From Recognition to Reconstruction: Towards Robust Procedural Mathematical Reasoning

Fateme Mazdarani and Carlos Toxtli
School of Computing, Clemson University
Clemson, SC, USA
fmazdar@clemson.edu, ctoxtli@clemson.edu

Abstract

Mathematical reasoning systems should be consistent when the same concept is expressed in equivalent forms. This requires not only recognizing shared mathematical meaning but also identifying the representation changes and reconstructing a traceable route between forms. We study this problem through mathematical theorems, whose equivalent formulations preserve the same underlying concept while differing substantially in representation. In this setting, theorem recognition is only the first step: a robust system must also identify the intervening transformations and recover their order. We introduce an evaluation of robust procedural mathematical reasoning with three components: theorem identification, transformation identification, and ordered reconstruction. Using validated theorem–representation pairs from an existing corpus, we build a procedural-robustness dataset with typed operations, reference procedures, and matched contrasts that distinguish valid reformulations from invalid ones. We first measure closed-book performance of four open-weight models to establish a baseline, then evaluate retrieval-augmented generation (RAG) as support mechanism for the same task. Models identify theorem identity more reliably than they recover the transformations and ordered routes connecting equivalent representations. RAG improves every stage, but procedural reconstruction remains the main bottleneck. The framework aims to inspire more consistent and traceable mathematical systems, with potential applications in mathematical search, tutoring, autoformalization, and scientific discovery.

1 Introduction

The ability to reason reliably across changes in representation is fundamental to intelligent systems. One concept may be expressed through different symbols, structures, abstractions, or intermediate constructions, yet these surface differences should not change a system's conclusions. A robust system should recognize the shared meaning, identify how the representation has changed, and reconstruct a traceable connection between the different forms.

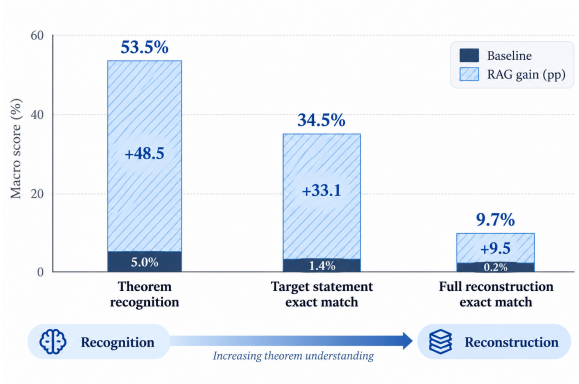

Figure 1: From recognition to reconstruction: macro-averaged scores across four models under increasingly strict evaluation criteria. Solid segments show baseline scores; hatched segments show RAG gains in percentage points. Bar-top labels report RAG scores.

Mathematical theorems provide a particularly demanding setting for studying this capability. A theorem may appear as an equation, a set relation, an optimization problem, or an auxiliary construction without changing its underlying content. Robustness therefore cannot be established through success on a single familiar notation. We formulate the broader capability as *robust procedural mathematical reasoning*. Given a reformulated theorem statement, a model must identify the original theorem, determine the transformations connecting the observed and reference representations, and order those transformations into a reconstruction route that reaches the reference statement. These are distinct capabilities: a model may correctly name the theorem while omitting a required operation, adding an unnecessary step, or placing the correct transformations in the wrong order.

This capability is relevant to mathematical search, tutoring, and autoformalization, where sys-

The Proceedings of the 4th Workshop on Mathematical Natural Language Processing (MathNLP 2026), pages 42–56
October 29, 2026 ©2026 Association for Computational Linguistics

tems must connect differently expressed statements while exposing the transformations that support the match (Yang et al., 2023; Wu et al., 2022). It may also make AI-assisted mathematical discovery easier to inspect and verify (Romera-Paredes et al., 2024; Trinh et al., 2024). Existing benchmarks commonly evaluate final-answer correctness in mathematical problem solving (Cobbe et al., 2021; Hendrycks et al., 2021) or success in formal proof generation (Polu and Sutskever, 2020; Zheng et al., 2021). These outcomes are important, but they do not directly show whether a model can explain how two equivalent representations are connected. A model may identify the correct theorem from surface cues while failing to recover the mathematical changes that preserve equivalence. Evaluating the reconstruction process can therefore reveal errors hidden by theorem-recognition accuracy alone and provide a more traceable account of model behavior. The TREAT dataset provides validated non-canonical reformulations of named theorems for evaluating recognition across representations (Mazdarani and Toxtli, 2026b). Yet, it does not evaluate the recovery process itself.

To support procedural evaluation, and address the gap, we extend a stratified subset of TREAT for procedural evaluation. We introduce a fixed ontology of transformation operations, assign typed labels to the changes required by each example, and provide an ordered reference route from the reformulated statement to the reference form. We also add matched contrasts that retain the overall representation while changing a mathematically decisive detail. The resulting procedural-robustness benchmark separates theorem identification from transformation identification, ordered reconstruction, and sensitivity to local validity.

Our evaluation proceeds in two stages. We first assess four open-weight models to establish their baseline performance on each component of the task. We then evaluate the same models using retrieval-augmented generation (RAG) under three conditions that provide theorem cards, transformation cards, or both. The theorem cards supply information about candidate theorems and their assumptions, while the transformation cards describe the operations used to connect different representations. This design allows us to test whether external theorem knowledge, procedural guidance, or their combination can address the weaknesses observed without retrieval. These gaps and design choices motivate the following research questions:

RQ1: How reliably do models identify the underlying theorem when it is expressed through a non-canonical but equivalent representation?

RQ2: How accurately do models identify the required transformations and reconstruct a route to the reference statement?

RQ3: To what extent does RAG improve theorem identification and procedural reconstruction?

RQ4: How does performance vary across representation mechanisms, route lengths, mathematical families, models, and matched validity contrasts?

2 Related Work

2.1 Formal knowledge and representation robustness

Language models can reproduce relational knowledge from pretraining (Petroni et al., 2019), but access remains sensitive to query form. ParaRel finds inconsistent factual predictions across paraphrases (Elazar et al., 2021), while negated and misleading probes expose failures on locally decisive relations (Kassner and Schütze, 2020). These results motivate testing whether access to the same object remains stable when its representation changes.

Mathematical benchmarks similarly vary surface form, symbols, templates, and related inputs (Zhou et al., 2024a; Mirzadeh et al., 2024; Zhou et al., 2024b). MATH Perturb and PutnamGAP extend controlled perturbation to advanced problems (Huang et al., 2025; Hao et al., 2025). TREAT pairs named theorems with equivalent formula-level representations and evaluates theorem identification (Mazdarani and Toxtli, 2026b). We use a stratified subset to ask the procedural question left open by recognition: can models reconstruct the representation changes linking an observed formulation to a reference statement?

