MATHEMATICA INCOGNITA: CHARACTERIZING OPEN-ENDED MATHEMATICAL EXPLORATION IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mathematics has long depended on problem-solving as a guidepost for structuring and measuring progress—a methodology that is strongly reflected in the development of mathematical AIs. However, open-ended exploration, as well as the related activities of theory building and asking new research questions, remain under-studied capabilities of language models. In order to evaluate open-ended exploration of language models, we develop a suite of problem-solving tasks within equational theories—sets of equational axioms that define an algebraic structure. This area of math is particularly useful for model evaluation due to the ease of instantiating diverse, self-contained theories that are rarely described in literature, as well as the possibility of generating problems within these theories which remain difficult for frontier reasoning models without external tool use. Our experimental tasks evaluate two desiderata of successful explorations: future utility and idea diversity. As a proxy for future utility, we measure the improvement in problem-solving performance when models are provided with an exploration document, and observe notable but uneven benefits from exploration across models and theories. To measure idea diversity, we implement a limited automated reasoning querying interface for theory explorations that enables unambiguous comparison of research ideas across model runs, and observe a common core and long tail of ideas across models. As language models gain a greater role in solving research-grade math problems, studying their behavior during open-ended exploration will become increasingly important for comprehensive understanding of their research capabilities.

1 INTRODUCTION

The ambition to automate mathematical reasoning has long been intertwined with the broader goal of describing all aspects of learning and intelligence in terms precise enough to computationally simulate (Turing, 1950; McCarthy et al., 1955). Although LLM-based agentic research systems have recently made dramatic strides in mathematical problem-solving, with notable breakthroughs in long-standing open questions (Alon et al., 2026; Alpöge & Buckmaster, 2026; OpenAI, 2026), these advances do not give a comprehensive account of model capabilities across the range of mathematical intellectual activity. In particular, model capabilities in open-ended exploration and theory-building remain understudied and poorly understood.

A notable challenge in evaluating open-ended mathematical exploration capabilities is the lack of a standardized, quantifiable metric. In order to ground our study, we focus on two desiderata of an exploration's outputs: utility in solving unencountered problems and idea diversity. These exploration outputs may take the form of new constructions or conceptual frameworks, as in theory-building, or of new questions and solutions within an existing theory (Gowers et al., 2000; Mishra et al., 2025). We identify these desiderata because exploration outputs with future utility or a novel perspective help other mathematicians advance their own explorations. In this way, exploration drives progress in the field, which in turn supports further exploration in a self-reinforcing cycle.

Based on this perspective of successful mathematical explorations, we introduce a framework for evaluating open-ended exploration by language models that measures our desiderata of future utility

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

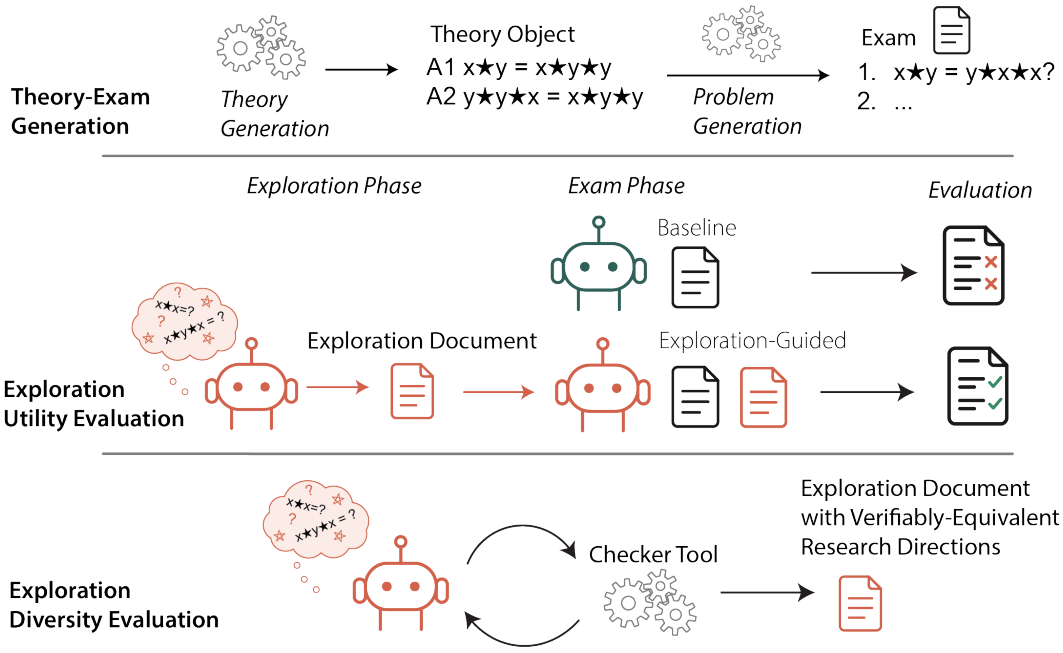


Figure 1: Overview of open-ended mathematical exploration evaluation. During theory-exam generation, we instantiate equational theories—equations that define an algebraic structure—together with difficult exams about each theory object using purely automated methods. In our first experimental task, we evaluate *exploration utility* achieved by language models based on the usefulness of open-ended theory exploration documents. This usefulness is judged based on the exam performance improvement of an exploration-guided model relative to a naive baseline model. In our second experimental task, we evaluate *exploration idea diversity* of language models. We implement a checker tool that offers models the ability to verify interesting equations and countermodels. The limited format of the checker tool interface allows for unambiguous comparison of research directions across model exploration sessions.

and idea diversity through experimental tasks. We ground our framework within the space of equational theories—sets of equations that define an algebraic structure—because this setting makes it easy to automatically generating exotic equational theory objects and difficult problems for many of them.

First, we instantiate pairs of equational theories and exams using various computational and automated reasoning techniques. Next, we prompt language models to explore the individual theories and develop their own research directions. The utility of the resulting exploration documents is measured by the improvement in a language model's exam performance when given the document, compared with a baseline in which it is not provided. Furthermore, we assess idea diversity by measuring overlap in models' research plans across repeated exploration runs, during which models are given access to rudimentary automated reasoning tools.

Although previous studies have formulated the task of automated mathematical exploration, each of these defined a setting in which programs build theories from scratch. One body of work defined programs that apply production rules to derive new concepts according to interestingness heuristics, either defined a priori or learned (Lenat, 1976; Colton, 2012; Tsoukalas et al., 2025). Another body of work posed theory-building as a game with intrinsic rewards, which was implemented via a conjecturing-proving loop that solved increasingly difficult conjectures (Poesia et al., 2024). These studies aimed to show that systems could discover rich, previously unseen mathematical structure. The systems therefore were launched *tabula rasa*, with access only to the most basic concepts As such, our study provides novel insights on the open-ended exploration of highly-capable language models that have already learnt the breadth of human mathematics.

Towards the goal of evaluating mathematical explorations of language models, we make the following contributions:

- We define a framework for studying mathematical intuition and theory-building capabilities in language models based on automated generation of equational theories and accompanying exams.
- We apply this framework to a two-phase explore–exam task that evaluates frontier models for their capabilities in generating useful open-ended explorations.
- We observe notable yet uneven benefits on model exam performance from prior exploration, demonstrating that language models have not solved evaluation of future utility during open-ended exploration.
- We implement an alternative exploration setting with a weak checker tool that enables verifiable comparison of idea diversity during open-ended discovery and report preliminary results showing a common core and long tail of ideas on repeated exploration runs.

