Syntactic Blind Spots: How Misalignment Leads to LLMs' Mathematical Errors

Dane Williamson, Yangfeng Ji, Matthew Dwyer
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{dw3zn, yj3fs, md3cn}@virginia.edu

Abstract

Large Language Models (LLMs) demonstrate strong mathematical problem-solving abilities but frequently fail on problems that deviate syntactically from their training distribution. We identify a systematic failure mode, `syntactic blind spots`, in which models misapply familiar reasoning strategies to problems that are semantically straightforward but phrased in unfamiliar ways. These errors are not due to gaps in mathematical competence, but rather reflect a brittle coupling between surface form and internal representation. To test this, we rephrase incorrectly answered questions using syntactic templates drawn from correct examples. These rephrasings, which preserve semantics while reducing structural complexity, often lead to correct answers. We quantify syntactic complexity using a metric based on Dependency Locality Theory (DLT), and show that higher DLT scores are associated with increased failure rates across multiple datasets. Our findings suggest that many reasoning errors stem from structural misalignment rather than conceptual difficulty, and that syntax-aware interventions can reveal and mitigate these inductive failures.

1 Introduction

Large Language Models (LLMs) show strong performance on mathematical benchmarks like GSM8K, SVAMP, MultiArith, and ASDiv (Cobbe et al., 2021; Patel et al., 2021; Roy and Roth, 2015; Miao et al., 2020), yet they frequently make systematic errors, often reapplying familiar solution strategies even when the problem structure changes (Zheng et al., 2024; Bao et al., 2025; Huang et al., 2025).

These errors reflect an overreliance on surface-level pattern matching rather than adaptive reasoning. We focus on a specific class of such failures, which we term *syntactic misalignment*, cases where LLMs fail because a problem's phrasing deviates structurally from patterns they have learned to solve, even though the underlying logic remains unchanged. As shown in Figure 1, rephrasing a problem to reduce syntactic complexity can often reverse these failures.

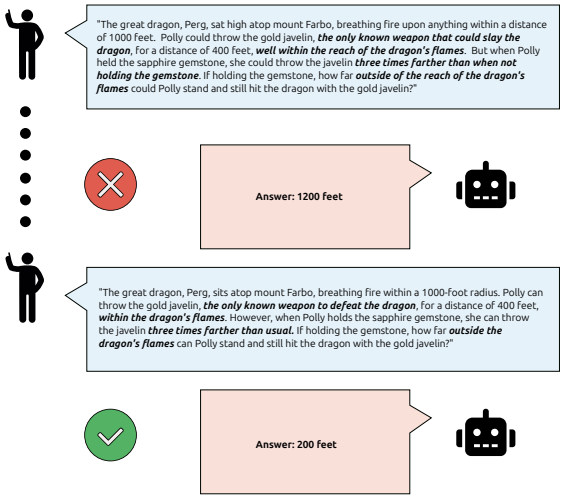


Figure 1: Structural rephrasing improves model accuracy by reducing syntactic complexity and dependency length.

Prior work has documented LLM sensitivity to superficial variation: word order changes, paraphrasing, or structural perturbations can all induce performance drops (Zhang et al., 2024; Srivastava et al., 2024; Huang et al., 2025). Models sometimes ignore altered constraints and produce answers consistent with the original phrasing.

We argue that this brittleness arises from a structural failure mode we call *syntactic induction*, the tendency to treat syntactic similarity as a proxy for problem similarity. This leads models to overapply familiar solution templates, even when the problem logic has changed. Inspired by cognitive science, we draw an analogy to *rule-based overgeneralizations* in human learners (Ben-Zeev, 1998; Karmiloff-Smith, 1986), where errors arise not from lack of competence but from misapplied procedural regularities.

To study this phenomenon, we develop a dependency-guided framework for identifying and

Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025), pages 1–14

mitigating syntactic induction failures. We use Dependency Locality Theory (DLT) to quantify syntactic complexity and rephrase high-complexity questions using syntactic templates drawn from successful examples. This reveals that many reasoning errors stem not from mathematical difficulty, but from a mismatch between surface form and learned problem schemas.

1.1 Contributions

This work makes four contributions:

- Introduces *syntactic induction failures* as a structured, recurring error mode in LLM mathematical reasoning.
- Bridges LLM error behavior with cognitive science, highlighting parallels in schema-driven failure.
- Proposes a dependency-guided framework for detecting and rephrasing structurally misaligned problems.
- Demonstrates that rephrasing structurally complex math questions significantly improves model accuracy across datasets and models.

2 Related Work

LLM Sensitivity to Structural Variation. While techniques like in-context learning and chain-of-thought prompting have improved LLM math performance (Brown and Mann, 2020; Wei et al., 2022), models remain brittle under surface-level perturbations. Studies have shown that modifying word order, phrasing, or structure leads to significant performance drops (Huang et al., 2025; He et al., 2024; Kang et al., 2024; Zheng et al., 2024; Srivastava et al., 2024). Even when semantics are preserved, models often revert to memorized patterns (Zhang et al., 2024), suggesting an overreliance on surface form as a proxy for problem identity.

Beyond formatting, deeper failure modes have been linked to data contamination (Magar and Schwartz, 2022; Sainz et al., 2023), deductive errors (Ling et al., 2023), and spurious correlations (Zhou et al., 2024; Bao et al., 2025). Most relevantly, Stechly et al. (2025) show that models struggle when questions are phrased in unfamiliar syntactic forms, motivating our focus on syntactic misalignment.

Cognitive Accounts of Structural Sensitivity. These observations parallel well-known findings in cognitive psychology. Chi and et al (1981) distinguish between a problem's surface structure (e.g., phrasing, grammar) and its deep structure (underlying logic). Even experienced solvers often rely on surface cues to access problem schemas (Novick, 1988; Hinsley et al., 1977), which guide solution strategies. Analogical reasoning studies further show that surface similarity influences both novice and expert behavior (Holyoak and Koh, 1987; Ross, 1984).

Our work draws on this perspective, proposing that LLMs exhibit a similar schema-triggered behavior. We formalize this through *syntactic induction*: the tendency to conflate surface-form similarity with structural equivalence. To quantify this effect, we adopt Dependency Locality Theory (DLT) (Gibson, 1998, 2000) and show that higher DLT scores are associated with LLM failure. Rephrasing high-DLT questions often recovers accuracy, supporting a structural account of reasoning breakdown.