2.2 Procedural mathematical reasoning

Most mathematical language-model evaluations emphasize answers, proofs, or formal derivations. TheoremQA studies theorem application (Chen et al., 2023a); NaturalProofs and NaturalProver evaluate grounded proof generation (Welleck et al., 2021, 2022); and autoformalization and theorem-proving systems operate in proof-assistant environments (Wu et al., 2022; Yang et al., 2023; Jiang et al., 2022). Process supervision and error localization further separate correct answers from correct derivations (Lightman et al., 2023; Zheng et al., 2024; Mazdarani and Toxtli, 2026a). Our

reconstruction sequence is not a theorem proof. It is a typed, representation-level account of how an equivalent statement maps to a reference statement under recorded assumptions. It connects to expression transformation (Edelmann and Kuncak, 2019) and operator-based formal tactics (Samoylov and Vosoughi, 2025), but targets traceable reconstruction of a known object rather than proof search.

2.3 RAG and structured-output evaluation

Retrieval-augmented generation conditions generation on retrieved context (Lewis et al., 2020). In mathematics, retrieved premises, theorems, and proofs support natural-language and formal proving (Welleck et al., 2022; Yang et al., 2023, 2025), while formula-text retrieval links symbolic expressions with explanatory context (Li, 2025). Here, the observed statement is the query, a deterministic retriever returns theorem or transformation cards, and the same model generates a structured reconstruction. RAG is therefore an enabling condition, not the object of optimization. Structured-prediction evaluation distinguishes complete-output exact match from substructure overlap (Yu et al., 2018; Chen et al., 2023b). We score statements and ordered procedures by exact match, transformation content by multiplicity-aware label F1 (Yang et al., 2019), and paired decisions by contrast-set consistency (Gardner et al., 2020; Johnson and Marasović, 2023). Execution-based evaluation is unavailable because the procedures do not share an executable formalism (Zhong et al., 2020).

3 Methods

3.1 Problem Formulation

Let y denote a theorem identity and r_y its designated reference statement. The source corpus provides a reformulation x validated as equivalent to r_y under recorded assumptions. We annotate a reference procedure

$$x = x_0 \xrightarrow{o_1} x_1 \xrightarrow{o_2} \cdots \xrightarrow{o_L} x_L = r_y, \quad (1)$$

where each o_j is a typed, item-specific representation rewrite. Given x and optional RAG context, the model returns ACCEPT, REJECT, or ABSTAIN. An accepted output contains a theorem prediction $\hat{y}$, reconstructed statement $\hat{r}$, and ordered operation sequence $\hat{\tau}$. These fields evaluate theorem identification, transformation identification, and ordered reconstruction. The reference statement r_y is the formulation selected by the source corpus as the common formulation. It is not assumed to be the unique form of the theorem. Similarly, the annotated procedure records one valid sequence of representation changes from x to r_y; other valid sequences may also exist. The procedure is intended for evaluating reconstruction and should not be interpreted as a proof of the theorem. For example, from $\delta_D(\chi'_l(K_{n,n}), n) = 0$, a successful output identifies Dinitz's theorem, applies distance removal, and reconstructs $\chi'_l(K_{n,n}) = n$. Appendix A.3 provides more examples.

3.2 Dataset Construction

TREAT contains 29,480 validated reformulations of 737 named results, with source and transformed statements, assumptions, metadata, and recovery descriptions (Mazdarani and Toxtli, 2026b). We select 1,400 reformulations from 306 theorem identities and map their recovery descriptions to a fixed operation ontology. Each procedure is oriented from the observed statement to the designated reference statement and split into atomic steps that record an operation identifier and its input and output expressions. Each reformulation is linked to its theorem identity, reference statement, transformation labels, and ordered reference procedure. The ontology contains 16 operations. Fourteen remove representation mechanisms, including distance encodings, set membership, optimization forms, and ordered pairs, while two normalize notation or relation direction. The full inventory appears in Appendix A.1. Each reformulation is also paired with a matched contrast that preserves the outer representation while changing one locally decisive component (Gardner et al., 2020). The four contrast types are target-expression perturbation, logical negation, relation-direction reversal, and type-matched replacement, with 350 examples of each (See Appendix A.1, Table 5).

The selected reformulations are stratified across five mathematical families and 20 procedure signatures. Each family contributes 280 items, each signature contributes 70, and no theorem contributes more than five. The reference procedures contain one, two, or three scored operations, corresponding to 980, 280, and 140 items, respectively. The full dataset contains 2,800 inputs: 1,400 reformulations and 1,400 matched contrasts. During evaluation, models receive only an opaque item identifier and the observed statement.

3.3 Experimental Design

We evaluate procedural reconstruction under four experimental conditions. The *baseline condition* uses the common task prompt, structured output contract, and operation inventory, without additional theorem or transformation evidence. We then evaluate whether retrieval-augmented generation (RAG) improves performance under three supported conditions: theorem RAG, transformation RAG, and combined RAG. The *theorem-RAG condition* provides cards containing theorem names, aliases, reference statements, assumptions, and domain metadata. The *transformation-RAG condition* provides cards describing reconstruction operations, applicability conditions, input and output types, and worked examples. The *combined-RAG condition* provides both types of cards. Comparing these conditions with the baseline allows us to measure the effects of theorem information, transformation guidance, and their combination.

RAG is implemented as a transparent and deterministic support mechanism because the experiment examines how external evidence affects procedural reconstruction rather than how retrieval itself can be optimized. For query q and document d, the retrieval score is

$$s(q, d) = \frac{|T(q) \cap T(d)|}{\sqrt{|T(d)|}}, \tag{2}$$

where $T(\cdot)$ contains lowercase alphanumeric tokens longer than one character. This reproducible lexical score requires no fitted corpus statistics (Robertson and Zaragoza, 2009). Each active index returns its top eight cards, while all other experimental settings remain fixed.

3.4 Models and inference

We evaluate GPT-OSS-20B, GPT-OSS-120B (OpenAI, 2025), GEMMA-4-31B (Google DeepMind, 2026), and LEANSTRAL-2603 (Mistral AI, 2026). Each model processes all 2,800 inputs in all four settings, yielding 44,800 predictions. Decoding uses temperature 0, top-$p = 1$, seed 20260723, and a 2,200-token output limit.

