SCIVERGE: OBJECTIVE EVALUATION OF SCIENTIFIC REASONING IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are beginning to contribute to scientific work, but their scientific reasoning is rated by human experts or language models rather than measured objectively. Measuring scientific reasoning often requires experimentation to determine whether the ideas a model considered have potential. Mathematics provides an alternative: when a scientific system is described mathematically, as a set of states and the actions that change them, whether an idea would work can be derived from the underlying mathematics. We present SciVerge, a benchmark that describes scientific systems mathematically to construct questions that objectively measure a language model's divergent thinking, the production of many possible approaches to a question, and convergent thinking, the narrowing of those approaches to the best one, the two components of creative thought identified in the psychology of creativity, which proposing and selecting a scientific approach require. Each question is posed in three phases: the model proposes approaches, commits to a subset of them, and attempts the committed approaches until one produces a proof or all have failed. SciVerge mathematically determines whether the approaches that are proposed and selected by a model can succeed, and decomposes divergent and convergent thinking into six metrics to evaluate the model's performance. SciVerge contains 500 evaluation instances drawn from over 100,000 generated across 21 systems in quantum computing, molecular biology, robotics, chemistry, crystallography, and number theory. We additionally construct a generation and screening pipeline that verifies the quality of each instance by proof before admission into the benchmark. Across seven models, the strongest solves 55 percent of instances. Scoring each phase reveals three failure modes, one per phase, that the solve rate alone would not, each calling for a different adjustment. Enabling reasoning improves selection and execution far more than proposal, and every model is far better at disproving a claim than at proving one: the strongest disproves 86 percent of the false claims but proves only 24 percent of the true ones.

1 INTRODUCTION

AI (Artificial Intelligence) systems have begun to produce results in scientific work: new materials (Merchant et al., 2023), protein structures (Jumper et al., 2021), mathematical constructions (Romera-Paredes et al., 2024), and in September 2026 a proof, by roughly 10,000 concurrent agents, that solutions of the three-dimensional Navier-Stokes equations can develop a singularity in finite time (OpenAI, 2026). However, the scientific reasoning behind such results is measured by benchmarks whose scores are ratings rather than measurements: a domain expert or a language model judges how promising a model's idea seems, because whether the idea would have worked is difficult to test without extensive experimentation (Alkan et al., 2025; Lu et al., 2024; Gottweis et al., 2026; Wang et al., 2026; Guo et al., 2025). Producing more results of this kind requires evaluations that objectively verify whether a model's idea would have worked, not just rate how promising it seems.

Mathematics provides an alternative to experimentation. A mathematical description of a scientific system specifies its possible states and the basic actions that change them; every property of the system can then be expressed mathematically in terms of those states and actions, and any claim about it can be proven or disproven. A system specified in this way is a *group*, the structure studied

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.

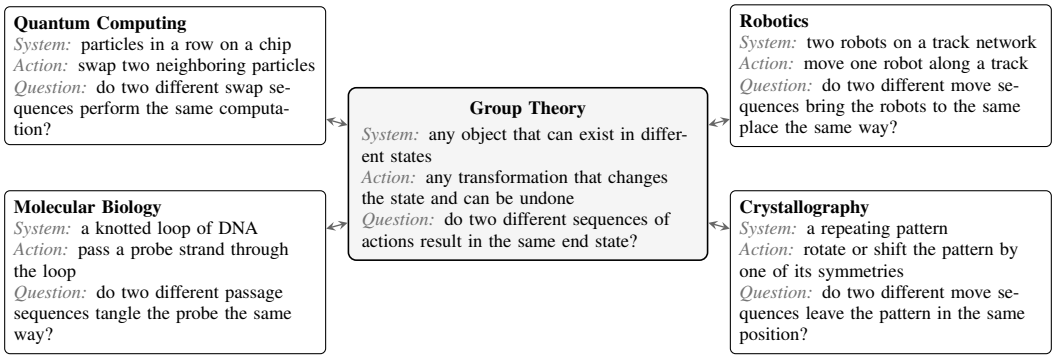


Figure 1: Four scientific questions and the group theory question underlying each. In each field the system is a set of states, an action is a reversible transformation, and the question is whether two action sequences have the same effect. That is a question about a group, so it can be answered by proof, and so can each field's version of it.

by group theory (Coxeter & Moser, 2013). Figure 1 shows that this specification is general enough to apply across several scientific fields, some of which already describe their systems in these terms, so a question a field asks about its system can be posed as a question about a group. Whether a proposed approach to that question can succeed is then a claim about the group, decidable mathematically rather than experimentally. A model's idea for solving the question can therefore be scored on whether it would succeed, rather than rated on how promising it seems.

We present SciVerge, a benchmark that describes scientific systems mathematically to construct questions that measure a language model's divergent thinking, the production of many possible approaches to a question, and convergent thinking, the narrowing of those approaches to the best one, the two components of creative thought identified in the psychology of creativity, which proposing and selecting a scientific approach require (Guilford, 1967). Each question is posed in three phases. In the diverge phase, the model proposes up to five approaches to the question; in the converge phase, it commits to at most three of them; in the execute phase, it attempts the committed approaches in turn until one produces a proof or all have failed. Following the psychology of creativity, we decompose divergent and convergent thinking into six metrics to measure the ability to propose, select, and execute scientific approaches (Torrance, 2018; Mumford et al., 2002). SciVerge mathematically determines whether each approach the model proposes and selects can succeed, allowing us to score divergent thinking on the number, variety, and viability of the approaches proposed (scored as fluency, flexibility, and potential), and convergent thinking on whether the model commits to only the proposals that can succeed and ignores the rest (discrimination and restraint). A sixth metric, establishment, scores whether the execution produced a proof. SciVerge contains over 100,000 instances spanning 21 systems in six fields, and an evaluation set consisting of 500 instances, balanced across question types and answers. We implement a verification procedure that analyzes proofs programmatically to screen generated instances before admission, determine whether each proposed approach can succeed, and evaluate proofs provided by the models.