Toward a Taxonomy of Rational Errors. Finally, our perspective aligns with work on human error categorization. Rule-based overgeneralizations (Ben-Zeev, 1998; Ashlock, 2002), where valid strategies are misapplied in the wrong context, mirror LLM errors under syntactic shift. We argue that LLMs, like learners, may benefit from structured taxonomies of failure to guide robustness interventions (see Appendix Figure 9).

3 Method: Rephrasing for Reducing Syntactic Misalignment

LLMs often fail when problems are phrased in structurally unfamiliar ways. To diagnose and mitigate these errors, we quantify syntactic complexity using Dependency Locality Theory (DLT) and rephrase structurally complex questions to align with previously successful examples. This section outlines the framework.

3.1 Quantifying Syntactic Complexity with DLT

We define syntactic complexity using a scoring function based on Dependency Locality Theory (DLT). For a math word problem, $q = (w_1, \ldots, w_n)$, the total DLT score is the sum of three costs over all tokens:

$$\mathrm{DLT}(q) = \sum_{i=1}^{n} \big(\mathrm{Integration}(w_i) + \mathrm{Storage}(w_i) + \mathrm{Discourse}(w_i)\big) \quad (1)$$

Given the dependency trees of the sentences in q each component in Equation 1 is defined as follows:

Integration : For a token w_i, let h_i be its head token following the dependency tree. The integration cost is the number of new discourse referents between them: $\mathrm{Integration}(w_i) = \sum_{w_j \in \mathrm{Intervening}(w_i, h_i)} \mathbf{1}_{\mathrm{Referent}}(w_j)$, where $\mathbf{1}_{\mathrm{Referent}}(w_j) = 1$ if w_j has POS tag in {NOUN, PROPN, NUM, VERB}.

Discourse : A token introduces discourse cost if it adds a new referent: $\mathrm{Discourse}(w_i) = \mathbf{1}_{\mathrm{Referent}}(w_i)$.

Storage : This is the number of unresolved syntactic expectations at w_i, denoted $\mathrm{Storage}(w_i) = |\mathcal{P}_i|$, where $\mathcal{P}_i$ is the set of pending predictions.

This formulation enables theoretically-grounded, systematic scoring of syntactic complexity for each question based on its dependency parse.

Example: *"Melissa brushes 12 horses on Monday."*

- **Discourse:** All content words ("Melissa", "brushes", "12", "horses", "Monday") introduce discourse referents (PROPN, VERB, NUM, NOUN), yielding a total cost of 5.
- **Integration:** "Horses" depends on "brushes", with "12" intervening. Since "12" is a referent, it contributes an integration cost of 1.
- **Storage:** "Melissa" introduces an unresolved expectation for a verb, resolved by "brushes". New expectations (e.g., for an object) are tracked until resolved.

3.1.1 Normalization

To ensure fair comparison across questions of varying length and content density, we normalize two components of the DLT score. Integration cost is divided by the number of discourse referents, and peak storage cost is scaled by question length. The resulting normalized DLT score is:

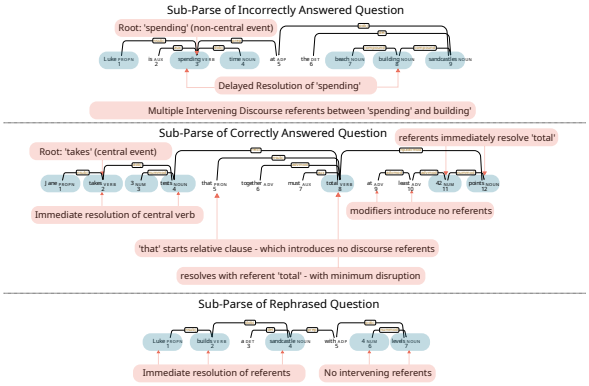


Figure 2: Dependency parses illustrating the rephrasing pipeline. The rephrased version reduces dependency depth and referential interference, lowering DLT-based processing cost.

$$\mathrm{DLT}_{\mathrm{norm}}(q) = \left(\frac{\sum \mathrm{Integration}}{\sum \mathrm{Discourse}}\right) + \left(\frac{\max \mathrm{Storage}}{|q|}\right) + \left(\sum \mathrm{Discourse}\right) \quad (2)$$

This yields a length-independent complexity measure. As shown in section 5, higher normalized-DLT scores correlate with model failure, making this a reliable predictor of syntactic brittleness.

3.2 Dependency-Guided Rephrasing

To correct syntactic misalignment, we rephrase a failed question $q_{\mathrm{incorrect}}$ to resemble a syntactically similar, correctly answered one q_{match}. We identify q_{match} using the Weisfeiler-Lehman Graph Kernel (WLK) (Shervashidze et al., 2011):

$$q_{\mathrm{match}} = \arg \max_{q \in Q_{\mathrm{correct}}} \mathrm{WLK}(G_{\mathrm{incorrect}}, G_q) \quad (3)$$

We then prompt an LLM to rewrite $q_{\mathrm{incorrect}}$ to match the structure of q_{match}, while preserving semantics:

$$q'_{\mathrm{incorrect}} = \mathcal{M}(q_{\mathrm{incorrect}}, q_{\mathrm{match}}, \mathcal{P}) \quad (4)$$

For example, consider this original question:

> "Luke is spending time at the beach building sandcastles. He eventually notices that..."

Its syntactic embedding leads to high DLT cost. The rephrased version:

> "Luke builds a sandcastle with 4 levels, where each level has half the square footage..."

flattens dependencies, reduces referential interference, and improves model accuracy.

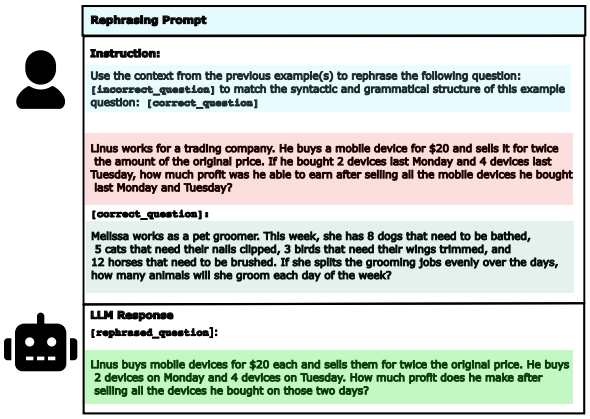


Figure 3: Format of rephrasing prompt. The LLM is prompted to generate a rephrased variant that more closely matches the surface structure of the correctly answered question.