Raw responses are retained before parsing. A logged deterministic parser applies predefined formatting repairs, normalizes field names, and maps out-of-vocabulary transformation strings to `OTHER`. It does not alter mathematical expressions, theorem predictions, or substantive reconstruction content. The 1,045 outputs (2%) that remain invalid after repair stay in the denominator and receive zero task credit. All intended records are present and no backend failures occurred. The shared system prompt, user template, and structured output contract are provided in Appendix A.1.

3.5 Evaluation Metrics

Each model output contains a decision, theorem prediction, reconstructed statement, and ordered sequence of transformation labels. We use exact-match metrics to evaluate fully specified outputs and an F1 score to measure partial recovery of the required transformations. This follows the general structured-prediction principle of evaluating both the complete output and its recoverable components (Chen et al., 2023b).

3.5.1 Theorem and Reconstruction Metrics

Theorem recognition accuracy requires an ACCEPT decision and the correct theorem identity after alias normalization.

Target statement exact match additionally requires the reconstructed statement to match the designated reference statement r_y after deterministic presentation normalization. This metric evaluates recovery of the designated target statement independently of the predicted operation sequence.

Transformation-label F1 measures order-invariant overlap between the predicted and reference transformation labels. Let $m_o(\tau)$ denote the multiplicity of operation o in procedure τ, and define

$$M(\hat{\tau}, \tau) = \sum_o \min!\left(m_o(\hat{\tau}), m_o(\tau)\right). \tag{3}$$

Precision is $M/|\hat{\tau}|$, recall is $M/|\tau|$, and their harmonic mean is reported. An empty predicted sequence receives a score of zero. The comparison preserves multiplicity but ignores order. The score is gated by theorem correctness because the same operation labels may apply to different theorems. It gives partial credit for correctly recovered labels despite omissions, additions, or reordering, but does not establish that the predicted sequence is mathematically valid.

Full reconstruction exact match requires an ACCEPT decision, the correct theorem, an exact match with the target statement, and exact agreement with the complete ordered reference procedure. This is the strictest metric; each item contains one annotated reference procedure rather than

every possible valid route, the metric measures reproducible agreement with the annotation rather than uniqueness of the mathematical path.

By design, *full reconstruction exact match* cannot exceed *target statement exact match*, which cannot exceed *theorem recognition accuracy*. This ordering is therefore expected. The informative quantities are the sizes of the gaps between these metrics and how those gaps change from the baseline condition to the RAG-supported conditions.

3.5.2 Matched-Contrast Metrics

Contrast false-positive rate is the percentage of contrasts assigned ACCEPT.

Contrast-set consistency is the percentage of matched pairs for which the valid reformulation is accepted and its contrast is rejected. Contrast-set consistency is stricter than item-level classification accuracy because both decisions must be correct within the same pair.

3.5.3 Retrieval and Statistical Analysis

We evaluate RAG's retrieval metrics separately from task performance; *Theorem recall at* k records whether the gold theorem card appears among the top k retrieved results. *Transformation coverage* records whether retrieval contains at least one required transformation and whether it contains the complete required set. *Joint coverage* requires retrieval of both the gold theorem card and every transformation in the reference procedure.

All task metrics are first computed separately for each model. Aggregate results are macro-averaged across models. Primary differences between the baseline and RAG-supported conditions are estimated using 10,000-repetition hierarchical bootstrap resampling over models and theorem clusters or matched pairs (Saravanan et al., 2020). We report absolute percentage-point differences with 95% percentile intervals.

4 Results

4.1 Baseline Capability and the Reconstruction Gap

Table 1 reports the performance of each model under the baseline and three RAG-supported conditions. Under the baseline condition, macro performance is low across all positive-input metrics, indicating limited ability to identify the underlying theorem and reconstruct the route from a reformulated statement to its reference form. Theorem recognition accuracy exceeds target statement exact match, which in turn exceeds full reconstruction exact match. Although this ordering follows from the nested definitions of the metrics, the size of the declines is informative. Models may identify the theorem without recovering the designated target statement, and they may recover the target without reproducing the complete ordered procedure. Table 7 in Appendix A.2 decomposes combined-RAG outcomes into mutually exclusive success and failure categories.

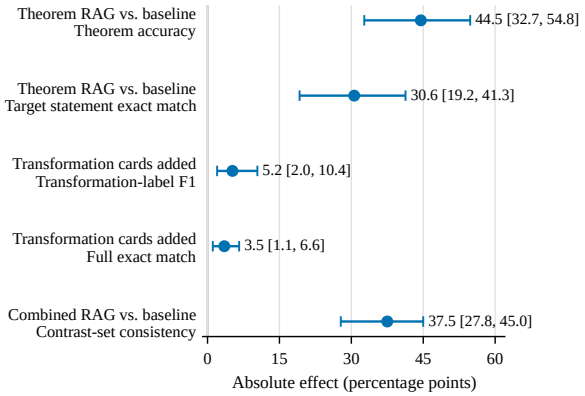

Figure 2: Primary RAG condition effects with 95% hierarchical bootstrap intervals. The first two rows compare theorem RAG with the baseline condition. The next two compare combined RAG with theorem RAG, isolating the effect of adding transformation cards. The final row compares combined RAG with the baseline condition.

4.2 Effects of RAG Support and Retrieval Coverage

The RAG conditions improve performance over the baseline, but the gains differ across stages of the task. Theorem-card RAG produces large gain in theorem identification: relative to the baseline condition, theorem recognition accuracy increases from 5.0% to 49.5%. Transformation cards provide complementary procedural support. Adding transformation cards to theorem RAG increases transformation-label F1 from 13.7% to 18.9%. Figure 2 demonstrate these effects.

Combined RAG produces the strongest macro-level results, reaching 53.5% theorem recognition accuracy, 34.5% target statement exact match, 18.9% transformation-label F1, and 9.7% full reconstruction exact match. Thus, retrieval provides substantial support for identifying the theorem and recovering the target statement, but considerably less support for reconstructing the complete ordered route.