2 Related Work

Problem-solving Math-AI systems. The field of AI for math has focused greatly on developing AI systems that can solve increasingly difficult math problems. Notable benchmarks include AIME and IMO problems formalized in miniF2F (Zheng et al., 2022), undergraduate competition problems in PutnamBench (Tsoukalas et al., 2024), and math research lemmas in First Proof (Abouzaid et al., 2026). Georgiev et al. (2025) described the use of the AlphaEvolve coding agent to explore the construction of solutions to a set of problems in fields ranging from combinatorics to number theory. AI systems have recently been used to resolve longstanding research problems such as the Erdős unit distance conjecture (Alon et al., 2026) and progress towards a candidate solution for Navier-Stokes (OpenAI, 2026; Alpöge & Buckmaster, 2026).

Automated theory-building. Lenat's AM used predefined heuristics to guide the construction and investigation of elementary mathematical concepts (Lenat, 1976). Colton's HR system combined production rules for concept formation with conjecture generation from examples, interestingness measures, and automated tools for proving or refuting conjectures (Colton, 2012). More recent work in this space has leveraged transformer-based models to learn criteria for directing search. Minimo posed theory building as a game with intrinsic rewards, which was implemented via a conjecturer-prover system that jointly learned to conjecture and prove theorems from axioms (Poesia et al., 2024). FERMAT implemented a symbolic environment in which an LLM-based evolutionary algorithm synthesized interestingness measures that guide the application of production rules (Tsoukalas et al., 2025). A separate line of recent research aimed to develop mathematical exploration using formal methods (Johansson & Smallbone, 2021).

Equational Theories Project. The Equational Theories Project (ETP), which ran from September 2024 to April 2025, had the goal of determining the implication graph of single equational theories of order at most 4 (Bolan et al., 2025). The motivation of this project was to demonstrate a framework for large-scale collaboration across both professional and amateur mathematicians with machine assistance. Over the course of the project, most machine-assisted results were derived from automated theorem provers rather than LLMs. Following this competition, the Foundation for Science and AI Research published an AI benchmark for solving these equational implications (SAIR Foundation, 2026). In contrast, our work instantiates 2-axiom equational theories; develops a wider suite of problems around these theories; and uses the exams as a means to probe open-ended exploration, rather than as a raw evaluation of problem-solving ability.

3 Equational Theories as a Microcosm of Math

Here we introduce equational theories, describe automated methods for instantiating them and writing problems about them, and explain how several of their properties make them a useful model for studying general mathematical activity.

```
variable ::= x | y | z
term     ::= variable | (term ⋆ term)
axiom    ::= term = term

For an axiom s = t, require:
  nodes(s), nodes(t) <= 5
  depth(s), depth(t) <= 2       (variables have depth 0)
  Vars(s) = Vars(t)
  2 <= |Vars(s)| <= 3
  s and t are syntactically distinct
```

Listing 1: Grammar for candidate axioms over one binary operation and additional conditions.

3.1 PRELIMINARIES

Equational theories are a syntactic description of certain algebraic structures. Algebraic structures, in turn, are fundamental objects of study in mathematics which consist of an interpreted set S and a collection of interpreted operations on S. Additional background is supplied in Appendix A.1.

An equational theory is specified by three components: a set, a collection of operations, and equational axioms which specify the behavior of the operations over the set. Notably, equational theories are limited to equational axioms and universally quantified variables.

To conclude this section, we introduce several useful definitions. The *signature* of an equational theory specifies the available operation symbols together with their arities, or the number of inputs of each operation.

Any algebraic structure over the theory's signature that satisfies all of the axioms is a *model* of the theory; in our example, these are the groups. Finite models can be described through model tables that specify the output of each operation for every possible combination of elements in the set.

Axioms can be used to derive new equations, or equational consequences, by applying axioms. These consequences hold in every model satisfying the axioms, which is called being *entailed* by the theory.

Finally, it is useful to note that entailment and non-entailment admit different forms of evidence. Entailment is proven through sequences of equations created by applying the axioms. On the other hand, non-entailment is proven through a *countermodel*: a counterexample model of the axioms A with an assignment under which either side of the equational statement differs.

3.2 INSTANTIATING THEORY OBJECTS

Equational theories represent a huge space of theory objects that can be programatically generated, thereby offering a highly convenient testbed for evaluating language model performance on manipulating and understanding mathematical theories.

Generating axioms. An equational theory's signature specifies the operation symbols and their arities, which are then used to write well-formed expressions within the theory. This information can be translated directly into production rules of a grammar for equational axioms, as shown in Listing 1. From this grammar, one can then generate arbitrary equations recursively by repeatedly applying the production rules.

We confined our focus to a small subset of equational theories for computational tractability as well as for compact, easily-understandable presentation of axioms. We generated axioms over a signature consisting of a single binary operation and defined the grammar over a set of three variables $\{x, y, z\}$. We further restricted ourselves to short, regular axioms in which each side has at most five syntax-tree nodes and depth at most two, with the same variable set on both sides. Finally, we instantiate two-axiom equational theories by selecting pairs of identities. Equation 1 demonstrates examples of the resulting axioms.

Example axiom shapes considered in our study:

$$\begin{aligned} x \star y &= y \star x \\ x \star y &= x \star (x \star y) \\ (x \star y) \star z &= (x \star z) \star y \end{aligned} \tag{1}$$

Selecting equational theories. Although specifying the grammar allows us to write arbitrary theories, this set contains various redundancies and degeneracies, such as a theory being another's mirror, or a second axiom being entailed by the first. Furthermore, we want to instantiate theories that are more unfamiliar to support the difficulty of our task, so we filter for associativity which is prevalent in well-known algebraic structures such as groups and groupoids. Finally, we want theories that contain enough structure to derive many new consequences, so we filter for a minimum number of consequences and a minimum number of rewrite steps that continues to derive new ones. We describe all of these considerations in Table 3.

3.3 Generating theory exams

Equational theories have further appeal as a testbed for language model evaluation due to the ease of using automated techniques for continuously generating new problem statements that are difficult for frontier models and not present in previous instantiations of the task. We introduce a procedure to generate several problem types that probe different types intuitions within an equational theory.

Problem generation Our problem format asks whether an identity is entailed by the axioms or additional premises without requesting a proof or countermodel. The response expected from a language model consists of a single JSON object: `{"answer":"follows"}` or `{"answer":"does_not_follow"}`. The proving machinery we use for writing and certifying correctness of problems consists of symbolic rewriting implemented in Python, the automated theorem prover Vampire for proving entailments, and the SMT solver cvc5 for searching for satisfying interpretations/countermodels (Kovács & Voronkov, 2013; Barbosa et al., 2022).

Based on our question format, we have two main paradigms of deriving questions: a bounded, breadth-first rewriting search to derive entailed consequences, as well as identity proposal paired with a combined solver that determines if the proposal is entailed or refuted. The combined solver consists of bounded rewriting search, followed by a Vampire entailment check, followed by cvc5 countermodel search depending on if the previous check fails to resolve an answer. All of these checks may fail, in which case the problem candidate is rejected as unresolved. Accepted questions come with an answer of entailment or refutation that is certified by rewrite steps, Vampire entailment, or a countermodel.

In order to cover a range of possible mathematical intuitions, we write problems of a variety of formats described in Table 1. Further implementation details of each of these problem types are described in Appendix A.4.