3.3 Procedure Overview

Our pipeline consists of:

1. Query the model; collect incorrect responses $Q_{\text{incorrect}}$.
2. Parse all questions using `spaCy` to extract dependency trees.
3. For each $q \in Q_{\text{incorrect}}$, find $q_{\text{match}} \in Q_{\text{correct}}$ via WLK similarity.
4. Prompt the LLM with k-shot examples to rephrase q syntactically in the form of q_{match}.

We then re-query the model on the rephrased versions q' and evaluate whether failures are recovered.

4 Experiment Setup

To assess the impact of syntactic restructuring, we re-evaluate the LLM on the rephrased variants $q'_{\text{incorrect}}$. If accuracy improves significantly, we attribute the original failure to a *syntactic induction failure*: the model's inability to generalize over unfamiliar surface forms despite semantic equivalence.

This evaluation allows us to systematically characterize and quantify a core weakness in LLM reasoning and establish the importance of syntactic alignment for mathematical understanding.

This section outlines our experimental framework for evaluating how syntactic structure influences LLM reasoning performance. We begin by stating our research question, (Section 4.1). We then describe the datasets and preprocessing methods (Section 4.2). Finally, we provide implementation details regarding parsing, tree similarity computation, and model evaluation (Section 4.3).

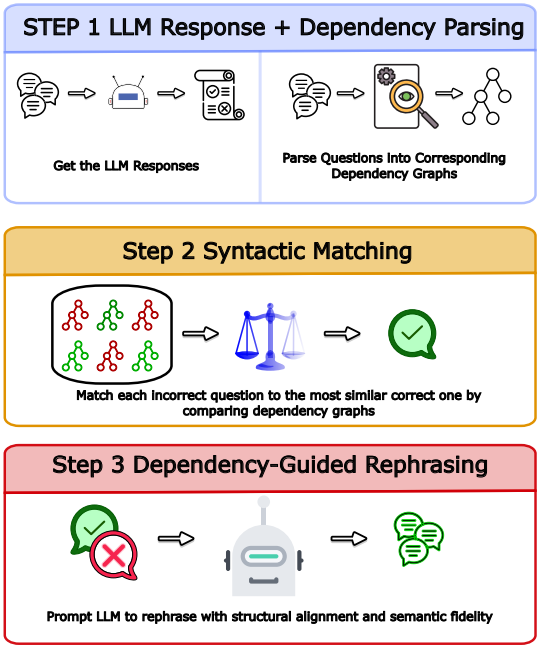


Figure 4: Rephrasing pipeline. An incorrectly answered question is aligned to a syntactically similar, correctly answered one via WL Kernel matching. A k-shot prompt then guides the LLM to generate a syntactically aligned but semantically identical rephrasing.

4.1 Research Questions

This work investigates whether syntactic complexity contributes to reasoning failures in LLMs and whether syntactic restructuring can mitigate those failures. Specifically, we ask:

1. **How effectively does the proposed DLT-based complexity framework differentiate between correctly and incorrectly answered math questions?**
2. **To what extent does syntactic rephrasing, guided by structural similarity to successfully answered questions, improve model accuracy on previously failed examples?**

Before evaluating the effects of syntactic rephrasing, we first investigate whether syntactic complexity alone can be predictive of model failure. For each model, we compute DLT complexity scores for all questions and divide the dataset into two groups: those answered correctly and those answered incorrectly.

To assess whether there is a statistically significant difference in complexity between the two

groups, we apply Welch's t-test. This test is appropriate when comparing the means of two samples with potentially unequal variances and sample sizes, conditions that naturally arise given varying model accuracies. The resulting *t-statistic* quantifies the separation between group means relative to their variances, while the corresponding *p-value* indicates whether the observed difference is likely to have occurred by chance.

This analysis allows us to test the hypothesis that higher syntactic complexity, independent of semantic content, is associated with increased model error. A significant result would suggest that DLT complexity serves as a useful predictor of LLM failure, motivating our subsequent rephrasing intervention.

We report the results of this comparison in subsection 5.1 and interpret its implications in subsection 6.1.

4.2 Datasets and Pre-processing

We evaluate five open-source LLMs, **LLaMA**, **Mistral**, **Qwen**, **Gemma**, and **Granite**, in a zero-shot setting on four established math benchmarks: **GSM8K**, **SVAMP**, **MultiArith**, and **ASDiv** (Touvron et al., 2023; Jiang and et al, 2023; Yang et al., 2024; Team et al., 2024; Mishra and et al, 2024). These datasets span diverse reasoning skills and syntactic forms, from simple arithmetic (GSM8K, MultiArith) to structurally perturbed (SVAMP) and linguistically varied (ASDiv) problems.

To analyze syntactic structure, we parse each question using the spaCy NLP toolkit (Honnibal and Montani, 2017), yielding dependency trees that capture syntactic relations. Let $T_{\text{incorrect}}$ denote the tree of an incorrectly answered question, and $\mathcal{T}_{\text{correct}}$ the set of trees from correctly answered ones.

4.3 Implementation Details

Dependency parsing and tree similarity computations are implemented using `spaCy` and `nltk`. We use Hugging Face implementations of all LLMs. [1] All experiments are conducted on NVIDIA RTX 2080 GPUs.

5 Experimental Results

We first test whether surface-level syntactic complexity predicts model failure. We then assess whether syntactic restructuring can recover accuracy on previously incorrect questions.

[1] See Appendix Table 4 for model and hyperparameter details.

Dataset	Model	Correct Mean	Incorrect Mean	Δ DLT
GSM8K	Gemma	22.90	25.71	**+2.81***
	Granite	22.17	25.43	**+3.27***
	LLaMA	23.53	28.10	**+4.57***
	Mistral	22.98	26.21	**+3.23***
	Qwen	23.86	29.01	**+5.15***
SVAMP	Gemma	17.34	18.70	**+1.36***
	Granite	16.95	18.58	**+1.62***
	LLaMA	17.60	18.88	**+1.28***
	Mistral	17.42	18.60	**+1.18***
	Qwen	17.80	18.57	+0.78
MultiArith	Gemma	17.24	17.29	+0.05
	Granite	17.39	17.11	-0.28
	LLaMA	17.25	17.15	-0.10
	Mistral	17.36	16.98	-0.38
	Qwen	17.19	19.26	**+2.07***
ASDiv	Gemma	16.70	16.87	+0.17
	Granite	16.10	17.37	**+1.27***
	LLaMA	16.39	18.43	**+2.04***
	Mistral	16.17	17.84	**+1.67***
	Qwen	16.70	17.58	+0.88

Table 1: Mean DLT complexity scores for correctly and incorrectly answered questions across datasets and models. Δ DLT is the difference. **Bolded values with * indicate statistically significant differences** ($p < 0.01$, Welch's t-test).[2]

5.1 Syntactic Complexity of Incorrect Questions

Table 1 reports the mean normalized DLT complexity scores on both sets of questions. GSM8K exhibits unanimously higher syntactic complexity scores on incorrectly answered questions across all models, with Welch's $p < 0.01$ in every case (full test statistics are provided in the supplementary material). On SVAMP, all deltas are positive, with four reaching statistical significance. MultiArith and ASDiv show weaker or inconsistent trends, with smaller or statistically insignificant differences.