We evaluate seven open-weight models with reasoning enabled and disabled. The strongest solves 55 percent of instances, leaving room for the benchmark to measure improvement. Our analysis of model performance within each phase reveals three failure modes, one per phase: proposing too few approaches to select among, selecting at chance, and selecting well but failing to execute. Two models with the same solve rate can fail in different modes and call for opposite adjustments, which the solve rate alone would not show. The decomposition also shows which phases an intervention changes: enabling reasoning, for example, leaves the number and variety of proposals almost unchanged, but moves selection from chance to 4 to 15 points above it, and raises the solve rate by up to 47 points. Splitting the results by answer shows where the remaining failures lie: every model is far better at disproving a claim than at proving one, and the strongest proves under a quarter of the true claims it is given. In summary, our contributions are:

- **Objective evaluation of scientific ideas.** By describing scientific systems mathematically, SciVerge determines by proof whether each approach a model proposes can succeed, so

that a model's proposed approaches are evaluated on their ability to succeed rather than rated by an expert or a language model.

- **A benchmark and generation pipeline.** Over 100,000 questions spanning 21 systems in six scientific fields, each verified by proof before admission, with a pipeline that can continuously generate and screen new instances.
- **A three-phase task with six metrics.** A protocol that observes a model's proposals, commitments, and execution separately, with metrics adapted from the psychology of creativity for divergent and convergent thinking.
- **An evaluation of current models.** Seven models with reasoning enabled and disabled, identifying three failure modes, one per phase, and showing that reasoning improves selection and execution far more than proposal.

2 SciVerge

SciVerge poses a question about a scientific system to a model and scores the model on the approaches it proposes, the ones it commits to, and whether it solves the question. Table 1 shows one such question, which we call an *instance*, as presented to the model. Making those scores objective requires three things, defined in turn below: a mathematical description of a system, so that a question about it has a provable answer (Section 2.1); a verification procedure that decides, by program, whether a model's answer is correct and whether an approach it proposes can succeed (Section 2.2); and a task that elicits proposals, commitments, and attempts separately so that each can be scored (Section 2.3). Section 2.4 describes how the instances were generated and screened with the same verification procedure.

2.1 Mathematical Description of a Scientific System

A question can be answered by proof only when the system it is about is specified completely. We specify a system by a finite set X of basic *actions*, each of which changes the state of the system and each of which has a *reverse* that undoes it, together with a finite set R of *relations*, sequences of actions that produce no net change. The pair $P = \langle X \mid R \rangle$ is the system's *presentation*. For example, the DNA system in Table 1 has two actions, one for each way the probe loop can pass, and one relation, $aaBBB$, stating that two passes through the hole have the same effect as three passes around the core. Everything a question can ask is phrased in terms of *words*, finite sequences of actions and reverses performed in order, written with a lowercase letter for each action and the uppercase letter for its reverse, with ε for the empty sequence. Two words are *equal* in the system, written $u =_P v$, when one can be transformed into the other by inserting or deleting a relation or an action followed by its reverse; equal words have the same net effect on the system. Equality is the only fact that the relations fix directly, and every question below reduces to it.

Each question is an *instance* $I = (P, \tau, \theta)$: a presentation P, one of five question types τ, and the words θ the question is about. Each type defines a property $\varphi_\tau(\theta)$ of those words, and the instance asks whether it holds in P. The five types ask whether two sequences of actions have the same net effect (equivalence), can be performed in either order (compatibility), or are the same sequence performed from different starting configurations (conjugacy); whether a sequence can be assembled from a given list of sequences (membership); and whether repeating a sequence eventually returns the system to its starting state (periodicity). Each type was chosen because a field in Figure 1 asks it of its systems and because it reduces to word equality; the formal definitions are given in the Appendix. Every instance has a definite *truth value* $y^*_I \in \{\mathrm{T}, \mathrm{F}\}$, the value of $\varphi_\tau(\theta)$ in P.

2.2 Verification

Every score in SciVerge is computed from the output of one procedure, Check, which decides whether a proof is correct. This subsection describes the structure of a certificate that serves as a proof for each type of answer, states why an accepted proof settles the question, and then extends the same procedure from answers to approaches.

Answers. A *response* to an instance is a pair (y, c) of a verdict $y \in \{\mathrm{T}, \mathrm{F}\}$ and a certificate c, an object that proves the verdict and can be checked mechanically. The two verdicts call for different

Table 1: An instance as presented to the model. The system is described by its actions and one relation among them, and the question asks about a sequence of actions; in the terms of Section 2.1, the actions are $X = \{a, b\}$, the relation is $R = \{aaBBB\}$, and the question is a membership instance.

A SciVerge instance: molecular biology	
System	A circular DNA molecule lies on a torus as the $(2, 3)$ torus knot, winding twice through the hole and three times around the core. A probe loop is passed around the molecule.
Actions	`a`: pass the probe once through the hole. `b`: pass the probe once around the core. An uppercase letter is the same passage in reverse, and a sequence of letters is performed left to right.
Relation	`aaBBB`: two passes through the hole have the same effect as three passes around the core.
Question	Can the passage sequence `w` = `bbbAABBAAbb` be built from the sequences `bbb`, `a`, `A` and their reverses alone?

certificates. Every property reduces to the equality of two words, and equality was defined by transformation: two words are equal when one can be reached from the other by inserting or deleting relations and canceling pairs. A certificate that the property holds is therefore a *derivation*, the sequence of words leading from one side of the property to the other, each obtained from the previous by one such step, together with any word the property asks for (the conjugator for conjugacy, the assembly for membership). A certificate that the property fails cannot take this form, since no finite sequence shows that no derivation exists. It is instead a *world*: a concrete realization of the system in which each action is a permutation of a finite set or a matrix, chosen so that every relation in R has no net effect, and in which the two sides of the property have different effects. A world obeys the same rules as the system, so equal words have the same effect in it, and two words with different effects cannot be equal. For periodicity, the world must instead show that w has infinite order. Check evaluates a derivation by comparing consecutive words and a world by computing the actions over the finite set or by exact rational arithmetic; we write $\mathrm{Check}(I, y, c) \in \{0, 1\}$ for the outcome. The two kinds of certificate make different demands. A derivation must be written out step by step, and its length grows with the distance between the two words, whereas a world is checked by evaluating the relations and the property once, so a short world can refute a claim whose derivation, if one existed, would be long. Proving and refuting are therefore distinct skills, and the evaluation set contains both in equal number so that each is measured.