A final consideration at this step is whether or not the problems meet a certain level of difficulty in order to develop an evaluation that can show improved performance. As such, we filter out problems for which entailment can be proved in less than two rewrite steps or refuted with a countermodel of size 2.

Exam selection. Once we have generated problems for a given theory, we assemble them into an exam, requiring a minimum number of problems per problem format. Furthermore, the exam as a whole, as well as each of the problem types, required a balance of half entailed/half refuted answers to prevent an above chance strategy that didn't require math knowledge, or to bias a solver strategy towards proof vs. countermodel solving strategies. One exception was the basic properties question, which accepted an entailed/refuted imbalance offset by additional equational question(s) as the balance of basic properties cannot be enforced on a theory.

Under these procedures, we set a minimum number of problems to generate and do not continue evaluating a theory for which these methods do not create an acceptable exam. Furthermore, the procedures described attempt to select for problem difficulty but do not guarantee it. As such, we

Table 1: **Problem types.** In order to probe a range of mathematical intuitions within a theory, we generate several formats of problems.

Problem type	Typical question	Theory intuition
Basic properties	Does $(x \star y) \star z = x \star (y \star z)$ hold?	Familiar algebraic laws
Equational consequences	Does $(x \star (x \star z)) \star x = (z \star z) \star ((z \star z) \star x)$ hold?	Knowledge of theorems entailed by axioms
Additional assumptions	Additionally assume $(x \star y) \star z = x \star (y \star z)$ universally. Does $x \star y = y \star x$ hold?	Consequences under stronger premises
Axiom ablation	Omit A_2 and assume only A_1. Does $(x \star y) \star z = x \star (y \star z)$ hold?	Which conclusions depend on which axioms
Derived structure	Define $u(x) = x \star x$ and $p(x, y) = x \star (x \star y)$. Does $u(p(x, y)) = p(u(x), u(y))$ hold?	Behavior of derived operations

implement an additional screening step based on performance of GPT-5.6 Sol (medium) on the exam given a budget of 2,000 tokens per problem. If the performance is above 60%, we do not retain the exam and its associated theory in our task. An example of an automatically-generated exam is given in Appendix A.5.

4 EXPERIMENTS

Based on the previously-described theory objects and their generated exams, we go on to develop a set of experiments that probe the theory exploration capabilities of frontier models. Within this study, we consider GPT-5.6-sol, Claude-Sonnet-5, and GLM-5.3, all queried through Open Router and running with medium effort.

4.1 FUTURE UTILITY OF EXPLORATION

Our first experimental task evaluates an LLM's ability to generate an exploration of an equational theory that will be useful for solving future unspecified problems. As a proxy for this future utility, we measure the performance improvement on an unseen exam of a model instance provided with an exploration document relative to a naive baseline.

Exploration phase. We prompt language models to explore equational theories without supplying specific research directions or problems to guide the model outputs. Importantly, we do not indicate that the outputs will be used for any downstream task. The text of the prompt is provided in Listing 3. We then compress the resulting document to a target of 500 tokens using a query to the same model family in order to roughly equalize exploration lengths across model families and runs.

Exam phase. We run two conditions of model instances for comparison: a naive baseline which sees the theory for the first time together with an exam question, and an explorer-guided condition which is provided with a document from the exploration phase. In order to prevent theory exploration within a problem solution from affecting the rest of the exam, each question is submitted to the model under evaluation in a separate query.

We evaluate performance based on the number of questions a model instance is able to correctly answer at a given per-question token generation threshold, up to a maximum of 10,000 tokens. Each model family and exploration condition is evaluated over 3 separate runs and 21 different theories.

Language models generate useful explorations. All of our evaluated models demonstrated improvements in exam performance when provided with an exploration document compared with the baseline, as depicted in Figure 2. These results show that language models are able to discover useful concepts within a mathematical theory even without a specified task.

In the case of GPT-5.6, the model is able to solve more problems with a much lower token threshold, whereas at higher token thresholds, the baseline model is able to achieve parity, with both close to

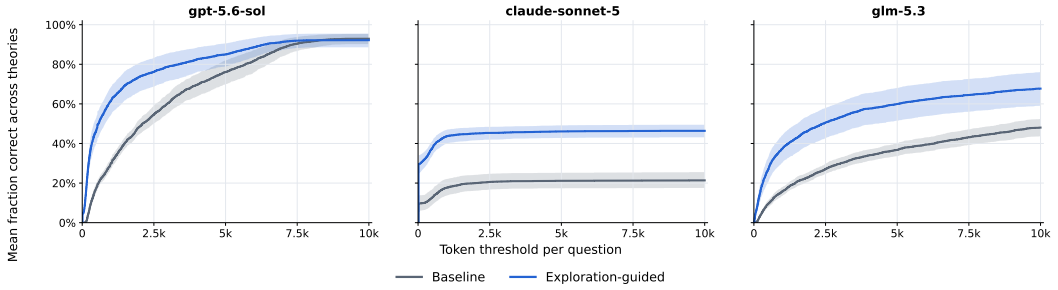


Figure 2: Accumulation curves of correctly-answered questions averaged across theories and runs. 95% CI derived from a hierarchical bootstrap over 3 runs within 21 theories.

solving 100% of the problems. On the other hand, Claude Sonnet 5 and GLM-5.3 both exhibit constant performance gains across all token thresholds and lower final performance on the exam problems by the maximum token budget.

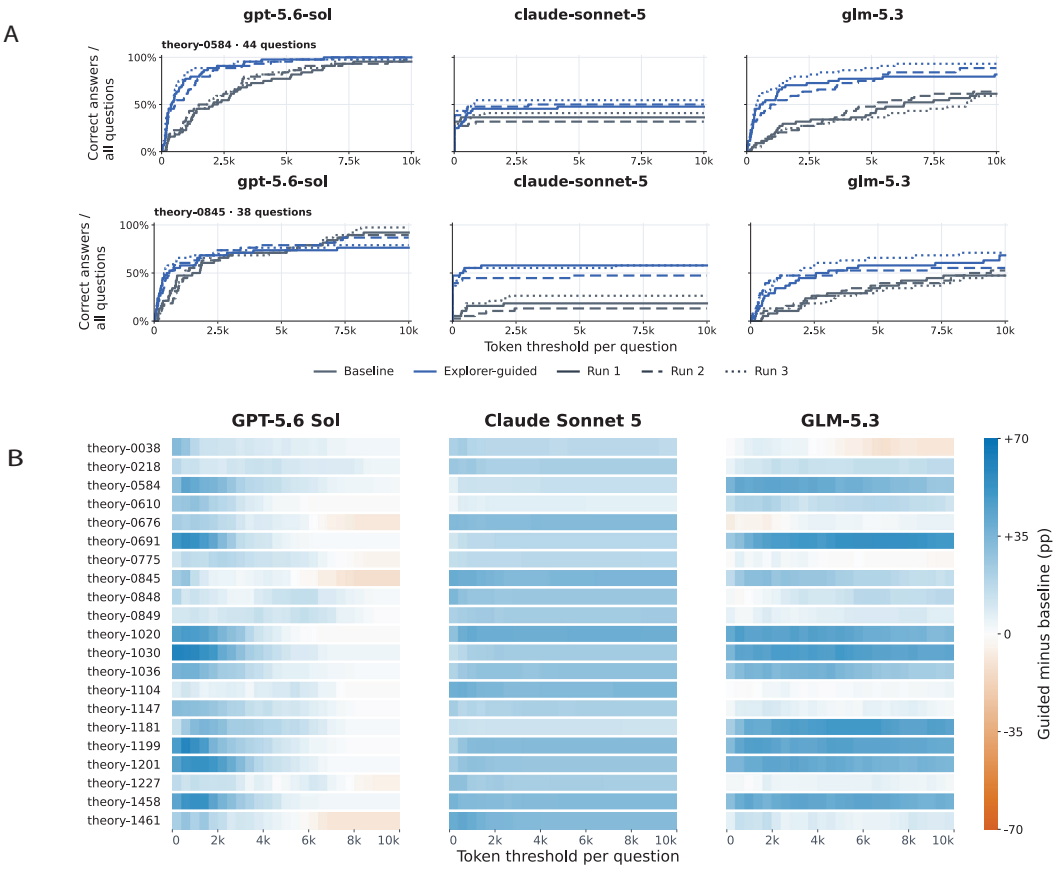


Figure 3: (A) Example single-theory accumulation curves of correctly-answered questions. (B) Heatmap of exploration-based performance improvement across theories and token thresholds.