5.2 Accuracy Gains from Rephrasing

We define performance improvement in terms of the change in accuracy after rephrasing, denoted by ΔA. Let Q_{total} denote the full set of questions, and $Q'_{\text{correct}} \subseteq Q_{\text{incorrect}}$ represent the set of previously incorrect questions that are now answered correctly after rephrasing. We compute:

$$\Delta A = \frac{|Q'_{\text{correct}}|}{|Q_{\text{total}}|} \tag{5}$$

Final model accuracy is then updated as:

$$\text{New Accuracy}(A) \tag{6}$$

$$= \text{Original Accuracy}(A_0) + \Delta A \tag{7}$$

[2] See Appendix Figure 5 for supporting visualizations.

Model	GSM8K				SVAMP				MultiArith				ASDiv			
	A_0	ΔA	A	#Recovered	A_0	ΔA	A	#Recovered	A_0	ΔA	A	#Recovered	A_0	ΔA	A	#Recovered
Gemma-7B	37.76	8.26	46.02	109	61.71	7.14	68.85	50	77.62	4.76	82.38	20	59.61	9.11	68.72	210
Granite-7B	24.03	11.68	35.71	154	44.43	11.86	56.29	83	50.00	15.48	65.48	65	64.08	9.68	73.75	223
LLaMA-8B	75.44	7.81	83.25	103	79.86	4.00	83.86	28	94.76	3.57	98.33	15	81.43	5.08	86.51	117
Mistral-7B	48.29	11.45	59.74	151	63.29	8.29	71.57	58	70.71	14.52	85.24	61	47.25	12.10	59.35	279
Qwen-7B	84.53	4.32	88.86	57	92.43	1.43	93.86	10	97.14	0.48	97.62	2	91.76	1.82	93.58	42

Table 2: Accuracy improvements and number of recovered answers from syntactic restructuring across GSM8K, SVAMP, MultiArith, and ASDiv. A_0 is baseline accuracy, ΔA is improvement after rephrasing, A is final accuracy, and #Recovered denotes incorrect answers corrected by rephrasing.

This formulation quantifies the overall gain attributable to syntactic restructuring, allowing us to isolate its impact.

Table 2 reports the accuracy improvements and the number of recovered answers from syntactic restructuring. All models improve on GSM8K and SVAMP, with lower-performing models (e.g., Gemma, Granite) showing the greatest relative gains. Improvements are less consistent on MultiArith and ASDiv, where most models already achieve high baseline accuracy or rephrasing yields fewer recoveries.

We observe that syntactic restructuring is most impactful on datasets with more narrative or structurally varied question phrasing (e.g., GSM8K, SVAMP), suggesting that syntactic mismatch contributes to model failures in these settings. Recovery counts ranged from 2 (Qwen on MultiArith) to 210 (Gemma on ASDiv), with rephrasing improving accuracy by as much as 15.5 % (Granite on MultiArith).

These findings provide strong empirical support for our hypothesis that LLMs fail on syntactically unfamiliar problems, and that rephrasing toward familiar structures mitigates these errors.

6 Further Analysis

The section provides further analysis regarding how syntactic structure influences LLM reasoning behavior. It examines four key dimensions: First, we show that elevated syntactic complexity, measured using DLT, predicts failure on narrative math tasks. Second, we demonstrate that rephrasing these complex questions into syntactically familiar forms improves model accuracy, supporting an interpretation of failure as schema misalignment. Third, we analyze these errors in light of cognitive theory, arguing that LLMs overapply familiar strategies to structurally novel inputs, a form-function misalignment. Finally, we outline implications for robustness and generalization, proposing syntax-aware interventions and cognitively grounded training approaches.

6.1 Syntactic Complexity Predicts Failure on Narrative Math Tasks

The results from Table 1 show that syntactic complexity, as measured by DLT scores, is positively associated with model failures, particularly on GSM8K and SVAMP. On GSM8K, all five LLMs exhibit statistically significant increases in complexity on incorrectly answered questions. On SVAMP, all deltas are positive, though only four reach statistical significance. For ASDiv, all models again show positive deltas, with three of them statistically significant. In contrast, the pattern is weaker on MultiArith, where only two of the five models show positive deltas and just one achieves significance. These results support the hypothesis that LLMs are sensitive to structural features of problem statements, especially on narrative-heavy datasets like GSM8K and SVAMP. By contrast, MultiArith's more uniform, low-complexity phrasing likely shifts the source of failure away from syntactic burden and toward reasoning depth.

These findings are consistent with the previously outlined phenomenon of *syntactic induction*, in which models perform worse on problems that deviate from familiar surface forms. In our experiments, LLMs consistently exhibited higher failure rates on syntactically complex questions, particularly when those forms differed structurally from common patterns. This suggests that model predictions are sensitive to surface structure, and that unfamiliar phrasing can impair accuracy even when underlying reasoning demands remain constant.

From a cognitive perspective, these errors reflect a failure in structural fluency. The DLT framework quantifies this fluency as a function of integration cost, storage cost, and discourse load. Elevated scores among failure cases suggest that LLMs, like human solvers, are vulnerable to breakdowns in processing when these syntactic burdens accumulate beyond an internalized threshold.

6.2 Rephrasing as Schema Alignment

Table 2 demonstrates that rephrasing structurally complex questions into syntactically familiar forms yields substantial accuracy improvements. This pattern is most pronounced on GSM8K and SVAMP, particularly among lower-performing models such as Gemma and Granite. While high-performing models like LLaMA and Qwen show smaller deltas, they also exhibit measurable gains, supporting the interpretation that rephrasing facilitates access to familiar problem-solving patterns across models.