Proposition 2.1 (Soundness)**.** *For every instance I and response (y, c), if $\mathrm{Check}(I, y, c) = 1$ then $y = y^*_I$.*

The proof of Proposition 2.1 is given in Appendix A.1. A verdict whose certificate is not accepted is scored as incorrect, and by soundness (Proposition 2.1) a verdict whose certificate is accepted is correct; every score in this paper is therefore computed from accepted certificates alone, with no human or language model in the loop.

Approaches. Scoring a model's proposals requires knowing, for each approach it could take, whether that approach can succeed on the instance, and Check decides this as well. An *approach* to an instance is a kind of certificate together with the parameters needed to produce it: a derivation using a named subset of the relations, a derivation through a named intermediate word or helper identity, a world of a named size, a matrix world of a named dimension, a world given by a named finite symmetry group, or a count of actions. Each kind ends in a certificate that Check accepts, and every accepted certificate is of one of these kinds, so the set of approaches is complete with respect to Check. An approach is *viable* on an instance when an accepted certificate of its kind exists for that instance; by soundness, a viable approach settles the question. Approaches that end in a world can be viable only on false instances and those that end in a derivation only on true ones. Viability is decided by a bounded search for each kind; the 54 of 2,600 approach-instance pairs in the evaluation set that the search does not settle, and approaches whose parameters the model supplies and the search cannot bound, are left unlabeled and excluded from any score that depends on them (Appendix A.5). For an instance and a response, we write A for the set of approaches proposed, $C \subseteq A$ for those committed to, and $V \subseteq A$ for those that are viable.

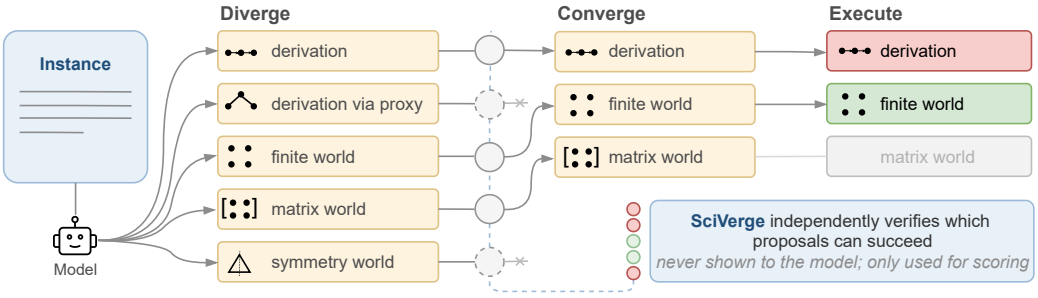


Figure 2: The SciVerge task consists of three phases: diverge, converge, then execute. Yellow rectangles are approaches the model proposed; solid circles are approaches it committed to, and dashed circles were left uncommitted. Green and red circles mark which proposals SciVerge verifies can succeed on the instance. In the execute phase, red marks a rejected certificate, green marks an accepted one, and gray represents an approach not attempted because an earlier one succeeded.

2.3 Task Protocol

The psychology of creativity separates divergent thinking, the generation of candidate ideas, from convergent thinking, the selection of the best among several candidates (Guilford, 1967). Based on this decomposition, each instance is posed to the model in three phases: the model proposes approaches to the question, commits to a subset of them, and executes the committed approaches. Each instance is presented using the same description of the system that the field uses, as in Table 1, together with the kinds of approaches available for its question type. Model performance is measured using six metrics, reported as mean percentages.

Diverge phase. The model proposes up to five approaches to the question, along with the parameters it would use. The proposals form the set A, and the subset $V \subseteq A$ that can succeed is decided as in Section 2.2, without the model's involvement. Three metrics score the model's proposals, motivated by psychology literature (Torrance, 2018). *Fluency* is $|A|/5$, the number of proposals over the maximum allowed. *Flexibility* is the number of distinct kinds among A over the number of kinds available for the question type, which ranges from five to nine. *Potential* is $|V|/|A|$, the fraction of proposals that can succeed, which can be measured only because of the mathematical basis of the questions: whether an approach can succeed is a mathematical fact about the instance rather than a judgment.

Converge phase. From its own proposals, the model commits to at most three, forming $C \subseteq A$. Scientific approaches are often inexpensive to propose but costly to execute. Deciding which to attempt is a skill in itself, which we test by imposing this limit. We provide two metrics to measure the ability to identify proposals that can succeed, following Mumford et al. (2002) on idea evaluation. *Discrimination* is $|C \cap V|/|C|$, the fraction of commitments that can succeed. *Restraint* is $|(A \setminus V) \setminus C|/|A \setminus V|$, the fraction of non-viable proposals left uncommitted. Selecting among the proposals at random gives an expected discrimination equal to potential, so the excess of discrimination over potential is the model's selection above chance.

Execute phase. For each committed approach, the model is asked to execute that approach and return a response (y, c) as defined in Section 2.2. The phase ends when a certificate is accepted, or every committed approach has been attempted. *Establishment* is scored one when a committed approach yields an accepted certificate, and zero otherwise; it is the benchmark's measure of whether the question was solved.

Figure 2 shows the protocol on one instance. The model proposes five approaches, of which verification finds that only the finite world and the matrix world can succeed. The model commits to the derivation, the finite world, and the matrix world; the derivation fails, the finite world yields an accepted certificate, and the matrix world is not attempted.

Table 2: The nine systems, the field that studies each, the number of families it contributes, the number of actions in each family ($|X|$), and the number of instances in the corpus. The full list of families is in the Appendix.

Field	System	Families	$\lvert X\rvert$	Instances
Quantum computing	Anyons exchanged in adjacent pairs	3	2–4	15,180
Quantum computing	Pauli operators on a block of qubits	1	7	7,786
Molecular biology	A knotted molecule and a probe loop	4	2	15,098
Robotics	Two vehicles moving along a network	3	5–7	11,326
Concurrency	Stations whose neighbors share a resource	2	3–4	11,352
Crystallography	A periodic plane pattern	4	3	24,056
Chemistry	Tiles rearranged by adjacent swaps	2	3–4	11,214
Number theory	A basis of a two-dimensional lattice	1	2	3,136
Number theory	Compositions of two shear maps	1	2	864

2.4 Instance Generation

The verification procedure of Section 2.2 also decides which instances enter the benchmark: an instance is admitted only when its answer has been proven and the proof accepted by Check. This subsection describes where the systems come from and how instances are generated and screened.