Model	RAG setting	Thm. Acc.	Target EM	Trans. F1	Full EM	Contrast Cons.	Contrast FPR
gpt-oss-20B	No RAG	5.6	1.4	1.2	0.1	17.4	0.9
	Theorem RAG	48.6	30.8	6.7	2.4	46.9	2.1
	Transform. RAG	6.2	1.9	2.3	0.6	28.3	2.4
	Combined RAG	56.9	35.1	19.5	10.6	54.9	2.0
gpt-oss-120B	No RAG	8.4	2.1	2.5	0.2	19.6	1.6
	Theorem RAG	65.4	35.4	20.1	7.0	60.0	6.1
	Transform. RAG	8.5	2.0	3.1	0.9	29.6	3.9
	Combined RAG	67.9	36.0	23.5	9.9	64.9	4.6
Gemma-4-31B	No RAG	5.8	2.1	2.2	0.5	8.9	0.9
	Theorem RAG	57.1	47.1	20.2	11.9	51.6	5.9
	Transform. RAG	11.1	3.5	4.0	1.1	18.3	3.1
	Combined RAG	58.9	49.7	22.4	14.1	53.1	7.4
Leanstral-2603	No RAG	0.1	0.0	0.0	0.0	0.2	0.0
	Theorem RAG	27.1	14.5	7.7	3.3	21.4	5.7
	Transform. RAG	0.2	0.0	0.0	0.0	0.6	0.0
	Combined RAG	30.1	17.1	10.2	4.1	23.1	6.9
Macro average	No RAG	5.0	1.4	1.5	0.2	11.5	0.8
	Theorem RAG	49.5	32.0	13.7	6.1	45.0	5.0
	Transform. RAG	6.5	1.8	2.4	0.6	19.2	2.4
	Combined RAG	**53.5**	**34.5**	**18.9**	**9.7**	**49.0**	**5.2**

Table 1: Results for each model and experimental condition (%). The baseline condition provides no retrieved evidence, while the three RAG conditions provide theorem cards, transformation cards, or both. Macro averages are reported in the final four rows. Thm. Acc. denotes theorem recognition accuracy; Target EM, target statement exact match; Trans. F1, transformation-label F1; Full EM, exact agreement on the theorem, target statement, and complete ordered reference procedure; Contrast Cons., contrast-set consistency; and Contrast FPR, the false-positive rate on matched contrasts.

4.3 Retrieval Coverage Analysis

The retrieval coverage analysis separates failures to retrieve relevant evidence from failures to use available evidence. Under combined RAG, the gold theorem card appears among the top eight results for 76.4% of valid inputs. The complete set of required transformations is retrieved for 57.6%, while both the gold theorem and every required transformation are available for 42.9% of inputs, as shown in Table 2.

On the 601 examples with complete retrieved coverage, theorem recognition accuracy reaches 74.5%, target statement exact match reaches 49.2%, transformation-label F1 reaches 41.9%, and full reconstruction exact match reaches 22.0%. These are coverage-conditioned results on items for which the retriever succeeds, not oracle estimates for the full dataset. They nevertheless show that relevant RAG context can be used productively while leaving a substantial challenge in selecting, applying, and ordering the available information.

4.4 Performance Across Reconstruction Structures

Reconstruction difficulty increases with procedure length. Under combined RAG, full reconstruction exact match is 13.8% for one-operation procedures and zero for two- and three-operation procedures (Table 3), while theorem recognition, target recovery, and transformation-label F1 remain nonzero. However, the length groups are not matched for theorem identity or representation mechanism. They contain 295, 139, and 73 distinct theorem identities, respectively. One-operation procedures cover all 14 representation mechanisms; two-operation procedures cover distance, containment, set membership, and truth-table encodings; three-operation procedures cover only distance and set membership. Although the five mathematical families are equally represented within each length group, this balance does not control theorem or mechanism composition. These differences could partly explain the observed performance decline. The length breakdown therefore describes performance across the sampled procedure groups rather than isolating the effect of adding reconstruction steps.

Performance also differs across representation mechanisms. Distance, argmin, witness, and quantifier encodings show stronger transformation recovery. In contrast, truth-table, fiber, cylinder, and ordered-pair encodings often lead models to identify the correct theorem without predicting the annotated transformation sequence. In this balanced sample, variation across representation mechanisms is greater than variation across broad mathematical families. Appendix A.2 provides the length and mechanism profiles (Figures 3 and 4) and family-level results (Table 6).

The matched contrasts measure whether model decisions remain sensitive to locally decisive

(a) RAG retrieval coverage (%)

Gold card coverage	R@1	R@4	R@8
Theorem card	60.0	72.4	76.4
At least one transformation card	n/a	n/a	69.1
All required transformation cards	n/a	n/a	57.6
Theorem and all transformation cards	n/a	n/a	42.9

(b) Combined RAG conditioned on coverage (%)

Retrieved gold context	n	Thm. Acc.	Target EM	Trans. F1	Full EM
Neither	125	0.0	0.0	0.0	0.0
Transformations only	205	0.0	0.0	0.0	0.0
Theorem only	469	64.2	39.8	2.7	0.6
Complete	601	74.5	49.2	41.9	22.0

Table 2: RAG retrieval coverage and generation performance conditioned on coverage. The count in panel (b) is the number of unique valid inputs.

L	n	Thm. Acc.	Target EM	Trans. F1	Full EM
1	980	55.8	36.4	23.0	13.8
2	280	48.4	31.7	9.2	0.0
3	140	47.1	26.6	9.8	0.0

Table 3: Combined RAG by procedure length (%).

changes. Combined RAG increases contrast-set consistency from 11.5% to 49.0%, while the contrast false-positive rate increases from 0.8% to just 5.2%. Logical negation and relation-direction reversal are the most difficult contrast types, whereas target-expression perturbations and type-matched replacements are rejected more reliably.

5 Discussion

The results reveal a clear separation between theorem recognition and procedural reconstruction. Retrieval substantially improves access to the underlying theorem and reference statement, but complete recovery of the ordered transformation route remains difficult. The matched contrasts further show that consistency across valid reformulations must be balanced with sensitivity to small changes that invalidate the theorem mapping.

5.1 RQ1: Theorem identification is necessary but not sufficient

The baseline results show that non-canonical representations make even familiar theorem knowledge difficult to access. Theorem RAG substantially improves recognition, indicating that many reformulated statements can be linked to the correct mathematical concept when relevant information is available. The remaining gap between theorem recognition, target recovery, and full reconstruction is more important than the improvement in recognition alone. Identifying the theorem establishes what the observed statement represents, but it does not explain how that statement relates to the designated reference form. Models often recover the conceptual destination without recovering the transformations needed to reach it. This distinction persists in the model-level results, where performance rankings change across theorem identification, target recovery, reconstruction, and contrast consistency. Theorem recognition is therefore a necessary first step, but it is not a sufficient measure of robustness across mathematical representations.

5.2 RQ2: Route Recovery Is the Main Bottleneck

Models may recover the reference statement without recovering the complete sequence of transformations that connects it to the observed form. Transformation-label F1 shows that many failed exact reconstructions still contain partial procedural information: models often identify the main operation but omit a normalization step, add an unnecessary operation, or predict the wrong order. Longer-procedure groups exhibit lower exact-match performance, although this comparison is confounded by differences in theorem and mechanism composition. Table 3 shows that full reconstruction exact match falls to zero for two- and three-operation procedures, although theorem recognition, target recovery, and transformation-label F1 remain nonzero. Models therefore recover parts of longer routes but do not reliably reproduce every required step.