These improvements reinforce the interpretation of failure as a *schema alignment problem*. According to prior work in cognitive psychology, solvers often rely on surface cues to activate latent problem schemas. When surface form is misaligned with internal expectations, reasoning may fail despite latent competence. Rephrasing appears to bridge this gap, effectively priming models to recognize the underlying problem structure.

The consistency of this effect across architectures suggests that the phenomenon is not model-specific but a general feature of current LLM design.

6.3 Syntax Cues the Wrong Strategy: Evidence of Form-Function Misalignment

Our findings reflect a consistent pattern: models often fail when questions are phrased in structurally unfamiliar ways, even if the underlying reasoning task remains the same. This aligns with cognitive accounts of human error, such as those described by Ben-Zeev (1998), in which solvers misapply familiar procedures to novel input formats.

While LLMs are not rule-based agents in the human sense, our results suggest that they similarly rely on surface-level cues to guide problem-solving behavior. When syntactic structure deviates from familiar patterns, models are more likely to generate incorrect responses, even when they demonstrate competence on simpler or canonical formulations of the same task.

That even high-performing models benefit from syntactic restructuring indicates that many failures are not due to limitations in arithmetic ability per se, but rather in applying the right strategy under structural variation. This points to a challenge beyond learning correct computations: models must also determine *how* and *when* to apply learned behaviors, a process that appears sensitive to variations in syntactic form.

6.4 Toward Syntax-Aware Generalization

These results carry several implications for improving LLM robustness and interpretability. First, they emphasize the importance of training or prompting models to abstract beyond surface form. Sensitivity to syntactic variation can limit generalization, even in domains where the underlying reasoning is sound.

Second, our analysis highlights the potential of syntax-aware interventions. By measuring DLT complexity and selectively rephrasing high-complexity inputs, systems could anticipate and mitigate failure without retraining. This suggests a role for lightweight, dynamic preprocessing pipelines in real-world deployments.

Finally, our findings suggest promising directions for future research (1) **Syntactic curriculum learning**: Gradually exposing models to varied syntactic structures during training to improve generalization under structural variation. (2) **Schema-guided error analysis**: Building error taxonomies based on syntactic mismatches to inform evaluation and debugging.

Overall, our analysis suggests that syntactic induction is a significant source of LLM failure in math reasoning tasks. By quantifying structural complexity and aligning input form with successful patterns, we can better anticipate and mitigate a subset of reasoning errors rooted in input formulation.

Crucially, our findings suggest that syntactic complexity is not merely correlated with failure but may play a mediating role in model performance on structurally complex inputs. This highlights structurally guided rephrasing as a lightweight and scalable strategy for recovering from such errors, without modifying model weights or requiring additional supervision.

7 Conclusion

This work investigated how syntactic structure influences the reasoning behavior of LLMs on mathematical problems. Across four benchmarks: GSM8K, SVAMP, MultiArith, and ASDiv, we found that LLMs systematically fail on syntactically complex inputs, despite their semantic simplicity. These failures were reliably predicted by elevated Dependency Locality Theory (DLT) scores and mitigated through targeted syntactic rephrasing.

Our findings demonstrate that many reasoning errors stem not from a lack of mathematical compe-

tence, but from `syntactic induction failures`, a tendency to misapply known solution strategies when surface structure deviates from training priors. Rephrasing misaligned questions into syntactically familiar forms improved accuracy across all models, with gains particularly notable in lower-performing systems like Gemma and Granite. This supports the view, rooted in cognitive science, that schema activation in both humans and LLMs is highly sensitive to surface cues.

By framing these errors within a rule-based taxonomy and formalizing complexity through DLT, we offer a structured explanation for inductive failure in LLMs. Rather than viewing mistakes as isolated or stochastic, we show they are predictable, syntax-sensitive, and recoverable through lightweight interventions.

Future Directions This work opens several directions for enhancing LLM robustness and interpretability. We highlight:

- **Syntactic diagnostic tools**: To identify schema misalignment based on DLT complexity or parse structure.
- **Structure-aware input representations**: Leveraging dependency graphs or programmatic abstractions to make problem structure more accessible.
- **Failure-aware training curricula**: Introducing controlled syntactic variation to encourage generalization beyond form-driven heuristics.

While our experiments focus on mathematical benchmarks, the implications are broader. Syntactic induction failures may underlie reasoning brittleness across domains. Addressing these failures offers a path toward LLMs that reason more like human experts: flexibly, structurally, and with awareness of when form does, and does not, align with function.

8 Ethics Statement

Our experiments were conducted on publicly available mathematical reasoning datasets, which do not contain sensitive personal data or pose identifiable risks to individuals or groups. The work does not involve human subjects or data collection. No known ethical risks were introduced, and all referenced work is properly cited and respected under academic norms.

9 Limitations

This study focuses on mathematical word problems and may not generalize to other domains. We evaluate only final-answer accuracy, without analyzing intermediate reasoning.

References

Robert B. Ashlock. 2002. *Error patterns in computation*, 8th ed. edition. Merrill PH, New Jersy.

Guangsheng Bao, Hongbo Zhang, and et al Wang. 2025. How Likely Do LLMs with CoT Mimic Human Reasoning? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7831–7850, Abu Dhabi, UAE. Association for Computational Linguistics.

Talia Ben-Zeev. 1998. Rational errors and the mathematical mind. *Review of General Psychology*, 2(4):366–383.

Tom Brown and et al Mann. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Michelene T H Chi and Feltovich et al. 1981. Categorization and representation of physics problems by experts and novices. *Cogn. Sci.*, 5(2):121–152.

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

Edward Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson. 2000. The Dependency Locality Theory: A distance-based theory of linguistic complexity. In *Image, language, brain: Papers from the first mind articulation project symposium*, pages 95–126, Cambridge, MA. MIT Press.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X. Wang, and Sadid Hasan. 2024. Does Prompt Formatting Have Any Impact on LLM Performance? *arXiv preprint*. ArXiv:2411.10541 [cs].

Dan Alvin Hinsley, J. R. Hayes, and Herbert A. Simon. 1977. From words to equations meaning and representation in algebra word problems.

K J Holyoak and K Koh. 1987. Surface and structural similarity in analogical transfer. *Mem. Cognit.*, 15(4):332–340.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025. Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations. *Preprint*, arXiv:2502.06453.

Albert Q. Jiang and Alexandre Sablayrolles et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2024. Unfamiliar Finetuning Examples Control How Language Models Hallucinate. *arXiv preprint*. ArXiv:2403.05612 [cs].