Every instance in SciVerge originates from a system that a scientific field uses to describe a real-world occurrence. A *family* is one presentation together with the field's description of what its actions do, so that a word is a real-world procedure and a question about words is a question about what that procedure does. A family is included only when the presentation is one the field's literature uses for that system and our transcription of it is verified to behave consistently with the group that literature describes (Appendix A.3). SciVerge contains 21 families; the DNA system of Table 1 and the anyon exchanges of a topological quantum device are two of them (Ernst & Sumners, 1990; Freedman et al., 2003). Some of the questions are the field's own: whether an operator string lies in a code's stabilizer group is the standard check that it acts as the logical identity (Gottesman, 1997; Nielsen & Chuang, 2001), recovering a key in braid-based cryptography is a conjugacy problem (Ko et al., 2000; Anshel et al., 1999), and whether two motion plans deform into one another without collision is the motion-planning question (Abrams & Ghrist, 2002). Table 2 lists systems with their fields and presentation sizes; the full list is in Appendix A.3.

The generator builds instances in reverse, constructing the certificate first and using it to make the question. For example, when creating an "Equivalence" or "Compatibility" instance where the answer is "true," the generator starts from a word and repeatedly applies relations to it, each application randomly inserting a relation, then reducing the resulting word. This produces a word equivalent to the one it started from by construction, and the list of applications is the derivation that certifies it. By altering the result in one place, this procedure produces a "false" instance at the same time, but the pair is admitted only once a counterexample is found for it. The other question types are built the same way, with the generator applying a relabeling for "Conjugacy," an assembly from the listed words for "Membership," a number of repetitions for "Periodicity," and then rewriting as above; the certificate is that object together with the derivation. Every certificate is verified before admission, and an instance whose certificate fails is discarded. Instances can therefore be generated on demand, so that an evaluation can always use instances minted after the evaluated model's training data was collected.

3 Experiments

3.1 Setup

We evaluate seven open-weight models: Qwen3.5-9B, Qwen3.6-27B, Qwen3.8-27B, Qwen3.8-Flash-Next, DeepSeek-V4-Flash, Gemma-4-31B, and GLM-4.7-Flash on an evaluation set containing 500 instances, 100 of each question type, half true and half false. Each model runs the three-phase task on the full evaluation set, one attempt per instance, twice: once with its reasoning mode enabled and once with it disabled, so that the effect of reasoning on each metric can be read

Table 3: The six metrics on the full evaluation set, in percent, with the model's reasoning mode enabled (✓) and disabled (×); best value in each column in bold, second best underlined. Establishment is followed by its value on true and on false instances separately, as (true/false). Models are grouped by family and ordered by size within a family.

		Divergent			Convergent		Execution
Model	Reas.	Fluency	Flexibility	Potential	Discrimination	Restraint	Establish. (T/F)
Qwen3.5-9B	✓	6.8	5.3	43.0	41.1	26.3	1.0 (0.0/2.0)
	×	58.2	41.8	35.1	33.3	20.4	3.0 (0.4/5.6)
Qwen3.6-27B	✓	53.3	38.3	45.3	53.7	45.8	30.4 (5.6/55.2)
	×	60.1	43.3	36.3	35.2	20.0	9.4 (0.0/18.8)
Qwen3.8-27B	✓	71.5	51.4	49.0	**63.9**	**53.8**	50.2 (17.2/83.2)
	×	65.2	46.8	39.5	39.4	16.1	19.6 (2.8/36.4)
Qwen3.8-Flash-Next	✓	73.0	52.8	47.6	59.9	48.5	**55.2** (24.4/86.0)
	×	73.0	52.3	37.3	35.9	25.9	8.4 (0.4/16.4)
DeepSeek-V4-Flash	✓	60.6	42.4	47.9	51.9	30.3	40.6 (11.2/70.0)
	×	**98.4**	**70.9**	38.4	36.9	24.0	5.8 (0.8/10.8)
Gemma-4-31B	✓	53.5	38.4	**48.6**	49.1	19.7	32.2 (6.8/57.6)
	×	59.4	42.6	34.8	35.3	24.7	12.2 (0.4/24.0)
GLM-4.7-Flash	✓	18.1	12.9	41.0	35.7	24.1	0.6 (0.0/1.2)
	×	88.2	63.1	37.2	35.1	30.3	2.0 (1.2/2.8)

within a model. Every metric is reported as a per-instance mean in percent. The full evaluation details are given in Appendix A.6.

3.2 Results

Failure modes. Table 3 reports the six metrics for every model with reasoning enabled and disabled. Figure 3 shows one model as an example of each failure mode, next to a balanced one. Each mode is a deficit in one phase. An *overcautious* model proposes almost nothing: GLM-4.7-Flash and Qwen3.5-9B with reasoning enabled have fluency below 20 and establish on at most 1 percent of instances, since an approach that is not proposed cannot be selected or executed. An *impulsive* model selects at chance: Gemma-4-31B with reasoning enabled has discrimination 49.1 against potential 48.6, the lowest restraint among the larger models. An *underachieving* model selects well but fails to execute: Qwen3.6-27B with reasoning enabled has discrimination 8.4 points above potential and restraint 45.8, yet establishes on only 30.4 percent. Gemma-4-31B and Qwen3.6-27B establish about equally often, so establishment alone would not separate them, and the adjustments they call for are opposite: selection for one, and execution for the other. Qwen3.8-27B and Qwen3.8-Flash-Next represent the balanced profile, with the highest discrimination and restraint and establishment (50.2 and 55.2) close to discrimination.