Performance also varies across mechanisms, suggesting that some reformulations make their inverse operations more visible than others. Detailed results appear in Appendix A.2, although the present evaluation cannot separate structural difficulty from model familiarity. Because each item contains one annotated reference route, full reconstruction exact match measures agreement with a reproducible scoring target rather than the uniqueness of the mathematical path. It therefore distinguishes reaching the correct endpoint from recovering a complete and inspectable route.

5.3 RQ3: RAG Supports but Does Not Solve Procedural Reconstruction

The retrieval sources provide complementary support. Theorem cards mainly improve theorem identification and target recovery, while transformation cards provide information about the operations connecting the observed and reference forms. Transformation cards are useful once theorem information is available, indicating that procedural guidance depends on a plausible theorem-level interpretation. Retrieval coverage separates failures to find relevant evidence from failures to use it. Table 2 shows higher performance when the gold theorem card and all required transformations are retrieved, yet full reconstruction remains difficult even under complete coverage. Access to the necessary evidence therefore does not guarantee correct selection, application, or ordering. Retrieval and reconstruction should consequently be treated as separate components. Retrieval can narrow the space of plausible theorems and transformations, but future work must determine whether the generated steps compose into a mathematically valid route.

5.4 RQ4: Robustness Requires Global Recognition and Local Validity

The matched contrasts test whether models remain selective when a reformulation is changed in a mathematically decisive way. Combined RAG improves contrast-set consistency, showing that retrieval helps models accept valid reformulations while rejecting their paired contrasts. However, the accompanying increase in false positives indicates that stronger theorem-level evidence can also make models more willing to accept invalid variants.

The contrast-type breakdown in Table 8 shows that logical negation and relation-direction reversal are the most difficult edits. These changes preserve most of the surrounding expression while altering a relation that determines whether the original theorem and reconstruction route remain applicable. Target-expression perturbations and type-matched replacements are rejected more reliably because they produce more visible changes to the mathematical structure.

This pattern highlights the need to combine global recognition with local validation. Identifying a strong match to a known theorem is not sufficient when a negated event, reversed inequality, or changed inclusion invalidates the connection. A robust system should therefore verify the locally decisive relations before accepting the theorem identity and proposed reconstruction route.

5.5 Broader Implications and Applications

The present results only establish benchmark behavior, yet inspires future studies. Procedural reconstruction can make mathematical systems more traceable by showing how an unfamiliar representation connects to a known result. In mathematical search, it can explain why a theorem expressed in different notation or structure is relevant. In tutoring and assessment, separating theorem identification, transformation recovery, and operation ordering can localize the source of an error. The same framework may support autoformalization, proof repair, and theorem proving by exposing the transformations needed to align nonstandard inputs with formal reference statements. It also motivates modular systems in which retrieval proposes relevant knowledge, reconstruction builds the route, and a symbolic or formal checker verifies each step. More broadly, exposing the transformations behind a proposed connection may make mathematical and scientific discovery more inspectable and testable than returning only a label or analogy (Udrescu and Tegmark, 2020; Romera-Paredes et al., 2024; Trinh et al., 2024).

6 Conclusion

We introduced robust procedural mathematical reasoning as the task of identifying a theorem, recovering the transformations between equivalent representations, and ordering them into a traceable route to a reference statement. We extended a stratified subset of validated theorem reformulations with typed operations, ordered reference procedures, and matched contrasts.

The results show that theorem recognition is substantially easier than procedural reconstruction. Retrieval-augmented generation improves theorem identification, target recovery, and transformation recovery, but complete reconstruction remains difficult even when the relevant evidence is retrieved. Reliable representation robustness therefore requires not only access to the correct theorem, but also the ability to select, apply, and order locally valid transformations.

Acknowledgment

This work was supported in part by the National Science Foundation under Award No. 2434704.

Limitations

This study provides a controlled evaluation of procedural reconstruction, and its conclusions are scoped to the benchmark, annotations, metrics, and experimental setup used here.

Each reformulation has one designated reference statement and one annotated reconstruction procedure, although other mathematically valid routes may exist. Full reconstruction exact match may therefore undercount valid alternatives, while transformation-label F1 ignores order and does not verify that the predicted operations compose into a valid route. Target statement exact match handles presentation-level variation but not unrestricted symbolic equivalence. Moreover, the procedures are not executable in a shared formal system, so operation preconditions, intermediate statements, and the use of theorem assumptions are not independently verified. Additional expert annotation or formal validation would strengthen these aspects.

The benchmark covers a stratified subset of TREAT, five mathematical families, 16 operations, and procedures of one to three steps, with fewer examples at longer lengths. The matched contrasts contain four controlled single-edit types. Consequently, the results may not extend to longer procedures, other mathematical domains, alternative transformation ontologies, multiple interacting edits, or failures involving implicit assumptions and domain restrictions. The subgroup analyses are descriptive and cannot fully separate structural difficulty from theorem familiarity or possible pretraining exposure. A human or expert baseline would also help calibrate the benchmark's absolute difficulty.

The experiments evaluate four open-weight models using one prompt, structured output schema, deterministic decoding configuration, and parser. The reported model averages should therefore not be interpreted as estimates of performance across the broader model population, and different prompting or decoding choices may change the results.

Finally, the RAG conditions use a reproducible lexical retriever rather than an optimized retrieval system. The theorem cards intentionally provide reference statements, so these conditions measure models' ability to use supplied evidence rather than independently recover all information. The effects of individual card fields are not isolated. Coverage-conditioned results are also associational, since examples with complete retrieval may be systematically easier. Future work should examine card ablations, stronger retrievers, broader model families, human evaluation, and symbolic or formal checking of reconstruction steps.

References

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023a. TheoremQA: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*.

Yunmo Chen, William Gantt, Tongfei Chen, Aaron Steven White, and Benjamin Van Durme. 2023b. A unified view of evaluation metrics for structured prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12868–12882. Association for Computational Linguistics.

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

Romain Edelmann and Viktor Kuncak. 2019. Neural-network guided expression transformation. *arXiv preprint arXiv:1902.02194*.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, and 7 others. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323. Association for Computational Linguistics.

Google DeepMind. 2026. Gemma 4 model card. Google AI for Developers.