Annette Karmiloff-Smith. 1986. From meta-processes to conscious access: Evidence from children's metalinguistic and repair data. *Cognition*, 23(2):95–147.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. *Preprint*, arXiv:2306.03872.

Inbal Magar and Roy Schwartz. 2022. Data Contamination: From Memorization to Exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the ACL*, pages 975–984, Online. Association for Computational Linguistics.

Mayank Mishra and Matt Stallone et al. 2024. Granite code models: A family of open foundation models for code intelligence. *Preprint*, arXiv:2405.04324.

L R Novick. 1988. Analogical transfer, problem similarity, and expertise. *J. Exp. Psychol. Learn. Mem. Cogn.*, 14(3):510–520.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of NAACL 2021: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Brian H. Ross. 1984. Remindings and their effects in learning a cognitive skill. *Cognitive Psychology*, 16(3):371–416.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(77):2539–2561.

Saurabh Srivastava, Annarose M. B, Anto P. V, Shashank Menon, Ajay Sukumar, Adwaith Samod T, Alan Philipose, Stevin Prince, and Sooraj Thomas. 2024. Functional Benchmarks for Robust Evaluation of Reasoning Performance, and the Reasoning Gap. *arXiv preprint*. ArXiv:2402.19450 [cs].

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2025. Chain of thoughtlessness? an analysis of cot in planning. *Preprint*, arXiv:2405.04776.

Gemma Team, Morgane Riviere, and Shreya Pathak et al. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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 *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

An Yang, Baosong Yang, and Binyuan Hui et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Hugh Zhang, Jeff Da, and Lee et al. 2024. A careful examination of large language model performance on grade school arithmetic. In *Advances in Neural Information Processing Systems*, volume 37, pages 46819–46836. Curran Associates, Inc.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. Processbench: Identifying process errors in mathematical reasoning. *Preprint*, arXiv:2412.06559.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2024. Explore Spurious Correlations at the Concept Level in Language Models for Text Classification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 478–492, Bangkok, Thailand. Association for Computational Linguistics.

A Appendix

A.1 DLT Rephrasing Algorithm (Proposed)

This subsection outlines a proposed procedure for preemptively identifying syntactically complex math word problems and evaluating the impact of rephrasing. While this algorithm was not executed in the main paper due to time constraints, it was designed to support future ablation studies. The pipeline filters questions based on elevated DLT complexity, applies dependency-guided rephrasing, and evaluates accuracy before and after restructuring. This formalization supports reproducibility and highlights a possible direction for proactive syntax-aware model evaluation.

Algorithm 1 DLT-Guided Rephrasing and Accuracy Evaluation

Require: Dataset $\mathcal{D} = \{(q_i, a_i)\}_{i=1}^{N}$ of questions and answers
Require: Normalized DLT complexity scoring function $\mathrm{DLT}(q)$
Require: Rephrasing function $\mathrm{Rephrase}(q)$
Require: Evaluation function $\mathrm{Accuracy}(\mathcal{Q}, \mathcal{A})$
1: Compute scores: $\{s_i = \mathrm{DLT}(q_i)\}_{i=1}^{N}$
2: Set threshold τ as the 75th percentile of $\{s_i\}$
3: Define $\mathcal{D}_{\text{complex}} = \{(q_i, a_i) \mid s_i \geq \tau\}$
4: For each (q_i, a_i) in $\mathcal{D}_{\text{complex}}$, compute $q_i' = \mathrm{Rephrase}(q_i)$
5: Evaluate original accuracy:

$$\mathrm{Acc}_{\text{orig}} = \mathrm{Accuracy}\left(\{q_i\}, \{a_i\}\right)$$

6: Evaluate rephrased accuracy:

$$\mathrm{Acc}_{\text{reph}} = \mathrm{Accuracy}\left(\{q_i'\}, \{a_i\}\right)$$

7: Compute improvement: $\Delta = \mathrm{Acc}_{\text{reph}} - \mathrm{Acc}_{\text{orig}}$
8: **return** τ, Δ

A.2 DLT Complexity Statistical Significance

This section provides full statistical results and supporting visualizations for the DLT complexity gaps reported in Table 1 of the main paper. The table below contains Welch's t-statistics and p-values for each dataset–model pair. Following the table, we include full boxplots of DLT scores by outcome (correct vs. incorrect) across all models and datasets. These visualizations offer a more detailed view of score distributions, variance, and effect sizes.

Table 3: Welch's t-test results comparing DLT complexity for correctly vs. incorrectly answered questions.

Model	ASDiv		GSM8K		MultiArith		SVAMP	
	t-stat.	p-val.	t-stat.	p-val.	t-stat.	p-val.	t-stat.	p-val.
Gemma	-0.66	0.51	-6.97	< 0.0001	-0.18	0.86	-3.94	< 0.0001
Granite	-5.29	< 0.0001	-6.22	< 0.0001	1.16	0.249	-4.87	< 0.0001
LLaMA	-5.92	< 0.0001	-7.44	< 0.0001	0.28	0.861	-2.94	0.0037
Mistral	-6.47	< 0.0001	-6.83	< 0.0001	1.58	0.116	-3.42	0.0007
Qwen	-1.76	0.08	-6.66	< 0.0001	-2.52	0.0275	-1.13	0.2649

A.3 Manual Evaluation of Rephrased Questions

To assess the quality and effectiveness of our syntactic rephrasing method, we conducted a manual evaluation. We selected 10 representative (original, rephrased) question pairs sampled from GSM8K, SVAMP, MultiArith, and ASDiv. Each pair was reviewed by an annotator along three criteria:

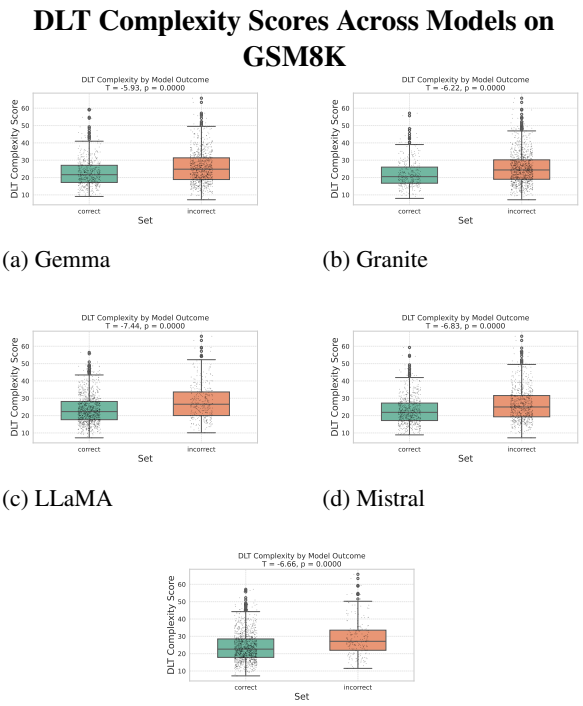
- **Semantic Match**: Does the rephrased version preserve the original problem's meaning?
- **Structural Simplification**: Does the rephrased version reduce syntactic complexity (e.g., fewer clauses, flatter dependencies)?
- **Answered Correctly**: Did the model originally answer incorrectly but succeed after rephrasing?

