorts:** The reasoning invents a crucial premise and **distorts** another. It states that 'all of Kelly Wearstler's designs have interesting geometries,' which is not supported by the original text. The original premises state that these designs are 'evocative' and 'dreamy.' It also **distorts** another premise by changing the antecedent of the conditional statement. Specifically, it rephrases 'If a design by Max that he adores has interesting geometries, then the design is a brutalist building and evocative' as 'if a design by Max is a design style that is Zaha Hadid's or Kelly Wearstler's, then it is a brutalist building'.

- **misinterprets:** The reasoning **misinterprets** several key premises. For option A (Louis), it states: 'If Louis is assigned to 1922, then Tiffany cannot be assigned to 1923.' This distorts condition C1 (Louis=1923 XOR Tiffany=1923), which actually implies that if Louis is NOT in 1923 (e.g., in 1922), then Tiffany *must* be in 1923. For option C (Onyx), it **misinterprets** condition C4, incorrectly deducing that 'Ryan must be assigned to 1923 or 1922' when Onyx is in 1922. For option D (Ryan), it **misinterprets** condition C3 (Tiffany_assigned -> Ryan_assigned) by stating 'If Ryan is assigned to 1922, then Tiffany must be assigned,' which is the converse and not necessarily true.

- **invents:** The reasoning **invents** a premise by stating '1. Bird Order: The birds are arranged in a fixed order: blue jay, owl, falcon, hawk, raven, crow, hummingbird.' This is a list of the birds' names, not their fixed arrangement. It then **invents** a specific bird order ('owl, blue jay, raven, falcon, hawk, hummingbird, crow') which contradicts the given premise that 'The falcon is the second from the left' (in the invented order, the falcon is fourth).

- **leap:** The reasoning contains multiple invalid logical steps and unsupported **leaps**. In Scenario 1, the deduction that P must be performed before O is unsupported, and the subsequent deductions about the positions of P and S rely on this invalid leap. More critically, the reasoning misses a direct contradiction: if $R = 1$ and $F = 2$, then T must be 1 by condition 1, which places both T and R at position

	Dataset	Edits	PF	IV	AC	AL	Overall	Correct	Count
Gemma3-1B	AR-LSAT	0	3	0	0	1	1.00	1	24
		3	0	0	0	2	0.50	2	24
	FOLIO	0	5	3	3	11	5.50	4	14
		2	4	3	3	12	5.50	5	14
	Deduction	0	4	0	0	2	1.50	2	9
		4	1	0	1	0	0.50	0	9
Gemma3-4B	AR-LSAT	0	9	0	1	5	3.75	1	24
		3	0	0	0	5	1.25	2	24
	FOLIO	0	11	11	10	14	11.50	11	14
		2	10	5	10	14	9.75	9	14
	Deduction	0	3	1	1	6	2.75	3	9
		4	6	0	1	8	3.75	6	9
Gemma3-12B	AR-LSAT	0	16	2	6	16	10.00	9	24
		3	7	0	0	4	2.75	4	24
	FOLIO	0	11	8	12	13	11.00	9	14
		2	13	9	12	13	11.75	12	14
	Deduction	0	6	4	6	7	5.75	6	9
		4	6	4	5	8	5.75	8	9
Gemma3-27B	AR-LSAT	0	12	2	5	6	6.25	6	24
		3	16	3	5	10	8.50	10	24
	FOLIO	0	13	11	13	13	12.50	12	14
		2	12	10	10	14	11.50	12	14
	Deduction	0	6	5	6	9	6.50	8	9
		4	7	5	9	9	7.50	9	9
Llama3-8B	AR-LSAT	0	5	0	0	2	1.75	7	24
		3	1	0	0	3	1.00	5	24
	FOLIO	0	8	2	4	10	6.00	7	14
		2	6	3	5	8	5.50	7	14
	Deduction	0	5	3	5	6	4.75	6	9
		4	0	0	0	3	0.75	3	9
Qwen3-4B	AR-LSAT	0	20	18	22	21	20.25	21	24
		3	18	14	17	15	16.00	18	24
	FOLIO	0	13	12	13	14	13.00	11	14
		2	12	12	13	12	12.25	10	14
	Deduction	0	8	8	9	8	8.25	9	9
		4	7	6	8	8	7.25	8	9
Phi4mini-4B	AR-LSAT	0	6	0	2	6	3.50	2	24
		3	6	0	0	9	3.75	4	24
	FOLIO	0	12	7	10	12	10.25	10	14
		2	8	2	5	12	6.75	9	14
	Deduction	0	6	2	2	5	3.75	5	9
		4	5	1	2	4	3.00	1	9

Table 10: Full qualitative evaluation results across models, datasets, and label-preserving edit levels. Overall reasoning and Correct final outputs are reported and should be considered with respect to the total number of records used in the analysis, termed as Count. For presentation, edit levels are limited to level 0 and the maximum available edit level for each dataset.