Yuren Hao, Xiang Wan, and Chengxiang Zhai. 2025. An investigation of robustness of LLMs in mathematical reasoning: Benchmarking with mathematically-equivalent transformation of advanced mathematical problems. *arXiv preprint arXiv:2508.08833*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

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. *arXiv preprint arXiv:2502.06453*.

Albert Q. Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygozdz, Piotr Milos, Yuhuai Wu, and Mateja Jamnik. 2022. Thor: Wielding hammers to integrate language models and automated theorem provers. *arXiv preprint arXiv:2205.10893*.

Jacob K. Johnson and Ana Marasović. 2023. How much consistency is your accuracy worth? In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 250–260. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*.

Zichao Li. 2025. Formula-text cross-retrieval: A benchmarking study of dense embedding methods for mathematical information retrieval. In *Proceedings of the 3rd Workshop on Mathematical Natural Language Processing*, pages 124–133. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Fateme Mazdarani and Carlos Toxtli. 2026a. Beyond the answer key: Robustness evaluation of large language models for step-level mathematical verification. In *Proceedings of the 2026 IEEE International Conference on Machine Learning and Applications (ICMLA)*.

Fateme Mazdarani and Carlos Toxtli. 2026b. TREAT: Evaluating access to formal knowledge across equivalent mathematical representations. In *Proceedings of the 2026 International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Mistral AI. 2026. Leanstral: Open-source foundation for trustworthy vibe-coding. Mistral AI technical release.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. OpenAI technical report.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2463–2473. Association for Computational Linguistics.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475.

Elisaveta Samoylov and Soroush Vosoughi. 2025. Modeling tactics as operators: Effect-grounded representations for Lean theorem proving. In *Proceedings of the 3rd Workshop on Mathematical Natural Language Processing*, pages 168–175. Association for Computational Linguistics.

Varun Saravanan, Gordon J. Berman, and Samuel J. Sober. 2020. Application of the hierarchical bootstrap to multi-level data in neuroscience. *Neurons, Behavior, Data Analysis, and Theory*, 3(5):1–25.

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482.

Silviu-Marian Udrescu and Max Tegmark. 2020. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. NaturalProofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022. NaturalProver: Grounded mathematical proof generation with language models. *arXiv preprint arXiv:2205.12910*.

Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. *arXiv preprint arXiv:2205.12615*.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023. LeanDojo: Theorem proving with retrieval-augmented language models. *arXiv preprint arXiv:2306.15626*.

Pengcheng Yang, Fuli Luo, Shuming Ma, Junyang Lin, and Xu Sun. 2019. A deep reinforced sequence-to-set model for multi-label classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5252–5258. Association for Computational Linguistics.

Wenjie Yang, Ruiyuan Huang, Jiaxing Guo, Zicheng Lyu, Tongshan Xu, Shengzhong Zhang, Lun Du, Da Zheng, and Zengfeng Huang. 2025. Retrieval-augmented language models are mimetic theorem provers. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21301–21313. Association for Computational Linguistics.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2412.06559*.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. miniF2F: A cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.

Ruiqi Zhong, Tao Yu, and Dan Klein. 2020. Semantic evaluation for text-to-SQL with distilled test suites. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 396–411. Association for Computational Linguistics.

Yue Zhou, Yada Zhu, Diego Antognini, Yoon Kim, and Yang Zhang. 2024a. Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2804. Association for Computational Linguistics.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. 2024b. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. *arXiv preprint arXiv:2407.08733*.

A Reproducibility and Additional Results

A.1 Prompt and Outputs

All models and experimental conditions use the same system prompt, user template, structured output contract, and fixed operation inventory. The model receives the observed statement, the active evidence lists, and the available transformation identifiers, and returns one JSON object containing a decision, theorem prediction, reference statement, reconstruction sequence, and supporting evidence identifiers.

Listing 1: User prompt template. Bracketed fields are populated at inference time.

```
OBSERVED_STATEMENT_LATEX:
{{observed_statement_latex}}

THEOREM_EVIDENCE:
{{theorem_evidence_or_NONE}}

TRANSFORMATION_EVIDENCE:
{{transform_evidence_or_NONE}}

REGISTERED_OPERATION_IDS:
{{operation_id_list}}

Use exactly the same decision standard whether evidence is present or
       NONE.
Evidence cards are optional aids, not labels. If the statement is a
       valid
encoding, accept only with a valid theorem-route reconstruction. If it
       is not
valid, reject or abstain even when retrieved cards look related.

CRITICAL ROUTE RULES:
- route.operations must be an array of registered operation strings
       only.
- Do not include RESTATE_CANONICAL_THEOREM in route.operations.
- route.steps must have the same length and order as route.operations.
- route.operations[i] must equal route.steps[i].operation.
- Step indices must be 1, 2, 3, ... with no gaps.
- Each step must chain: step i output_latex should be step i+1
       input_latex.
- If the decision is REJECT or ABSTAIN, use route.operations=[] and
       route.steps=[].
- If a necessary operation is not registered, use OTHER and explain it
       in operation_free_text.
- Keep every justification and rejection explanation concise,
       preferably under 35 words.
- Do not include long derivations, prose proofs, or markdown.

Return exactly one JSON object with this top-level shape:
{
  "schema_version": "route-decoder-v2.0",
  "decision": "ACCEPT | REJECT | ABSTAIN",
  "theorem": {
    "id": null,
    "name": null,
    "canonical_statement_latex": null,
    "assumptions_used": [],
    "confidence": 0.0
  },
  "route": {
    "operations": [],
    "steps": [],
    "final_statement_latex": null,
    "minimality": "UNKNOWN",
    "confidence": 0.0
  },
  "rejection": {
    "reason": "NO_THEOREM_MATCH | NOT_EQUIVALENT |
       INVALID_TRANSFORMATION | CONTRADICTED_ASSUMPTIONS | ILL_TYPED |
       AMBIGUOUS | INSUFFICIENT_EVIDENCE | NONE",
    "failed_step_index": null,
    "explanation": ""
  },
  "evidence_used": {
    "theorem_card_ids": [],
    "transform_rule_ids": []
  }
}

For ACCEPT, rejection.reason must be NONE. For REJECT or ABSTAIN,
       theorem and
route may be null/empty, but all fields must still be present.

Produce the route-decoder-v2.0 JSON object now. Do not output markdown.
```

Route generation and validity. The `input_latex` and `output_latex` fields record the expressions associated with each predicted operation, and the contract instructs the model to connect consecutive steps. These fields make the proposed reconstruction explicit and available for inspection, but they are not formally executed or used by the automatic metrics to establish the mathematical validity of individual steps or the complete route. The procedure metrics instead evaluate the predicted operation labels and their ordering relative to the annotated reference procedure.