All 10 examples were rated as preserving semantic fidelity while simplifying structure, and all were answered correctly by the model post-rephrasing. These results reinforce the claim that syntactic restructuring can reduce complexity while maintaining problem intent, allowing models to succeed on previously failed inputs.

Table 5 summarizes the outcomes for each evaluated example.

A.4 LLM Evaluation Framework

This section lists the models and decoding parameters used in our experiments. Table 4 provides full details for both the math QA models and the rephrasing model. All math questions were evaluated in a zero-shot setting using greedy decoding (temperature = 0, no sampling). For rephrasing, we used LLaMA-3 with mild sampling settings to introduce syntactic variation while preserving semantic intent. These parameters were fixed across all datasets to ensure consistency and reproducibility.

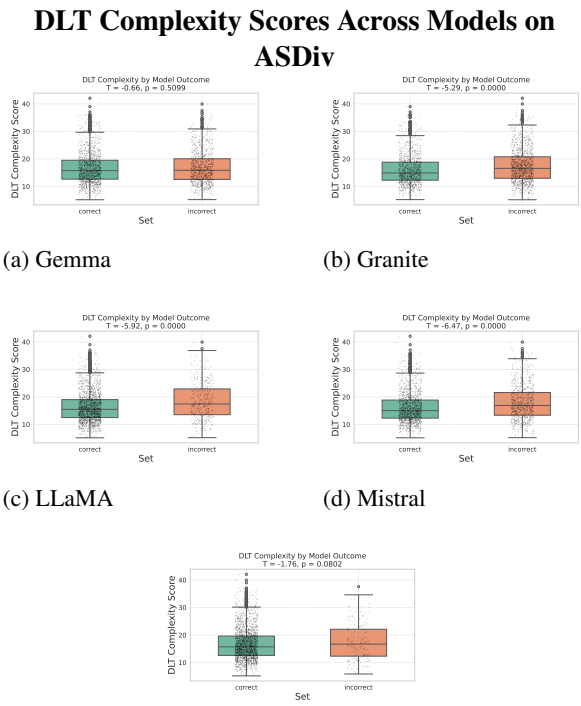


(a) Gemma (b) Granite

(c) LLaMA (d) Mistral

(e) Qwen

Figure 5: DLT complexity scores by model outcome (correct vs. incorrect) across five LLMs on GSM8K. In each subplot, incorrectly answered questions (orange) exhibit higher mean complexity and greater variance than correct ones (green). Welch's t-statistics and p-values confirm these differences are statistically significant.



(a) Gemma (b) Granite

(c) LLaMA (d) Mistral

(e) Qwen

Figure 6: DLT complexity scores by model outcome (correct vs. incorrect) across five LLMs on ASDiv.

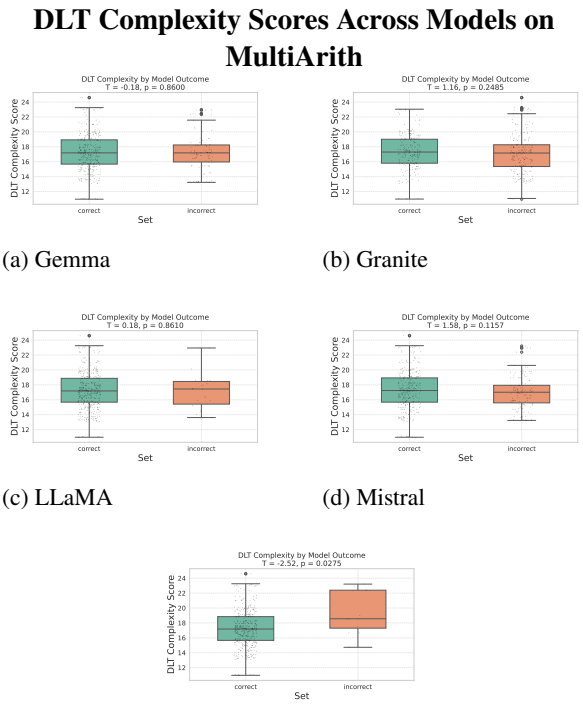
A.5 Cognitive Psychology: Rational Errors

To contextualize our findings within broader theories of reasoning failure, we draw on insights from cognitive psychology and mathematical pedagogy. Specifically, we reference the work of Ben-Zeev (1998), who frames many student errors in mathematics not as random mistakes, but as *rational errors*. systematic overgeneralizations of otherwise valid strategies.

Figure 9 illustrates this framework. The top panel shows a classic subtraction mistake: subtracting each digit in place-value order without accounting for borrowing. This type of mistake is not due to irrationality but reflects a learner's internalization of an overly simplified rule. The bottom panel presents a taxonomy of inductive failure modes, such as *syntactic induction* and *semantic induction*, which describe how solvers may misapply surface-level patterns or real-world analogies inappropriately.

These mechanisms are highly relevant to our analysis of LLM behavior. Our experiments show that LLMs often fail on syntactically novel questions not because they lack competence, but because they overapply strategies learned from structurally familiar formsm precisely the type of error Ben-Zeev characterizes as rational. In particular, what we term `syntactic induction` failures in LLMs echoes this cognitive framing, highlighting deep parallels between human and model error patterns.

We include these diagrams to situate our findings in a well-established theory of rule-based reasoning errors and to support our claim that LLM failures are often structured, interpretable, and attributable to form-function misalignment rather than arbitrary noise.