1. In Scenario 2, the inference that T cannot be performed immediately before F is a direct logical error based on condition 1. The inference that O must be first is also invalid given the constraint $F__O$. The statement that the remaining compositions L, H, and R must be performed fourth, fifth, and sixth directly contradicts the scenario's initial premise that R is performed last.

- **skips:** The reasoning **skips** the crucial intermediate steps needed to systematically combine the relative-position constraints and derive the fixed order of the birds. It presents an assumed order without derivation, identifies a contradiction, reaches a premature conclusion, and then makes an unsupported jump to the final correct conclusion about the leftmost bird without showing the valid intermediate steps.

- **contradicts:** The final answer, "D) The robin is the rightmost," **contradicts** the model's own derived sequence: Quail, Hummingbird, Robin, Raven, Blue Jay, Falcon, in which Falcon is the rightmost. It also **contradicts** the model's own evaluation of option D, which it marks as inconsistent with the derived order.

These tokens show certain behaviors of large language models that needs to be studied in detail. This qualitative analysis is a starting step and provides valuable insights into the internal tendencies of LLMs and supports the development of more rigorous evaluation methods.

I Models and Configurations

We use seven open-source models obtained from Hugging Face repositories. The Gemma models are run through the API for Chain-of-Thought and Symbolic CoT inference. Since the API does not support inference-time scaling, we run the scaling experiments on A100 GPUs. For the remaining models, we use A100 GPUs with 100GB memory and run inference with temperature (=0.1), unless otherwise specified.

For inference-time scaling, the number of samples for Gemma 27B is limited to 8 due to memory and resource constraints. For all other models, we generate 32 samples at temperature (=0.7). For CoT prompting, we use a maximum generation length of 10,000 tokens. For the other two prompting strategies, we limit generation to 2,000 tokens to manage memory usage. Any performance distortion, such as that observed for Qwen, may be due to the smaller output-token limit, which can lead to incomplete inferences on datasets such as AR-LSAT, where reasoning is substantially more complex.

For symbolic data generation, back-translation, and qualitative-analysis generation, we use `Gemini 2.5 Flash`, unless otherwise specified.

We use only open-source models that are publicly available through Hugging Face repositories. All models and datasets used in this work are publicly available and do not contain sensitive information.

	Model	Preserved Label				Altered Label			
		Base#	**Edit#**	**Base Acc.**	**Edit Acc.**	**Base#**	**Edit#**	**Base Acc.**	**Edit Acc.**
FOLIO	Gemma3-1B	68	182	47.06%	41.47%	50	69	54.00%	26.67%
	Gemma3-4B	68	182	72.06%	74.44%	50	69	72.00%	42.67%
	Gemma3-12B	68	182	79.41%	85.34%	50	69	74.00%	58.00%
	Gemma3-27B	68	182	76.47%	80.02%	50	69	80.00%	56.67%
	Llama3-8B	68	182	52.94%	58.23%	50	69	54.00%	32.67%
	Qwen3-4B	68	182	88.24%	88.11%	50	69	94.00%	71.33%
	Phi4mini-4B	68	182	70.59%	70.12%	50	69	58.00%	53.33%
Deduction	Gemma3-1B	123	397	33.33%	29.34%	157	175	18.47%	21.34%
	Gemma3-4B	123	397	46.34%	43.43%	157	175	31.21%	30.89%
	Gemma3-12B	123	397	85.37%	88.77%	157	175	80.25%	72.56%
	Gemma3-27B	123	397	86.99%	89.78%	157	175	87.26%	74.31%
	Llama3-8B	123	397	69.11%	65.05%	157	175	49.68%	41.03%
	Qwen3-4B	123	397	97.56%	96.88%	157	175	95.54%	95.22%
	Phi4mini-4B	123	397	68.29%	71.42%	157	175	65.61%	57.38%

Table 11: Base and edit accuracy across datasets and models under label-preserving and label-altering conditions. Accuracy values are reported as percentages with eval accuracy reported as a grouped accuracy.