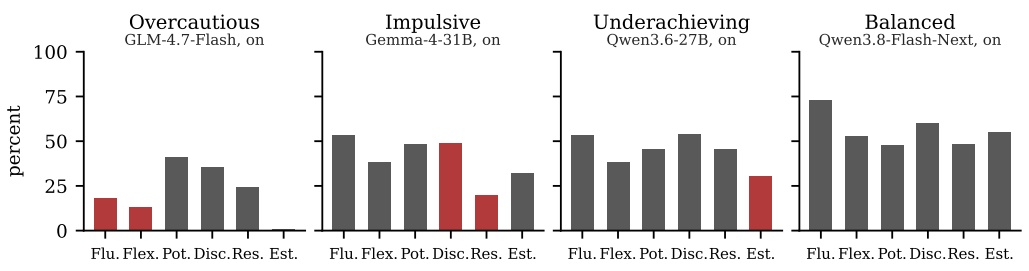


Figure 3: The six metrics for one model in each failure mode and one balanced model, all with reasoning enabled. Red marks the metrics that define the mode; each mode is a deficit in one phase. Flu. fluency, Flex. flexibility, Pot. potential, Disc. discrimination, Res. restraint, Est. establishment.

Table 4: All three phases on one instance, with reasoning enabled. Each proposal is an approach with the parameters the model supplied; ✓ marks one that can produce an accepted certificate on this instance, and × one that cannot.

Question	A circular DNA molecule lies on a torus as the $(2, 5)$ torus knot; `a` is a probe loop passing once through the hole, `b` once around the core, and `aa` has the same effect as `bbbbb`. Is there a passage sequence `x` such that performing `x`, then `u`, then reversing `x` leaves the loop in the same state as `v`? `u` = `bbba` `v` = `bbbAAbaB`$\cdots$`bbbbaBBB` (46 letters) Answer: No.	
	Qwen3.8-Flash-Next	**GLM-4.7-Flash**
Proposed	Known symmetry; `a`→reflection, `b`→rotation of a pentagon ✓ Matrix model; 2×2 matrices mod 11, both given ✓ Conserved count; noted as "not decisive" × Helper identity; `aa` = `bbbbb` ×	Conserved count; "$-13 \neq -94$, so not conjugate" × Small state model; no parameters × Larger state model; size 8 × Matrix model; 2×2 mod a prime, matrices not given × Known symmetry; target "dihedral" ✓
Committed	Known symmetry, then matrix model	Conserved count
Execution	Returns a five-point world in which the images of `u` and `v` are not conjugate. Certificate is accepted.	Miscounts the letters of `v`, concludes the counts differ, then searches for a modulus in which the images differ; they coincide mod 2, 3 and 5, and the search enters a repeating line at mod 7 that runs to the end of the budget. Invalid certificate.

Effect of reasoning. Table 3 shows how reasoning affects each phase. With reasoning disabled, every model selects at chance: discrimination is within about 2 points of potential for all seven, and DeepSeek-V4-Flash, which proposes the most (fluency 98.4, flexibility 70.9), establishes on only 5.8 percent of instances. Enabling reasoning leaves fluency and flexibility nearly unchanged for five of the models and raises potential slightly, but moves discrimination 4 to 15 points above potential for four of them and raises establishment by 20 to 47 points. Gemma-4-31B's discrimination stays roughly equal to potential, yet its establishment still rises from 12.2 to 32.2. For Qwen3.5-9B and GLM-4.7-Flash, reasoning instead induces the overcautious mode, reducing fluency from 58.2 and 88.2 to 6.8 and 18.1.

True vs. false instances. The last column of Table 3 splits establishment by the instance's answer. As the two certificate kinds in Section 2.2 would suggest, the two sides are not equally hard. For the five models that establish at all, potential is 25 to 34 on true instances against 62 to 71 on false ones, and establishment 6 to 24 against 55 to 86. Selection is also weaker on the true side, where discrimination exceeds potential by at most 13 points and falls below it for three models, against up to 17 points on the false side. The gap follows the length of the proof. On true instances, 15 to 82 percent of execution calls by the five models reach the token budget, against 9 to 31 percent on false instances, and periodicity, whose derivation is the longest, is established on at most 2 percent of true instances by any model. Since the sides are balanced, the strongest model's 55.2 percent establishment is mostly refutation: 86.0 on false instances against 24.4 on true ones. Table 7 in the Appendix gives additional details on the split by answer.

Case study. Table 4 summarizes the three phases on a single molecular biology instance for Qwen3.8-Flash-Next and GLM-4.7-Flash with reasoning enabled. The question asks whether two sequences of strand passages on a knotted DNA molecule are the same sequence performed from a different starting configuration; they are not, and the proof is a description of a system in which the two sequences behave differently. In the diverge phase, GLM-4.7-Flash proposes five approaches, exceeding Qwen3.8-Flash-Next, which proposes four, in both fluency and flexibility. Verification finds that two of Qwen3.8-Flash-Next's proposals can succeed and one of GLM-4.7-Flash's can. Qwen3.8-Flash-Next specifies the counterexample it would build in its two leading proposals and notes that its counting approach, which cannot succeed on this instance, is inconclusive. In the converge phase, Qwen3.8-Flash-Next commits to its two viable proposals and to neither of the others. GLM-4.7-Flash commits only to the counting approach, on the basis of an incorrect calculation in its

proposal that the two sequences contain different numbers of each passage. In execution, Qwen3.8-Flash-Next produces an accepted certificate on its first commitment. GLM-4.7-Flash attempts to build a counterexample from the count, finds that the sequences coincide in each counterexample it tries, and eventually fails to construct a certificate. GLM-4.7-Flash's commitment on this instance is representative of the impulsive failure mode we observe: it proposes a workable approach, but does not commit to it, instead attempting one that cannot succeed.

4 Related Work

Benchmarks of scientific reasoning pose open-ended tasks and score responses with a domain expert or a language model (Alkan et al., 2025; Lu et al., 2024; Gottweis et al., 2026; Wang et al., 2026); the score is a judgment, and no judge can determine whether an approach the model did not take would have succeeded. Formal mathematics benchmarks score a response by whether a proof assistant accepts it (Zheng et al., 2021; Azerbayev et al., 2023; Tsoukalas et al., 2024). SciVerge applies the same standard to questions posed by scientific fields, certifies false answers as well as true ones, and extends verification from the answer to each approach the model proposes. The measures we use come from psychology: divergent-thinking tests have been administered to language models with human or model raters (Guzik et al., 2023), studies of research ideation report that ideas rated as novel lose that advantage when executed (Si et al., 2025; 2026), and benchmarks of idea generation score ideas with model judges (Guo et al., 2025). SciVerge measures the same quantities by verification and observes the execution of the selected idea in the same run.