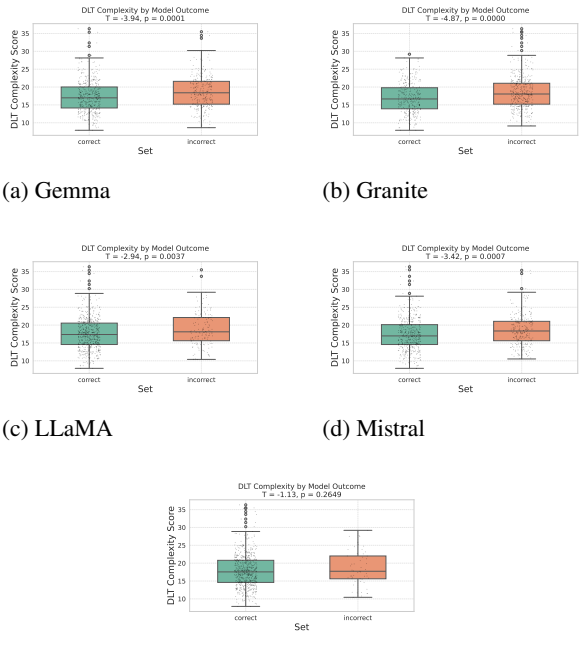


(a) Gemma (b) Granite

(c) LLaMA (d) Mistral

(e) Qwen

Figure 7: DLT complexity scores by model outcome (correct vs. incorrect) across five LLMs on Multiarith.



(a) Gemma (b) Granite

(c) LLaMA (d) Mistral

(e) Qwen

Figure 8: DLT complexity scores by model outcome (correct vs. incorrect) across five LLMs on SVAMP.

Model	Params	Max	Temp	Top-p	Sample
LLaMA-3	8B	8192	0.0	1.0	False
Mistral	7B	8192	0.0	1.0	False
Qwen2.5	7B	8192	0.0	1.0	False
Gemma	7B	8192	0.0	1.0	False
Granite	7B	8192	0.0	1.0	False

(a) Math QA models and decoding hyperparameters. All models use greedy decoding (temperature = 0, no sampling).

Rephrasing Model	Params	Max	Temp	Top-p	Sample
LLaMA-3	8B	8192	0.1	0.9	True

(b) Rephrasing model used for syntactic restructuring.

Table 4: LLMs used in experiments.

Figure 9: Correct borrowing (left) vs. a common subtraction mistake (right). Students without instruction on borrowing often overgeneralize the principle of subtracting smaller from larger numbers.

Dataset	Model	Original Question	Rephrased Question	Semantic Match	Simplified	Answered Correctly
GSM8K	LLaMA-8B	A company pays each of its employees $600 in a month. The company has a policy of increasing the salaries of each of its employees by 10% of the initial salary every year for those who've stayed in the company for five years. If Sylvie just clocked 5 years in the company last December, what's her annual salary after three more years of service?	A company pays its employees $600 per month. The company has a policy of increasing salaries by 10% of the initial salary every year for employees who have stayed with the company for five years. Sylvie just completed five years of service last December. What is her annual salary after three more years of service?	✓	✓	✓
MultiArith	Mistral-7B	Roger had 68 dollars. If he spent 47 bucks on a new game, how many 7 dollar toys could he buy with the money he had left?	Roger has 68 dollars. He spends 47 dollars on a new game. How many 7-dollar toys can he buy with the money he has left?	✓	✓	✓
ASDiv	Qwen2.5-7B	Andrew's 4 friends decided to bring food as well. If each of them brought 4 slices of pizza, how many slices of pizza do they have in total?	Andrew's 4 friends decide to bring food as well. They each bring 4 slices of pizza. How many slices of pizza do they have in total?	✓	✓	✓
GSM8K	Gemma-7B	Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone. If Suzy's iPhone is 1 year old, how old is Brandon's iPhone?	Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone, which is 1 year old. How old is Brandon's iPhone?	✓	✓	✓
SVAMP	Granite-7B	Edward spent $6 to buy 2 books each book costing him the same amount of money. Now he has $12. How much did each book cost?	Edward spent $6 to buy 2 books, each costing the same amount. Now he has $12. How much did each book cost?	✓	✓	✓
SVAMP	Granite-7B	Billy sells DVDs. He has 8 customers on Tuesday. His first 3 customers buy one DVD each. His next 2 customers buy 2 DVDs each. His last 3 customers don't buy any DVDs. How many DVDs did Billy sell on Tuesday?	Billy sells DVDs to 8 customers on Tuesday. His first 3 customers buy one DVD each, and his next 2 customers buy 2 DVDs each. His last 3 customers don't buy any DVDs. How many DVDs does Billy sell on Tuesday?	✓	✓	✓
ASDiv	Mistral-7B	It's Rachel's birthday. Her parents wanted her to have fun so they went to the circus that happened to be in town that day. Upon arriving at the circus, they went to the ticket booth and asked how much each ticket cost. If each ticket costs $44.00 and they bought 7 tickets, how much money did they spend on tickets?	Rachel's parents take her to the circus on her birthday. They buy 7 tickets, each costing $44.00. How much money do they spend on tickets?	✓	✓	✓
MultiArith	Qwen-7B	Will invited 9 friends to a birthday party, but 4 couldn't come. If he wanted to buy enough cupcakes so each person could have exactly 8, how many should he buy?	Will invites 9 friends to a birthday party, but 4 can't come. If he wants to give each person 8 cupcakes, how many cupcakes should he buy?	✓	✓	✓
GSM8K	LLaMA-8B	Jerome had 4 friends who came to visit him on a certain day. The first friend pressed on the doorbell 20 times before Jerome opened, the second friend pressed on the doorbell 1/4 times more than Jerome's first friend. The third friend pressed on the doorbell 10 times more than the fourth friend. If the fourth friend pressed on the doorbell 60 times, how many doorbell rings did the doorbell make?	Jerome has 4 friends visiting him, and the first friend rang the doorbell 20 times before Jerome opened it. The second friend rang the doorbell 1/4 times more than the first friend, the third friend rang it 10 times more than the fourth friend, and the fourth friend rang it 60 times. How many times did the doorbell ring in total?	✓	✓	✓
GSM8K	LLaMA-8B	Jam has three boxes full of pencils and 2 loose pencils which give a total of 26 pencils. If her sister, Meg, has 46 pencils, how many boxes do Jam and Meg need to store all their pencils?	Jam has three boxes of pencils and 2 loose pencils, which together total 26 pencils. Her sister, Meg, has 46 pencils. How many boxes do Jam and Meg need to store all their pencils?	✓	✓	✓

Table 5: Manual evaluation of rephrased questions. A checkmark indicates success for each criterion.