Premise Fidelity	Inference Validity	Completeness	Answer Alignment	Overall Reasoning
reasoning	reasoning	reasoning	reasoning	reasoning
premise	invalid	fails	answer	answer
assigned	option	option	final	final
condition	assigned	steps	directly	logical
model	logical	constraints	option	invalid
states	states	does	conclusion	premise
distorts	model	incomplete	supported	evaluation
constraint	unsupported	intermediate	concludes	premises
misinterprets	premise	fully	model	completeness
position	position	systematically	true	inference
original	incorrectly	options	flawed	dimensions
second	condition	valid	follows	significant
invents	inference	conditions	based	issues
option	steps	necessary	does	fidelity
performed	constraint	constraint	contradicts	validity
premises	makes	correctly	options	suffers
recorded	second	explore	states	unsupported
given	leap	complete	possible	fundamental
incorrectly	does	skips	correct	leading
stating	contains	condition	provided	inferences

Table 12: Top 20 Tokens Across Justification Categories with negative tokens highlighted.

AR-LSAT: Prompt and an In-Context Example for back-translation

System:
You are given a symbolic logic expression and a natural-language statement with an incorrect logical operator. Correct the statement to match the symbolic logic. Always output in valid JSON only. Do not add any explanations or text outside the JSON.

User:
Task Description:
You are given: 1) A symbolic logic expression describing a rule. 2) A natural-language statement that uses the correct entities and vocabulary but contains an incorrect logical operator.
The incorrect statement differs from the symbolic logic only in the logical operator used.

Important Logical Semantics:
- AND → both conditions must be true
- OR → at least one condition must be true
- XOR → exactly one condition is true (but not both)
- NOT → the condition is false

Output Requirements:...
Guidelines: ...

Example:
```
{"symbolic_logic":"Xor(position(L) == 8, position(H) == 8)",
"incorrect_statement":"The eighth composition performed is either L or H.",
"correct_statement":"The eighth composition performed is either L or H, but not
both."}
```

Figure 4: Instruction prompt (shortened) for translating first-order logic to natural-language statements for AR-LSAT

Logical Deduction: Prompt and an In-Context Example for back-translation

System:
You are a parser that converts logical expressions into natural-language statements...

User:
Task Description:
You are given: 1) A symbolic logic expression describing a relationship (e.g., ordering, position, equality, or inequality) among objects in a domain. 2) A natural-language statement that uses the correct vocabulary but represents the logic incorrectly. Your task is to interpret the symbolic logic expression and produce a correct natural-language statement that accurately reflects its logical meaning.
Output Requirements:...
Guidelines:...

Example:
```
{"symbolic_logic":"tractor < bus",
"incorrect_statement":"The tractor is older than the bus.",
"correct_statement":"The tractor is younger than the bus."}
```

Figure 5: Instruction prompt (shortened) for translating logical form to natural-language statements for Deduction

FOLIO: Prompt and an In-Context Example for back-translation

System:
You are a parser that converts First-Order Logic (FOL) expressions into natural-language statements...

User:
Task Description:
You are given:
1) A first-order logic (FOL) expression.
2) A natural-language statement that uses the correct vocabulary but has an incorrect logical structure.
Your task is to interpret the FOL expression and produce the correct natural-language statement that accurately reflects its logical meaning.
Output Requirements:...
Guidelines:...

Example:
{"symbolic_logic":"$\forall x(Fish(x) \rightarrow Plant(x))$",
"incorrect_statement":"No fish are plants.",
"correct_statement":"All fish are plants."}

Figure 6: Instruction prompt (shortened) for translating logical form to natural-language statements for FOLIO

Reasoning Evaluation: Prompt for Qualitative Binary Coding

System:
You are an expert qualitative-analysis system designed to classify patterns, strategies, and evaluation dimensions present in model-generated reasoning or text. You must always output a single valid JSON object...

User:
Task Description:
Your task is to evaluate a reasoning path using the original task information and the model's reasoning text. The goal is to extract structured qualitative results from the reasoning path using four binary coding dimensions.

You will be given:
1) The original question, including any context, options, or constraints.
2) The reasoning text produced by a model.

Evaluate the reasoning using the following four codes. Assign each code a binary score, where `1` means the reasoning satisfies the code and `0` means it does not satisfy the code, has meaningful issues, ...

Evaluation Codes:
1) **Premise fidelity**: Assess whether the reasoning accurately uses only the information provided in the original question and avoids misreading, distorting, or inventing premises.
2) **Inference validity**: Assess whether the reasoning steps logically follow from the given premises and from prior reasoning steps.
3) **Completeness**: Assess whether the reasoning includes the necessary intermediate steps needed to support the conclusion.
4) **Answer alignment**: Assess whether the final answer follows from and is supported by the reasoning.
Evaluation Rules:...

Assistant:

```
{
  premise_fidelity: {
    score: 0,
    justification: The model significantly reinterprets premise 5 ...
  },
  inference_validity: {
    score: 1,
    justification:
  },
  ...
  }
}
```

Figure 7: Instruction prompt (shortened) for evaluating model-generated reasoning paths using four binary qualitative coding dimensions