Implementation terminology. The released implementation retains the legacy JSON fields `canonical_statement_latex` and `route`, which correspond to the *reference statement* and *reconstruction sequence* terminology used throughout the paper. This naming difference is purely terminological and does not affect the prompts, model outputs, or scoring procedure.

Listing 2: System prompt used for all models and experimental conditions.

```
You are a mathematical route decoder. The observed statement may be
       either:
(1) a valid equivalence-preserving encoding of a canonical theorem, or
(2) an adversarial negative that must be rejected.

Do not assume the statement is valid. Use only transformations whose
       preconditions are satisfied. Retrieved theorem cards are
       hypotheses, not answers. Retrieved transformation cards are rules
       , not proof that a rule applies.

Decision policy:
- ACCEPT only when you can identify a theorem and give an ordered
       inverse route whose every step is valid and whose endpoint is the
        stated canonical theorem under the listed assumptions.
- REJECT when the statement is contradicted, ill-typed, does not match
       a theorem, or requires an invalid transformation.
- ABSTAIN only when the evidence is genuinely insufficient or ambiguous.


Route policy:
- Output only scored inverse operations. Do not include
       RESTATE_CANONICAL_THEOREM as a route operation.
- For every operation, provide its input and output expression so the
       route can be executed and checked.
- Use OTHER only when no registered operation applies, and describe it
       in operation_free_text.
- A theorem name without a valid route is not sufficient for ACCEPT.

Return exactly one JSON object conforming to route-decoder-v2.0. Do not
        output markdown, commentary, or text outside the JSON object.
```

Operation vocabulary. Table 4 summarizes the 16 operations used for scoring. The fallback label `OTHER` is assigned only when a predicted operation cannot be mapped to the fixed inventory.

Transformation type	Reconstruction operation
Representation removal	
Argmin or minimizer	Remove an argmin or minimizer wrapper.
Characterization	Recover the target from an equivalent characterization.
Containment	Recover a relation from set containment.
Cylinder	Recover pointwise equality from cylinder equality.
Distance	Recover equality or membership from zero distance.
Fiber or preimage	Recover equality from fibers or preimages.
Intersection	Remove a diagonal or graph intersection encoding.
Monotone map	Reverse a strictly monotone map applied to a relation.
Optimization	Remove an optimization, slack, or residual wrapper.
Ordered pair	Remove an ordered pair or product space encoding.
Quantifier	Recover the direct statement from a quantified characterization.
Set membership	Recover a relation from membership in a set.
Truth table	Remove a Boolean or truth table encoding.
Witness	Eliminate an existential witness encoding.
Notation and relation normalization	
Abbreviation expansion	Expand corpus shorthand into reference notation.
Relation normalization	Restore conventional direction or side ordering.

Table 4: Transformation types in the fixed reconstruction ontology.

Contrast type	Meaning-changing edit	n
Target-expression perturbation	Replace the target expression with a nearby incompatible expression.	350
Logical negation	Negate the locally decisive proposition under the same outer encoding.	350
Relation direction	Reverse an inequality, inclusion, or related binary relation.	350
Type-matched replacement	Replace an object with an alternative of the same mathematical type.	350

Table 5: Matched contrast construction. Each edit preserves the outer representation while changing one locally decisive component.

Statement normalization. Reconstructed statements are compared after deterministic presentation normalization. The normalizer applies Unicode normalization, standardizes common LaTeX spacing and delimiter commands, maps equivalent inequality and arrow commands to shared forms, removes redundant grouping braces, and standardizes simple superscript and subscript notation. It does not reorder terms, simplify expressions, invoke a computer algebra system, or test unrestricted symbolic equivalence.

A.2 Breakdowns and error analysis

Figures 3 and 4 report the combined-RAG profiles by reference procedure length and transformation. The family comparison in Table 6 is directly interpretable because every family contains the same mixture of the 20 transformation signatures.

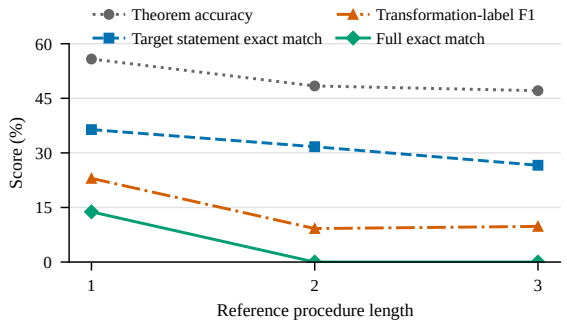

Figure 3: Combined RAG by reference procedure length.

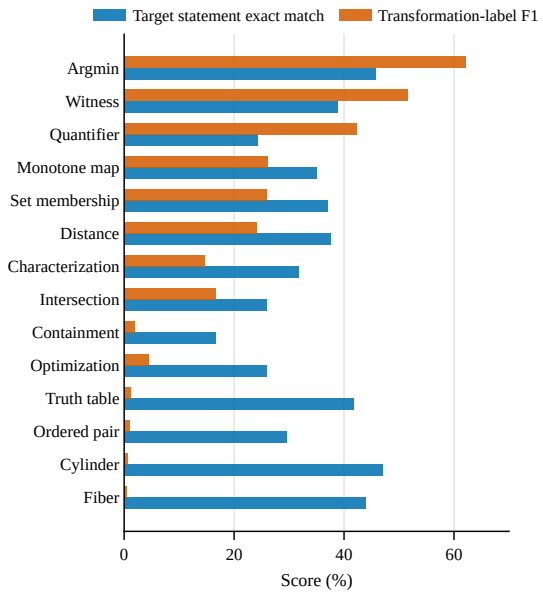

Figure 4: Combined RAG by representation mechanism. Transformation-label F1 is order invariant and gated by theorem correctness.

Family	n	Thm. Acc.	Target EM	Trans. F1	Full EM
Algebra and linear algebra	280	55.9	42.0	21.7	13.0
Analysis	280	47.2	24.5	16.7	6.7
Combinatorics and graph theory	280	59.6	37.4	22.6	11.1
Number theory	280	51.5	37.6	16.3	9.8
Probability and information theory	280	53.0	30.9	17.1	7.8

Table 6: Combined RAG by mathematical family (%).

Outcome	Share
Full reconstruction exact match	9.7
Target statement correct; procedure does not exactly match reference	24.8
Correct theorem; target statement mismatch	19.0
Accepted with incorrect theorem	0.2
Rejected or abstained	40.7
Malformed output	5.6

Table 7: Mutually exclusive outcomes on valid inputs under combined RAG (%). Categories are assigned in the displayed order.