Exploration utility varies across models, theories, and problem types. Although we find that exploration is beneficial for problem solving on unseen exams, the utility of these explorations varies across models, theories, and problem types. When plotting accumulation curves on individual theories, we find that on some theories, explorations are helpful, whereas on others, exploration can

actually hurt exam performance, as shown in Figure 3A. The theories over which these explorations help or hurt differ across models, as depicted in Figure 3B.

To better understand the impact of the exploration document on the exam solution generation of different language models under varying conditions, we studied difference in performance across proof vs. countermodel questions as shown in Table 2. We find that exploration provides a stronger improvement to countermodel performance for GPT-5.6-Sol and Claude Sonnet-5. However, Claude's performance improvement may be driven by its substantial increase in tendency to answer with refutation when provided with an exploration document. In addition, we studied differences in performance across problem types as shown in Table 9 and found that exploration documents help the most on basic problems, equational consequence problems, and derived structure problems.

Table 2: Mean accuracy and response proportions at a 2k token threshold. Base and Guided denote baseline and exploration-guided conditions. Invalid and truncated answers are not included in the tally.

	GPT-5.6 Sol (%)		Claude Sonnet 5 (%)		GLM-5.3 (%)	
Answer	Base	Guided	Base	Guided	Base	Guided
Accuracy						
Proof	57	74	19	39	36	60
Countermodel	40	74	21	51	12	35
Response proportions						
Proof	28	37	10	23	20	33
Countermodel	20	37	16	40	8	19

4.2 Idea diversity of exploration

Our second experimental task evaluates an LLM's ability to generate diverse ideas while exploring an equational theory. To formulate this quality in an easily-verifiable manner, we introduce a limited checker tool interface to automated reasoning capabilities for models to use during their exploration sessions. We then evaluate the diversity of the model queries to this checker tool over repeated runs.

Checker-assisted exploration. We implement a limited checker tool so that model explorations specify well-defined mathematical objects for their study, which can then be used to unambiguously compare the ideas explored within a natural language transcript.

The checker tool interface requires the language model to specify the exact mathematical artifact for test. There are two types of queries it can make: `check_entailment`, which submits one equation to cvc5 bounded proof and finite-model search, and `check_countermodel`, which submits a model table and a statement to a model checker. The first query returns `proved`, `refuted_entailment`, or `unresolved`, whereas the latter returns `refuted_countermodel` if the countermodel is successful and `invalid_countermodel` otherwise. In this setting, the exploration prompt in Listing 3 is extended with basic formatting instructions for use of the checker tool. Any queries that do not conform to the specified format return unresolved.

Evaluating idea diversity across repeated exploration runs. In order to evaluate the diversity of ideas that a language model studies during open-ended exploration, we sample 25 checker-assisted exploration sessions per model over 5 randomly-selected theories. We then study the recurrence of the same checker queries across model runs on the same theory.

Language model exploration ideas have a common core and a long tail. When running exploration sessions with the checker interface, we find that language models query the interface and regulate whether they do more or less investigation using this tool, as shown in Figure 4A. On average, different model classes query the checker a similar number of times. In contrast, model families

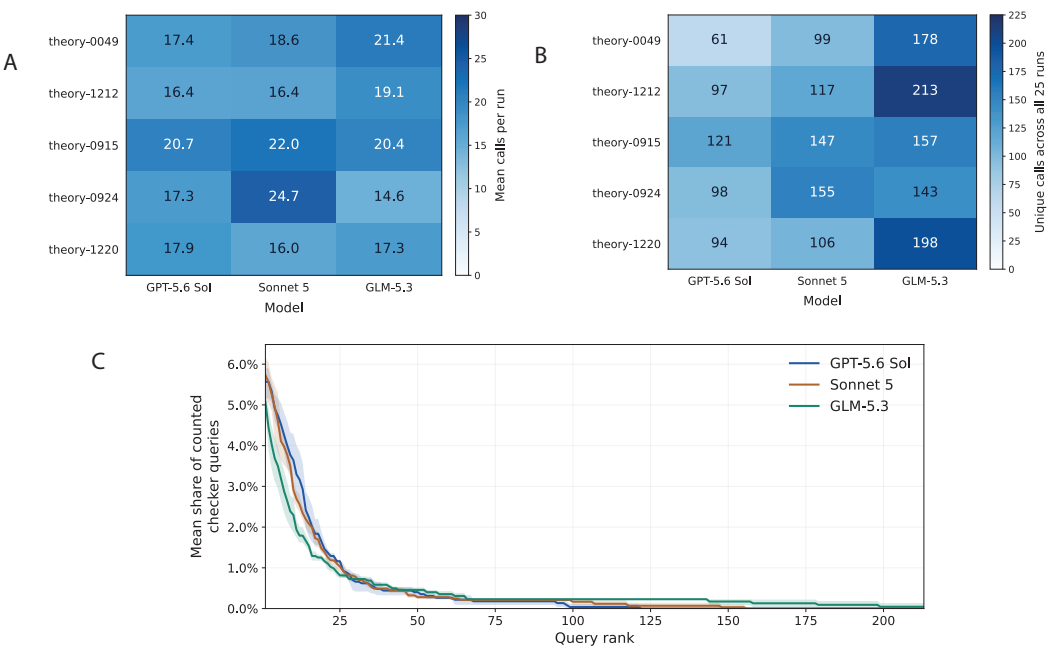


Figure 4: (A) Average number of checker calls per model on each theory across 25 runs. (B) Total distinct queries sent to the checker across all runs. (C) Average rank-frequency plot of checker calls averaged across theories. 95% CI calculated using a bootstrap over theories.

exhibit notable differences in checker query diversity as measured by number of unique queries (Figure 4B), with GLM-5.3 generating the most diverse set of queries. This trend is also reflected in the longer tail of queries seen in Figure 4C, which plots the percentage of runs in which a query occurs, ordered by most-commonly queried, and averaged across 5 theories.

Overall, language model research directions as represented by queries to the checker tool were characterized by a common core of queries and a long tail of less common queries. In order to further understand the quality of this diversity, it would be relevant in future work to characterize the usefulness and interestingness of these long tail queries, along with their justification in the reasoning trace.

5 Discussion

We introduced a framework for studying open-ended mathematical exploration in frontier language models based on exploring and answering questions about equational theories. Based on this framework, we evaluated the ability of three of these frontier models to generate useful information and produce diverse explorations. We found that although models demonstrate a benefit on unseen tasks after open-ended mathematical exploration, these benefits are unevenly distributed across theories, models, and question types. As such, open-ended exploration is a capability in language models that can be improved.