5 Discussion

Scoring the proposals objectively allows the model's process to be analyzed, not only its outcome: which approaches it considered, which it chose, and whether the chosen one was executed correctly. A judge can rate how plausible a proposal seems but cannot determine whether it would have succeeded, and so cannot score whether a model chose well among its own ideas. When a single phase accounts for most of a model's unestablished instances, the model exhibits one of the three failure modes described in Section 3.2, and the necessary adjustment is specific to that phase: additional proposals would not help an impulsive model, and an underachieving model does not benefit from improved selection. When no single phase dominates, such as for Qwen3.8-27B and Qwen3.8-Flash-Next, the metrics still locate the weakness. Their establishment matches discrimination, so the approaches these models commit to are largely executed to a certificate, but potential indicates that only half of the proposals can succeed and discrimination shows that a third of their commitments cannot. Improvement for these models lies in the quality of what they propose and choose rather than in their ability to execute.

Each phase requires an ability that extends beyond the benchmark. A model that performs well on SciVerge has shown it can generate approaches with the potential to succeed, on systems drawn from several scientific domains. It has shown it can judge, among candidate approaches, which are worth attempting. And it has shown it can execute an approach and prove its result. Each of these is required in scientific work: a researcher facing an open question must conceive of ways to solve it, decide which of them to spend their time and resources on, and finally, carry out the selected approach and justify the results. The extent to which a model can perform each phase of this process is what SciVerge measures.

6 Conclusion

We presented SciVerge, a benchmark of true-or-false scientific questions whose mathematical basis allows not just the answer, but each proposed approach to be scored objectively on whether it can succeed. We construct a three-phase task to record the approaches a model proposes, those it selects, and whether a selected approach produces a proof, with six metrics drawn from the psychology of creativity to score the phases. Across seven models we find three failure modes, one per phase, and that current models disprove claims far more readily than they prove them. SciVerge offers an objective measurement of how a model proposes, selects, and executes a scientific idea, which we hope will guide the development of models that do all three.

7 AI USE STATEMENT

We used generative AI tools to assist or provide feedback in developing the conceptual framework and methodology, formulating mathematical claims, proof writing, implementing the methods, and interpreting results. All AI-assisted work was reviewed by the authors: the proof was checked by hand, code was tested against independent computations, and every reported number was recomputed from the evaluation logs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

8 ETHICS STATEMENT

SciVerge involves no human subjects and no personal data. Every instance is generated from published mathematical descriptions of scientific systems and verified by proof, so the benchmark was built without annotators. The questions concern the mathematical structure of abstract systems, and we do not foresee harmful applications of performance on them beyond those of general improvements in mathematical reasoning.

9 REPRODUCIBILITY STATEMENT

The evaluation set, the generation and screening pipeline, the verification procedure, and the harness that runs the three-phase task are available at `https://anonymous.4open.science/r/SciVerge-9DD4/` and as supplementary material. Section 2.2 defines the certificates and Appendix A.1 proves their soundness. Section 2.4 and Appendix A.3 describe how instances are generated and how each presentation is checked against the literature. Appendix A.6 gives the serving configuration, token budgets, decoding settings, and run statistics, Appendix A.4 gives results by question type and by answer, and the prompts, parser, and evaluation logs from which every number in the paper is computed are included with the code.

REFERENCES

Aaron Abrams and Robert Ghrist. Finding topology in a factory: configuration spaces. *The American mathematical monthly*, 109(2):140–150, 2002.

SI Adian. Algorithmic undecidability of problems of recognition of certain properties of groups. In *Dokl. Akad. Nauk SSSR*, volume 103, pp. 533–535, 1955.

Atilla Kaan Alkan, Shashwat Sourav, Maja Jablonska, Simone Astarita, Rishabh Chakrabarty, Nikhil Garuda, Pranav Khetarpal, Maciej Pióro, Dimitrios Tanoglidis, Kartheik G Iyer, et al. A survey on hypothesis generation for scientific discovery in the era of large language models. *arXiv preprint arXiv:2504.05496*, 2025.

Iris Anshel, Michael Anshel, and Dorian Goldfeld. An algebraic method for public-key cryptography. *Mathematical Research Letters*, 6(3):287–291, 1999.

Mois Ilia Aroyo et al. *International tables for crystallography*. Wiley Online Library, 2013.

Emil Artin. Theory of braids. *Annals of Mathematics*, 48(1):101–126, 1947.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Ruth Charney. An introduction to right-angled Artin groups. *Geometriae Dedicata*, 125(1):141–158, 2007.

Harold SM Coxeter and William OJ Moser. *Generators and relations for discrete groups*. Springer Science & Business Media, 2013.

Claus Ernst and DeWitt Sumners. A calculus for rational tangles: applications to DNA recombination. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 108, pp. 489–515. Cambridge University Press, 1990.

Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

Robert Ghrist. Configuration spaces and braid groups on graphs in robotics. *AMS IP Studies in Advanced Mathematics*, 24:29–40, 2001.

Daniel Gottesman. *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pp. 1–3, 2026.

Joy Paul Guilford. *The nature of human intelligence*. McGraw-Hill, 1967.

Sikun Guo, Amir Hassan Shariatmadari, Guangzhi Xiong, Albert Huang, Myles Kim, Corey M Williams, Stefan Bekiranov, and Aidong Zhang. IdeaBench: Benchmarking large language models for research idea generation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5888–5899, 2025.

Erik E Guzik, Christian Byrge, and Christian Gilde. The originality of machines: AI takes the Torrance Test. *Journal of Creativity*, 33(3):100065, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

Ki Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang, and Choonsik Park. New public-key cryptosystem using braid groups. In *Annual International Cryptology Conference*, pp. 166–183. Springer, 2000.

Hugh C Longuet-Higgins. The symmetry groups of non-rigid molecules. *Molecular Physics*, 6(5): 445–460, 1963.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Michael D Mumford, Devin C Lonergan, and Ginamarie Scott. Evaluating creative ideas: Processes, standards, and context. *Inquiry: Critical thinking across the disciplines*, 22(1):21–30, 2002.

Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*, volume 2. Cambridge university press Cambridge, 2001.

OpenAI. On the Navier–Stokes Millennium Prize problem. `https://openai.com/index/navier-stokes-solution/`, September 2026. Accessed September 2026.

Michael O Rabin. *Recursive unsolvability of group theoretic problems*. Princeton University, 1956.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Ivan N Sanov. A property of a representation of a free group. In *Doklady Akad. Nauk SSSR (NS)*, volume 57, pp. 16, 1947.

Jean-Pierre Serre. *Trees*. Springer Science & Business Media, 2002.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, volume 2025, pp. 94003–94092, 2025.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation-execution gap: Execution outcomes of LLM-generated versus human research ideas. In *International conference on learning representations*, volume 2026, pp. 74195–74240, 2026.

E Paul Torrance. *Torrance tests of creative thinking: Interpretive manual*. Scholastic Testing Service, Bensenville, IL, 2018.

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.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

A Appendix

A.1 Proof of Proposition 2.1

Proof. For $y = \mathrm{T}$, each step of an accepted derivation inserts or deletes a relation or a canceling pair, so each step preserves $=_P$, and the derivation exhibits $\varphi_\tau(\theta)$ directly. For $y = \mathrm{F}$, an accepted world ρ sends every relation to the identity, so ρ extends to a homomorphism from the group presented by P to a group of permutations or matrices; equal words therefore act identically under ρ, and two words that act differently cannot be equal, so $\varphi_\tau(\theta)$ fails in P. For periodicity, the same argument applies to the infinite-order witness: if $w^n =_P \varepsilon$ for some n, then $\rho(w)^n$ is the identity, contradicting infinite order. □

A.2 Question Types

Table 5 gives the formal definition of each question type in terms of word equality, with the property $\varphi_\tau(\theta)$ that an instance of that type asks about.

Table 5: The five question types, the property each asks about, and the certificate accepted for each verdict. $\langle \mathtt{h}_1, \ldots, \mathtt{h}_k \rangle$ denotes the words assembled from the $\mathtt{h}_i$ and their reverses. A world is an assignment of a permutation or matrix to each action under which every relation acts as the identity; it separates two words when they act differently under it.

τ	$\varphi_\tau(\theta)$	Certificate if T	Certificate if F
Equivalence	$\mathtt{u} =_P \mathtt{v}$	derivation $\mathtt{u} \to \mathtt{v}$	world separating $\mathtt{u}$, $\mathtt{v}$
Compatibility	$\mathtt{uv} =_P \mathtt{vu}$	derivation $\mathtt{uv} \to \mathtt{vu}$	world separating $\mathtt{uv}$, $\mathtt{vu}$
Conjugacy	$\exists\, \mathtt{x}$: $\mathtt{Xux} =_P \mathtt{v}$	$\mathtt{x}$, derivation $\mathtt{Xux} \to \mathtt{v}$	world with no such $\mathtt{x}$
Membership	$\exists\, \mathtt{e} \in \langle \mathtt{h}_1, \ldots, \mathtt{h}_k \rangle$: $\mathtt{w} =_P \mathtt{e}$	$\mathtt{e}$, derivation $\mathtt{e} \to \mathtt{w}$	world separating $\mathtt{w}$ from every $\mathtt{e}$
Periodicity	$\exists n \geq 1$: $\mathtt{w}^n =_P \varepsilon$	n, derivation $\mathtt{w}^n \to \varepsilon$	matrix world, $\mathtt{w}$ of infinite order

A.3 Systems and Transcription Check

Table 6 lists every family in SciVerge with its field, its source in the literature, and the size of its presentation.

The transcription of each presentation is verified computationally. We map the actions of a presentation onto permutations or matrices and check that every relation acts trivially, compute the abelianization from the relations directly, and compare both against the values recorded for the group the literature describes. For the finite families this settles the identification, since a map onto the named group together with a matching order is an isomorphism. For the rest, the check runs in one direction: a disagreement proves the transcription wrong, while agreement supports the identification without settling it. No general algorithm can determine whether two presentations describe the same group, as this task is undecidable (Adian, 1955; Rabin, 1956).

A.4 Results by Answer and by Question Type

Table 7 gives potential, discrimination, and establishment on true and on false instances separately, with reasoning enabled. Table 8 gives establishment by question type for every run. On true instances the order of difficulty is the same for every model that establishes at all: equivalence is easiest (12 to 46 percent), then membership and compatibility, then conjugacy (0 to 12), then periodicity (0 to 2). Periodicity on a true instance requires a derivation from w^n to the empty word, which is the longest derivation in the evaluation set. On false instances the spread across types is smaller and no type is uniformly hardest.

Budget sensitivity on true instances. Because execution calls on true instances often reach the 64,000-token budget (Section 3.2), the three models with the highest truncation rates were rerun on the 250 true instances with a 128,000-token budget, reasoning enabled. Establishment on true instances rose from 17.2 to 26.4 for Qwen3.8-27B, with the fraction of execution calls reaching

Table 6: The 21 families, grouped by the system they describe. $|X|$ and $|R|$ are the numbers of actions and defining relations; $|R| = 0$ marks a system whose actions satisfy no relations. Instance counts are for corpus `v0.6.4` (100,012 instances).