Contrast edit type	n	Contrast FPR
Logical negation	350	7.9
Relation direction	350	7.7
Target-expression perturbation	350	2.4
Type-matched replacement	350	2.9

Table 8: Contrast false-acceptance rate by edit type under combined RAG (%).

A.3 Complete worked examples

Tables 9 and 10 present five complete examples with one-, two-, and three-operation routes. For each valid statement, the expected output is ACCEPT with the named theorem, designated reference statement, and ordered route. Each matched contrast is expected to be REJECT with an empty route.

Field	Example content
Example 1: Distance encoding (Dinitz theorem)	
Assumptions	$n \geq 1$, and δ_D is a metric, so $\delta_D(u, v) = 0$ iff $u = v$.
Observed statement	$x_0 = \delta_D\big(\chi'_l(K_{n,n}), n\big) = 0$.
Reference statement	$x_1 = \chi'_l(K_{n,n}) = n$.
Operation label	`REMOVE_DISTANCE_ENCODING`.
Intermediate states	$x_0 \xrightarrow{\texttt{REMOVE_DISTANCE_ENCODING}} x_1$.
Matched contrast	$\neg\big(\delta_D(\chi'_l(K_{n,n}), n) = 0\big)$.
Expected output	Valid: ACCEPT; theorem = Dinitz theorem; reference = x_1; route = [`REMOVE_DISTANCE_ENCODING`]. Contrast: REJECT; route = [ ].
Example 2: Order-cone encoding (AM–GM inequality)	
Assumptions	$x, y > 0$, $x \neq y$, and $C_{\leq} = \{(u, v) \in \mathbb{R}^2 : u \leq v\}$.
Observed statement	$x_0 = \left(\sqrt{xy}, \dfrac{x+y}{2}\right) \in C_{\leq}$.
Reference statement	$x_2 = \dfrac{x+y}{2} \geq \sqrt{xy}$.
Operation labels	`REMOVE_SET_MEMBERSHIP_ENCODING`, followed by `NORMALIZE_RELATION_DIRECTION`.
Intermediate states	$x_0 \to x_1 = \sqrt{xy} \leq \dfrac{x+y}{2}$ $\to x_2 = \dfrac{x+y}{2} \geq \sqrt{xy}$.
Matched contrast	$\left(\dfrac{x+y}{2}, \sqrt{xy}\right) \in C_{\leq}$, which reverses the decisive relation.
Expected output	Valid: ACCEPT; theorem = AM–GM inequality; reference = x_2; route = [`REMOVE_SET_MEMBERSHIP_ENCODING`, `NORMALIZE_RELATION_DIRECTION`]. Contrast: REJECT; route = [ ].
Example 3: Witness encoding (mean value theorem)	
Assumptions	$a < b$, f is continuous on $[a, b]$, and differentiable on (a, b).
Observed statement	$x_0 = \exists c \in (a, b)\ \exists z \in \mathbb{R} : f'(c) = z \wedge z = \dfrac{f(b) - f(a)}{b - a}$.
Reference statement	$x_1 = \exists c \in (a, b) : f'(c) = \dfrac{f(b) - f(a)}{b - a}$.
Operation label	`REMOVE_WITNESS_ENCODING`.
Intermediate states	$x_0 \xrightarrow{\texttt{REMOVE_WITNESS_ENCODING}} x_1$.
Matched contrast	$\neg\left(\exists c \in (a, b)\ \exists z \in \mathbb{R} : f'(c) = z \wedge z = \dfrac{f(b) - f(a)}{b - a}\right)$.
Expected output	Valid: ACCEPT; theorem = mean value theorem; reference = x_1; route = [`REMOVE_WITNESS_ENCODING`]. Contrast: REJECT; route = [ ].

Table 9: Complete task examples with one- and two-operation reconstruction routes.

Field	Example content
Example 4: Diagonal encoding (rank–nullity theorem)	
Assumptions	$T : V \to W$ is linear, V is finite-dimensional, and $\Delta_{\mathbb{N}} = \{(m, m) : m \in \mathbb{N}\}$.
Observed statement	$x_0 = \big(\mathrm{rk}(T) + \mathrm{null}(T), \dim V\big) \in \Delta_{\mathbb{N}}$.
Reference statement	$x_2 = \dim(\mathrm{im}\, T) + \dim(\ker T) = \dim V$.
Operation labels	`REMOVE_SET_MEMBERSHIP_ENCODING`, followed by `EXPAND_ABBREVIATION`.
Intermediate states	$x_0 \to x_1 = \mathrm{rk}(T) + \mathrm{null}(T) = \dim V$ $\to x_2 = \dim(\mathrm{im}\, T) + \dim(\ker T) = \dim V$.
Matched contrast	$\big(\mathrm{rk}(T) + \mathrm{null}(T), \dim V + 1\big) \in \Delta_{\mathbb{N}}$.
Expected output	Valid: ACCEPT; theorem = rank–nullity theorem; reference = x_2; route = [`REMOVE_SET_MEMBERSHIP_ENCODING`, `EXPAND_ABBREVIATION`]. Contrast: REJECT; route = [ ].
Example 5: Distance-to-diagonal encoding (Pythagorean theorem)	
Assumptions	$a, b, c > 0$ are the side lengths of a right triangle, $s := a^2 + b^2$, and $\Delta_{\mathbb{R}} = \{(u, u) : u \in \mathbb{R}\}$.
Observed statement	$x_0 = \mathrm{dist}\big((s, c^2), \Delta_{\mathbb{R}}\big) = 0$.
Reference statement	$x_3 = a^2 + b^2 = c^2$.
Operation labels	`REMOVE_DISTANCE_ENCODING`, followed by `REMOVE_SET_MEMBERSHIP_ENCODING`, followed by `EXPAND_ABBREVIATION`.
Intermediate states	$x_0 \to x_1 = (s, c^2) \in \Delta_{\mathbb{R}}$ $\to x_2 = s = c^2$ $\to x_3 = a^2 + b^2 = c^2$.
Matched contrast	$\mathrm{dist}\big((s, c^2), \Delta_{\mathbb{R}}\big) = 1$.
Expected output	Valid: ACCEPT; theorem = Pythagorean theorem; reference = x_3; route = [`REMOVE_DISTANCE_ENCODING`, `REMOVE_SET_MEMBERSHIP_ENCODING`, `EXPAND_ABBREVIATION`]. Contrast: REJECT; route = [ ].

Table 10: Complete task examples with two- and three-operation reconstruction routes.