A notable future direction for this work is to measure the change in exam performance of a language model using an exploration document that was generated by a model from a different family. Such work would allow for further insight into the communicability of math developed during a meeting. Another possible direction is to sweep the size of exploration documents generated in order to see how much mathematical information can be compressed before it is lost, as would be indicated by baseline-level performance on the tasks. Finally, further qualitative analysis on the transcripts, especially GLM-5.3 for which all of the reasoning tokens are unencrypted, may prove insightful in terms of generating new hypotheses about how exploration and theory generation works in language models.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods, helping develop theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, supporting qualitative and thematic data analysis, and interpreting results.

We have not used generative AI tools for generating synthetic data sets, and formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, assisting with translation, and cleaning and reformatting dataset are not applicable to this work.

Additionally, we used generative AI tools for modifying scientific figures and suggesting experimental parameters. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Mohammed Abouzaid, Andrew J Blumberg, Martin Hairer, Joe Kileel, Tamara G Kolda, Paul D Nelson, Daniel Spielman, Nikhil Srivastava, Rachel Ward, Shmuel Weinberger, et al. First proof. *arXiv preprint arXiv:2602.05192*, 2026.

Noga Alon, Thomas F Bloom, W Timothy Gowers, Daniel Litt, Will Sawin, Arul Shankar, Jacob Tsimerman, Victor Wang, and Melanie Matchett Wood. Remarks on the disproof of the unit distance conjecture. *arXiv preprint arXiv:2605.20695*, 2026.

Levent Alpöge and Tristan Buckmaster. Finite-time blowup for the 3d incompressible euler equations with smooth forcing, 2026. URL `https://cims.nyu.edu/~tristanb/euler.pdf`. Preprint.

Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5: A versatile and industrial-strength smt solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, pp. 415–442. Springer, 2022. doi: 10.1007/978-3-030-99524-9_24.

Matthew Bolan, Joachim Breitner, Jose Brox, Nicholas Carlini, Mario Carneiro, Floris van Doorn, Martin Dvorak, Andrés Goens, Aaron Hill, Harald Husum, et al. The equational theories project: Advancing collaborative mathematical research at scale. *arXiv preprint arXiv:2512.07087*, 2025.

Simon Colton. *Automated theory formation in pure mathematics*. Springer Science & Business Media, 2012.

Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv preprint arXiv:2511.02864*, 2025.

William Timothy Gowers et al. The two cultures of mathematics. *Mathematics: frontiers and perspectives*, 65:65, 2000.

Moa Johansson and Nicholas Smallbone. Conjectures, tests and proofs: An overview of theory exploration. *arXiv preprint arXiv:2109.03721*, 2021.

Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In *Computer Aided Verification*, pp. 1–35. Springer, 2013. doi: 10.1007/978-3-642-39799-8_1.

Douglas Bruce Lenat. *AM: an artificial intelligence approach to discovery in mathematics as heuristic search.* Stanford University, 1976.

John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence. Technical report, Dartmouth College, August 1955. Proposal for the 1956 Dartmouth Summer Research Project on Artificial Intelligence.

Shubhra Mishra, Yuka Machino, Gabriel Poesia, Albert Jiang, Joy Hsu, Adrian Weller, Challenger Mishra, David Broman, Joshua B Tenenbaum, Mateja Jamnik, et al. A matter of interest: Understanding interestingness of math problems in humans and language models. *arXiv preprint arXiv:2511.08548*, 2025.

OpenAI. Finite time blowup for navier–stokes, September 2026. URL https://cdn.openai.com/pdf/32d9f210-8b73-45e0-91bc-82a30aef8a9a/navier-stokes.pdf.

Gabriel Poesia, David Broman, Nick Haber, and Noah D Goodman. Learning formal mathematics from intrinsic motivation. *Advances in Neural Information Processing Systems*, 37:43032–43057, 2024.

SAIR Foundation. Equational theories benchmark. https://huggingface.co/datasets/SAIRfoundation/equational-theories-benchmark, 2026. Hugging Face dataset.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *Advances in Neural Information Processing Systems*, 37: 11545–11569, 2024.

George Tsoukalas, Rahul Saha, Amitayush Thakur, Sabrina Reguyal, and Swarat Chaudhuri. Learning interestingness in automated mathematical theory formation. *Advances in Neural Information Processing Systems*, 38:38896–38928, 2025.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. doi: 10.1093/mind/LIX.236.433.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: A cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2109.00110.

A Appendix

A.1 Equational theory supplement

Groups are a notable example of an algebraic structure: they are highly useful for representing symmetries and geometric transformations, and as such have been applied to a broad range of mathematical and scientific problems, from demonstrating the impossibility of a general solution to a quintic polynomial to classifying molecular and crystalline structures.

As a motivating example, the class of groups can be described as an equational theory by specifying the domain, operations, and axioms as follows. Over the set G and the operations $\{\cdot, ^{-1}, e\}$, we specify the axioms:

1. Associativity. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
2. Identity. $e \cdot x = x$
3. Inverse. $x^{-1} \cdot x = e$

Under this definition, an algebraic structure is called a group when each axiom holds under every assignment of its variables to an element in G.

A.2 Equational theory generation

Breadth of equational theory space. Although we scope out a highly-specific region of the space of equational theories for our project, the space of equational theories is enormous. Relaxing any of our restrictions to signature, variable count, number of axioms, and term size dramatically increases the number of theories. For example, by expanding the term specifications to allow for max seven nodes and depth of three, the number of identities grows to 4,868, and the number of two-axiom theories grows to $\binom{4868}{2} = 11{,}846{,}278$.

Given the breadth of the number of theories, it is difficult to explore the space of all of these theories. This is reflected in the literature, as the past decades have documented several notable cases of discoveries of surprising structure in equational axiomatizations that were not obvious before – single axiomatizations, equational theories project. The range of this space of theories, when considered together with the ease of defining a grammar for generating arbitrary theorems, makes equational theories highly interesting for developing challenges for language models.

A.3 THEORY FILTRATION STEP

Table 3: **Theory filtering steps.** We specify several theory filters in order to deduplicate theoretical structures and select theories for which we can derive non-trivial consequences.

Stage	Condition	Tools	Theory count
Initial theories	Form pairs of distinct identities	Python enumeration	1,953
Operation mirroring	Keep one representative after reversing the arguments at every operation node in both axioms, then checking for equivalence up to variable renaming and equality reversal	Python parsing	1,017
Axiom independence	Require countermodels to entailment in both directions between the two axioms	Python models, rewrites; cvc5; Vampire	954
Nonassociativity	Require a model satisfying both axioms but violating associativity	Python models, rewrites; cvc5; Vampire	855
Consequence count	At least 50 nontrivial consequences within the stated rewrite bounds	Python rewrite search	486
Consequence distance	A nontrivial consequence has shortest bounded rewrite distance at least 6	Python rewrite search	373

A.4 PROBLEM TYPES AND CONSTRUCTION METHODS

The procedure for rewrite-based consequence generation/search is described in Listing 2:

```
Input: axioms, starting expressions, maximum rewrite depth

For each starting expression s:
    current expressions = {s}
    # Example: s = x * (y * z)
    # Example axiom: x * y = y * x

    Repeat up to the maximum rewrite depth:
        Find new expressions t obtainable from the current
        expressions by applying one axiom, in either direction,
        anywhere inside.
        # Apply commutativity to s.
        # First round includes (y * z) * x:
        #
        # Apply commutativity to the inner expression y * z.
        # Second round includes (z * y) * x:

        Discard expressions that are too large or already visited.

        Record s = t for each new expression t, saving its
        rewrite path.
        # We have x * (y * z) = (z * y) * x.

        Use the new expressions as the next round's current
        expressions.
```

Listing 2: Pseudocode for deriving consequences from the axioms through breadth-first rewriting. Search stops when a budget of visited (starting expression, term) pairs is exhausted.

The current question writer uses five subjects. Candidate generation and answer certification are separate: a proposed identity is retained only when a proof or a checked countermodel establishes its answer. The tables below describe all generation routes in the current writer.

Basic properties. A set of basic properties depicted in Table 4 are proposed as problems and sent to the equational solver to check for a proof or countermodel.

Table 4: Basic-property construction templates.

Template	Proposed identity
Idempotence	$x \star x = x$
Commutativity	$x \star y = y \star x$
Associativity	$(x \star y) \star z = x \star (y \star z)$
Left distributivity	$x \star (y \star z) = (x \star y) \star (x \star z)$
Right alternativity	$(x \star y) \star y = x \star (y \star y)$
Left absorption	$x \star (x \star y) = x \star y$
Right absorption	$(x \star y) \star y = x \star y$
Mediality	$(x \star y) \star (z \star w) = (x \star z) \star (y \star w)$

Equational consequences. We implement five methods for deriving an equational consequence. Across all generation methods, candidates are filtered for nontriviality and deduplicated before exam selection.

Table 5: Equational consequence construction methods in order of execution.

Method	Candidate construction	Possible answers
Rewrite search	Apply breadth-first rewrite search.	Follows
Rewrite identity mutations	Replace one variable occurrence in a derived rewrite identity with another variable; check the new identity with combined solver.	Either
Lemma-assisted rewrite	Derive a lemma using rewrite search. Then add it as a rewrite rule and rerun rewrite search to derive identities that share a lemma.	Follows
Lemma-assisted mutations	Mutate lemma-assisted conclusions and search for countermodels.	Does not follow
Short-identity sampling	Propose short equations over x, y, with at most five nodes per side. Check with combined solver.	Either

Additional assumptions. We use two sources of extra assumptions: basic properties enumerated in Table 4 or short identities enumerated over x, y, z of at most five nodes per side, with the same variable set on both sides. We then use saved finite models to prove that the assumption is not entailed from the axioms only, and furthermore that the resulting theory with the added assumption is non-trivial: it admits a non-singleton model. Once the assumption checks are passed, we apply the typical rewrite search for identities.

Table 6: Additional-assumption construction methods.

Method	Candidate construction	Retained answers
Basic assumptions	Assume a basic property; search for consequences using bounded rewrites.	Follows
Generated-law assumptions	Assume an enumerated identity; search for consequences using bounded rewrites	Follows
Consequence mutation	Change one variable occurrence in a derived conclusion. Search for a countermodel satisfying the base axioms and the assumption.	Does not follow

Axiom ablation. We take entailed identities derived from basic properties and equational consequences, omit an axiom, and check whether or not the identity is still entailed from the theory. Additionally, run rewrite search on an individual axiom. Then check if the derived identities are entailed on the other axiom.

Table 7: Axiom-ablation construction methods.

Method	Candidate construction
Recheck existing consequences	Take previously-established entailed basic-property and equational identities. Omit one axiom and certify the same conclusion again.
Rewrite search on individual axioms	Run rewrite search with each base axiom separately. Check if the derived identity holds under the other axiom.

Derived structure. We specify a set of unary and binary constructions $u(x)$ and $p(x)$ to study for behavior under the theory. The unary constructions consist of $x \star x$, $(x \star x) \star x$, $x \star (x \star x)$, and $(x \star x) \star (x \star x)$. The binary constructions consist of $(x \star x) \star (y \star y)$, $(x \star y) \star y$, $x \star (x \star y)$, and $(x \star y) \star (y \star x)$.

Based on these constructions or pair of constructions, we generate questions according to the types specified in Table 8.

Table 8: Derived-structure question types.

Question type	**Description**	**Example equation**
Operation laws	Properties of an individual derived operation: basic property binary laws for p as listed in Table 4 or idempotence, involution, and constancy for u.	$u(u(x)) = u(x)$
Preservation	Whether a derived unary map u preserves the original binary operation or a derived binary operation p.	$u(p(x, y)) = p(u(x), u(y))$
Identities on the image of u	associativity of p on image inputs, or commutation of left and right multiplications acting on image elements.	$y \star (z \star u(x)) = z \star (y \star u(x))$

A.5 EXAMPLE EXAM: PAIR 1199

The following 44 questions are reproduced in their saved exam order. All variables are universally quantified. The base theory has one binary operation $\star$ and the axioms

$$A_1 : ((x \star y) \star x) = (y \star x)$$
$$A_2 : (x \star (x \star y)) = (y \star x)$$

Determine whether each proposed identity follows from its displayed assumptions. Respond with `follows` or `does_not_follow`.

Question 1 Does the following identity hold in every model satisfying the axioms?

$((y \star (y \star x)) \star y) = (x \star (x \star (((x \star (x \star y)) \star y) \star y)))$

Question 2 For this question, do not assume A2; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((x \star (y \star x)) \star x) = (((x \star (x \star (x \star y))) \star y) \star x)$

Question 3 Does the following identity hold in every model satisfying the axioms?

$((x \star y) \star y) = (x \star (y \star y))$

Question 4 Does the following identity hold in every model satisfying the axioms?

$((x \star y) \star (x \star z)) = ((x \star (y \star y)) \star (((y \star z) \star x) \star z))$

Question 5 Define $u(x) = (x \star x)$.

Define $p(x, y) = ((x \star y) \star y)$.

Does the following identity hold in every model satisfying the axioms?

$p(p(u(x), u(y)), u(z)) = p(u(x), p(u(y), u(z)))$

Question 6 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((y \star y) \star (y \star y)) = (y \star ((y \star (y \star (y \star (y \star y)))) \star y))$

Question 7 Define $u(x) = ((x \star x) \star (x \star x))$.

Does the following identity hold in every model satisfying the axioms?

$(y \star (z \star u(x))) = (z \star (y \star u(x)))$

Question 8 Define $u(x) = (x \star (x \star x))$.

Does the following identity hold in every model satisfying the axioms?

$u(u(x)) = u(x)$

Question 9 Does the following identity hold in every model satisfying the axioms?

$((y \star (x \star z)) \star z) = (((x \star z) \star ((x \star ((z \star x) \star z)) \star y)) \star z)$

Question 10 Does the following identity hold in every model satisfying the axioms?

$((x \star y) \star z) = (x \star (y \star z))$

Question 11 Define $p(x, y) = (x \star (x \star y))$.

Does the following identity hold in every model satisfying the axioms?

$p(x, y) = p(y, x)$

Question 12 For this question, do not assume A2; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((x \star (x \star x)) \star z) = ((z \star (z \star (z \star (x \star (x \star x))))) \star z)$

Question 13 Does the following identity hold in every model satisfying the axioms?

$(x \star ((x \star y) \star z)) = (x \star (((z \star y) \star (y \star (y \star (y \star z)))) \star z))$

Question 14 For this question, do not assume A2; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$(((z \star x) \star x) \star z) = ((x \star ((z \star (z \star x)) \star x)) \star z)$

Question 15 Additional laws, imposed on the entire model (universally quantified):

$(x \star (x \star x)) = x$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$((y \star x) \star z) = ((x \star ((z \star (y \star y)) \star x)) \star z)$

Question 16 Define $u(x) = ((x \star x) \star (x \star x))$.

Define $p(x, y) = (x \star (x \star y))$.

Does the following identity hold in every model satisfying the axioms?

$u(p(x, y)) = p(u(x), u(y))$

Question 17 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((x \star (x \star x)) \star z) = ((z \star (x \star ((x \star (x \star x)) \star x))) \star z)$

Question 18 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$(x \star ((z \star y) \star z)) = (x \star ((y \star z) \star ((y \star z) \star z)))$

Question 19 Does the following identity hold in every model satisfying the axioms?

$(x \star (y \star z)) = ((x \star y) \star (x \star z))$

Question 20 Does the following identity hold in every model satisfying the axioms?

$((x \star y) \star (z \star w)) = ((x \star z) \star (y \star w))$

Question 21 Does the following identity hold in every model satisfying the axioms?

$((y \star (y \star x)) \star x) = (((x \star (x \star (x \star y))) \star y) \star x)$

Question 22 Does the following identity hold in every model satisfying the axioms?

$((y \star (x \star x)) \star x) = (x \star (x \star (y \star (x \star (x \star (x \star y))))))$

Question 23 Does the following identity hold in every model satisfying the axioms?

$(x \star (x \star y)) = (x \star y)$

Question 24 Additional laws, imposed on the entire model (universally quantified):

$(x \star (x \star x)) = x$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$((y \star x) \star z) = ((x \star ((y \star (y \star y)) \star x)) \star z)$

Question 25 Does the following identity hold in every model satisfying the axioms?

$(z \star ((z \star y) \star x)) = (z \star ((x \star (z \star (z \star (z \star y)))) \star x))$

Question 26 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((x \star (x \star z)) \star z) = ((x \star (((x \star (z \star x)) \star x) \star z)) \star z)$

Question 27 Does the following identity hold in every model satisfying the axioms?

$((x \star y) \star y) = (x \star y)$

Question 28 Additional laws, imposed on the entire model (universally quantified):

$((x \star x) \star x) = x$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$(z \star (x \star y)) = ((x \star z) \star ((y \star x) \star y))$

Question 29 Does the following identity hold in every model satisfying the axioms?

$((y \star (y \star x)) \star x) = (x \star (x \star y))$

Question 30 For this question, do not assume A2; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$(((x \star z) \star x) \star x) = ((((x \star (z \star x)) \star x) \star (z \star x)) \star x)$

Question 31 Define $u(x) = (x \star (x \star x))$.

Does the following identity hold in every model satisfying the axioms?

$((u(x) \star z) \star y) = ((u(x) \star y) \star z)$

Question 32 Additional laws, imposed on the entire model (universally quantified):

$((x \star x) \star x) = x$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$(z \star (x \star y)) = ((z \star z) \star ((y \star x) \star y))$

Question 33 Additional laws, imposed on the entire model (universally quantified):

$((x \star y) \star z) = (x \star (y \star z))$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$(z \star (x \star y)) = (z \star ((x \star (x \star (y \star y))) \star x))$

Question 34 Define $p(x, y) = ((x \star y) \star y)$.

Does the following identity hold in every model satisfying the axioms?

$p(x, p(x, y)) = p(x, y)$

Question 35 Define $u(x) = (x \star x)$.

Does the following identity hold in every model satisfying the axioms?

$u((x \star y)) = (u(x) \star u(y))$

Question 36 Does the following identity hold in every model satisfying the axioms?

$(x \star y) = (y \star x)$

Question 37 Additional laws, imposed on the entire model (universally quantified):

$((x \star x) \star y) = (y \star x)$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$((y \star x) \star z) = ((z \star (((z \star x) \star y) \star y)) \star z)$

Question 38 Additional laws, imposed on the entire model (universally quantified):

$((x \star x) \star y) = (y \star x)$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$(z \star (x \star y)) = (z \star (y \star ((z \star (x \star x)) \star y)))$

Question 39 Define $p(x, y) = ((x \star x) \star (y \star y))$.

Does the following identity hold in every model satisfying the axioms?

$p(p(x, y), y) = p(x, p(y, y))$

Question 40 Define $u(x) = (x \star (x \star x))$.

Define $p(x, y) = ((x \star x) \star (y \star y))$.

Does the following identity hold in every model satisfying the axioms?

$u(p(x, y)) = p(u(x), u(y))$

Question 41 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$((x \star (x \star z)) \star z) = ((x \star z) \star (((z \star (z \star x)) \star x) \star z))$

Question 42 Additional laws, imposed on the entire model (universally quantified):

$(x \star x) = x$

Does the following identity hold in every model satisfying the axioms and the displayed additional laws?

$(y \star (x \star x)) = ((x \star (x \star (y \star (y \star x)))) \star x)$

Question 43 For this question, do not assume A1; use only the remaining displayed base axioms.

Does the following identity hold in every model satisfying the remaining base axioms?

$(((z \star z) \star z) \star z) = (((z \star (((z \star z) \star z)) \star z) \star z)$

Question 44 Does the following identity hold in every model satisfying the axioms?

$((x \star x) \star y) = ((y \star y) \star x)$

A.6 Exploration Implementation

```
Explore this theory: choose your own investigations, identifying
for yourself promising, interesting and/or useful directions.
Develop them and describe the scope and limitations of your
conclusions. Distinguish established results from conjectures,
and state any additional assumptions explicitly.

Write a self-contained account of your main insights and their
supporting evidence.

[Theory axioms appended here.]
```

Listing 3: Natural language exploration prompt.

A.7 GPT EXPLORATION DOCUMENT FOR PAIR 1199

This is the full exploration document produced by GPT-5.6-Sol in the first repetition (stored index 0) of the corresponding evaluation run. The text is reproduced with formatting adapted for LaTeX, before compression into the short guide used by the receiver. The mathematical claims below are those of the model-generated document.

1. The axioms force commutativity Write xy for $x * y$, retaining parentheses when needed. The axioms are

$$\text{(A1)} \quad (xy)x = yx, \qquad \text{(A2)} \quad x(xy) = yx.$$

Let

$$p = xy, \qquad q = yx.$$

Substitute xy for x, and x for y, in A1:

$$((xy)x)(xy) = x(xy).$$

Using the original axioms on both sides gives

$$qp = q.$$

The same substitution in A2 gives

$$(xy)((xy)x) = x(xy),$$

hence

$$pq = q.$$

Now interchange x and y in the first calculation. This gives

$$pq = p.$$

Consequently $p = q$, so

$$\boxed{xy = yx.}$$

Thus every model is commutative, even though neither axiom explicitly states commutativity.

---

2. A simpler equivalent axiom system Once commutativity is known, both axioms reduce to

$$x(xy) = xy. \tag{B}$$

Conversely, suppose a binary operation is commutative and satisfies B. Then

$$(xy)x = x(xy) = xy = yx,$$

which is A1, while B and commutativity give A2 directly.

Therefore the original theory is exactly the theory of commutative magmas satisfying

$$\boxed{xy = yx, \qquad x(xy) = xy.}$$

In the idempotent case, structures satisfying these identities are often called **2-semilattices**, though terminology varies.

The identity $x(xy) = xy$ says that once x has been multiplied by y, multiplying the result by x again has no further effect.

---

3. Every product is idempotent Let $p = xy$. By B and commutativity,

$$yp = y(xy) = xy = p.$$

Now apply B with $x = p$ and $y = y$:

$$p(py) = py.$$

Since $py = p$, this becomes

$$p^2 = p.$$

Hence

$$\boxed{(xy)(xy) = xy.}$$

Thus every value of the binary operation is idempotent, although arbitrary elements need not themselves be idempotent.

Let

$$E = \{e \in A : e^2 = e\}.$$

Because every product is idempotent, $A^2 \subseteq E$, where

$$A^2 = \{xy : x, y \in A\}.$$

Conversely, if $e \in E$, then $e = e^2$, so $e \in A^2$. Therefore

$$\boxed{E = A^2.}$$

This gives a useful structural picture:

- every multiplication lands in E;
- E is closed under multiplication;
- the restriction of the operation to E is commutative and idempotent;
- on E, the identity $x(xy) = xy$ still holds.

So every model has an idempotent "core" E, and all products lie in that core after one multiplication.

The square map

$$q : A \to E, \qquad q(x) = x^2$$

is a retraction:

$$q(q(x)) = (x^2)^2 = x^2,$$

and $q(e) = e$ for $e \in E$. In general, however, q need not determine all products and need not be a homomorphism.

---

4. Idempotence of all elements is not forced Consider a two-element set $\{0, 1\}$ with constant multiplication

$$xy = 0$$

for every x, y. Both axioms hold, since every expression involving a multiplication evaluates to 0. But

$$1^2 = 0 \neq 1.$$

Thus the theory does **not** imply $x^2 = x$.

More generally, every constant operation $xy = c$ is a model.

This also shows that the distinction between the whole carrier A and its idempotent core $E = A^2$ is genuine.

---

5. Associativity is not forced There are very small nonassociative models. Let

$$A = \{a, b, c\},$$

make every element idempotent, and define the products of distinct elements by

$$ab = b, \qquad bc = c, \qquad ca = a,$$

with commutativity determining the reverse products.

This is a conservative operation: xy is always either x or y. Every commutative, idempotent, conservative operation automatically satisfies B:

- if $xy = x$, then $x(xy) = xx = x = xy$;
- if $xy = y$, then $x(xy) = xy$.

However,

$$(ab)c = bc = c,$$

while

$$a(bc) = ac = a.$$

Hence

$$(ab)c \neq a(bc).$$

Therefore

Associativity is independent of the given axioms.

In fact, any tournament orientation gives such an idempotent model: for each distinct pair $\{x, y\}$, choose one of x, y to be xy. This demonstrates that the class of models is extremely broad.

---

6. An order-like interpretation—and its limitation On the idempotent core E, define

$$u \leq v \quad \Longleftrightarrow \quad uv = v.$$

Because elements of E are idempotent, this relation is reflexive. It is also antisymmetric: if $uv = v$ and $uv = u$, then $u = v$.

Moreover, B gives

$$u(uv) = uv, \qquad v(uv) = uv,$$

so uv is an upper bound of both u and v in this relation.

However, $\leq$ need not be transitive. In the three-element example above,

$$a \leq b, \qquad b \leq c,$$

but $ac = a$, so $a \not\leq c$. Indeed, the relation forms a directed cycle.

Thus the operation resembles a join operation, but without associativity it need not arise from a partial order or give least upper bounds. Calling these structures "semilattices" without qualification would therefore be misleading.

---

7. Exact classification under the additional assumption of associativity Now add the explicit assumption

$$(xy)z = x(yz).$$

Because commutativity is already established, the models become commutative semigroups satisfying

$$x^2y = xy.$$

Let E again be the set of idempotents and define $q(x) = x^2$. For all x, y,

$$q(x)q(y) = x^2y^2.$$

Associativity and the absorption identity give

$$x^2y^2 = (x^2y)y = (xy)y = xy.$$

Hence

$$\boxed{xy = q(x)q(y).}$$

The restriction of the operation to E is now associative, commutative, and idempotent, so E is an ordinary semilattice. Also,

$$q(xy) = (xy)^2 = xy = q(x)q(y),$$

so $q : A \to E$ is a homomorphism and a retraction.

This yields a complete classification of the associative models.

CLASSIFICATION THEOREM An associative model consists precisely of:

1. a semilattice E;
2. a nonempty set A containing E;
3. a retraction $q : A \to E$, meaning $q(e) = e$ for $e \in E$;

with multiplication defined by

$$\boxed{xy = q(x)\, q(y),}$$

where the product on the right is the semilattice operation in E.

Conversely, every such construction satisfies A1 and A2 and is associative.

Thus associative models are "inflations" of semilattices: elements in the same fiber of q are indistinguishable after they participate in a multiplication.

Special cases include:

- If q is the identity, the model is an ordinary semilattice.
- If E has one element, the operation is constant.
- Nonidempotent elements can occur, but every product immediately collapses into the semilattice core.

---

8. Further consequences

8.1 REPEATED MULTIPLICATION BY A FACTOR STABILIZES The equivalent axiom B gives

$$x(xy) = xy, \qquad y(xy) = xy.$$

Thus a product absorbs both of its factors.

8.2 CANCELLATIVE MODELS ARE TRIVIAL Suppose left cancellation holds. Since

$$x(xy) = xy = x\, y,$$

cancelling the initial x from

$$x(xy) = xy$$

gives $xy = y$. Commutativity then also gives $xy = x$, so $x = y$ for all x, y. Therefore the carrier has one element.

Hence every cancellative model—and in particular every quasigroup model—is trivial.

8.3 ALL NON-VARIABLE TERM VALUES ARE IDEMPOTENT Any term whose outermost symbol is multiplication has the form uv. Its value is therefore a product and is idempotent. Consequently, in every model,

$$t^2 = t$$

for every non-variable term t.

This does not imply $x^2 = x$ for a lone variable x.

---

9. Scope and limitations The main established conclusions are:

A1 and A2 are equivalent to commutativity plus $x(xy) = xy$.

They imply:

- every product is idempotent;
- the set of products is exactly the set of idempotent elements;
- every model has an idempotent core containing all multiplication values;
- neither global idempotence nor associativity is forced;
- under associativity, the models are exactly retractive inflations of semilattices.

Without associativity, a comparably rigid classification should not be expected: the theory already includes every commutative idempotent conservative magma, and hence arbitrary tournament-type operations. The order-like relation induced by multiplication may fail even transitivity.

No conjectural claims are needed for the results above. The only additional assumption used in the classification theorem is associativity, and it is essential.

A.8 EXAM PERFORMANCE BY QUESTION TYPE

Table 9: Accuracy by question type. Base and Guided denote baseline and exploration-guided conditions. Values are averaged over three runs within each theory and then equally over 21 theories.

	GPT-5.6 Sol (%)		Claude Sonnet 5 (%)		GLM-5.3 (%)	
Question type	Base	Guided	Base	Guided	Base	Guided
2k token threshold						
Basic properties	49	94	22	82	7	56
Equational consequences	29	64	12	50	15	47
Additional assumptions	55	80	31	40	32	51
Axiom ablation	83	87	29	32	42	41
Derived structure	30	63	8	47	12	47
7.5k token threshold						
Basic properties	97	98	23	82	24	65
Equational consequences	84	86	13	51	32	63
Additional assumptions	96	98	33	41	49	64
Axiom ablation	99	99	32	34	70	71
Derived structure	83	85	9	49	28	59