System and families	$\lvert X\rvert$	$\lvert R\rvert$	Instances
Quantum computing. Anyons in a row, exchanged in adjacent pairs (Artin, 1947; Freedman et al., 2003)			
`braid_B3`	2	1	3,002
`braid_B4`	3	3	5,792
`braid_B5`	4	6	6,386
Quantum computing. Pauli operators on a block of qubits (Gottesman, 1997; Nielsen & Chuang, 2001)			
`pauli_P3`	7	28	7,786
Molecular biology. A knotted molecule and a probe loop (Ernst & Sumners, 1990)			
`torus_knot_T(2,3)`	2	1	2,992
`torus_knot_T(2,5)`	2	1	4,976
`torus_knot_T(3,4)`	2	1	4,004
`figure8_4_1`	2	1	3,126
Robotics. Two vehicles moving along a network (Ghrist, 2001; Abrams & Ghrist, 2002)			
`graph_braid_B2K5`	7	1	5,354
`graph_braid_B2K33`	5	1	4,982
`graph_braid_B2star5`	6	0	990
Concurrency. Stations whose neighbors share a resource (Charney, 2007)			
`raag_path_P3`	3	2	5,786
`raag_path_P4`	4	3	5,566
Crystallography. A periodic plane pattern (Aroyo et al., 2013; Coxeter & Moser, 2013)			
`wallpaper_p2`	3	4	5,228
`wallpaper_p3`	3	4	6,254
`wallpaper_p4`	3	4	6,084
`wallpaper_p6`	3	4	6,490
Chemistry. Tiles rearranged by adjacent swaps (Longuet-Higgins, 1963)			
`symmetric_S4`	3	6	5,180
`symmetric_S5`	4	10	6,034
Number theory. A basis of a two-dimensional lattice (Serre, 2002)			
`SL(2,Z)`	2	3	3,136
Number theory. Compositions of two shear maps (Sanov, 1947)			
`sanov_F2`	2	0	864

Table 7: Potential, discrimination, and establishment on true and false instances separately, in percent, with reasoning enabled. A true instance is proven by a derivation and a false one by a world.

	True instances			False instances		
Model	Pot.	Disc.	Est.	Pot.	Disc.	Est.
Qwen3.5-9B	33.3	31.7	0.0	52.2	50.0	2.0
Qwen3.6-27B	27.3	33.8	5.6	63.1	72.8	55.2
Qwen3.8-27B	34.2	46.8	17.2	63.3	80.4	83.2
Qwen3.8-Flash-Next	32.5	40.8	24.4	62.5	78.3	86.0
DeepSeek-V4-Flash	32.0	32.1	11.2	63.8	71.5	70.0
Gemma-4-31B	25.0	23.9	6.8	71.2	73.3	57.6
GLM-4.7-Flash	26.9	20.6	0.0	55.8	51.0	1.2

the budget falling from 56 to 11 percent, so the 64k figure for that model is budget-bound. For DeepSeek-V4-Flash it rose from 11.2 to 14.4 (truncation 82 to 62 percent) and for Qwen3.6-27B from 5.6 to 6.8 (78 to 74 percent); for these two the larger budget is consumed by reasoning that does not reach a derivation, and their true-side figures remain lower bounds. Periodicity on true instances stayed at or below 10 percent for all three at 128k. The main text reports the 64k run throughout so that all models are compared at one budget.

Table 8: Establishment by question type, in percent, for every run in Table 3. Each cell is over the 100 instances of that type, 50 true and 50 false.

Model	Reas.	Equivalence	Compatibility	Conjugacy	Membership	Periodicity
Qwen3.5-9B	✓	0.0	0.0	1.0	0.0	4.0
	×	10.0	0.0	5.0	0.0	0.0
Qwen3.6-27B	✓	41.0	24.0	34.0	28.0	25.0
	×	14.0	10.0	15.0	1.0	7.0
Qwen3.8-27B	✓	59.0	53.0	51.0	54.0	34.0
	×	31.0	14.0	12.0	2.0	39.0
Qwen3.8-Flash-Next	✓	69.0	48.0	53.0	62.0	44.0
	×	18.0	9.0	7.0	2.0	6.0
DeepSeek-V4-Flash	✓	55.0	33.0	41.0	35.0	39.0
	×	12.0	1.0	7.0	2.0	7.0
Gemma-4-31B	✓	46.0	30.0	33.0	24.0	28.0
	×	30.0	16.0	7.0	0.0	8.0
GLM-4.7-Flash	✓	2.0	1.0	0.0	0.0	0.0
	×	4.0	0.0	4.0	1.0	1.0

A.5 Viability Labels

Each question type offers a fixed set of approach kinds: eight for equivalence, seven for compatibility, nine for conjugacy, seven for membership, and five for periodicity. Kinds ending in a world (a world of a named size, a matrix world of a named dimension, a world given by a named finite symmetry group, and the count of actions) can be viable only on false instances, and kinds ending in a derivation only on true ones, so by soundness half of the labels are fixed with no search. For the rest, viability is decided by a bounded search run once per instance when it is minted: the derivation using all relations is viable on every true instance, since every true instance is generated with one; worlds are searched over permutation actions on at most 20 points; matrix worlds are searched in dimension 2 with entries in $\{-1, 0, 1\}$, with the trace and determinant as the separating invariants for conjugacy; symmetry worlds are checked against a fixed list of named finite groups; and the count is evaluated exactly. Each search is capped at 10 seconds per approach. Over the 2,600 approach-instance pairs in the evaluation set, 1,046 are viable, 1,500 are not, and 54 are unlabeled: 44 matrix worlds whose search was exhausted or timed out, and 10 small worlds on instances whose catalog was capped. Unlabeled pairs leave the denominators of potential, discrimination, and restraint. Approaches whose parameters the model supplies freely (a helper identity, an intermediate word, a conjugator, or an assembly) are not labeled in advance and are likewise excluded from those three metrics; they count toward fluency and flexibility, and a certificate produced through one of them counts toward establishment. A label is raised to viable, and every run re-scored, if a model produces an accepted certificate through an approach labeled non-viable; a label is never lowered on the basis of a failed attempt.

Table 9: Run statistics for every run in Table 3, in percent except the mean number of commitments. Settled is the fraction of instances with any accepted certificate; establishment counts only certificates produced through the committed approach. Truncation is the fraction of calls that reached the 64,000-token budget.

						Truncation	
Model	Reas.	Plan parsed	Commitments	Settled	Establishment	Call 1	Call 2
Qwen3.5-9B	✓	8.2	0.21	1.0	1.0	91.8	68.3
	×	97.8	2.18	3.0	3.0	1.2	16.6
Qwen3.6-27B	✓	81.6	1.59	34.4	30.4	17.0	59.2
	×	99.6	2.30	10.6	9.4	0.0	16.8
Qwen3.8-27B	✓	97.0	2.11	56.6	50.2	1.8	39.8
	×	99.2	2.56	21.4	19.6	0.0	1.1
Qwen3.8-Flash-Next	✓	97.6	2.22	60.0	55.2	2.2	24.0
	×	100.0	2.40	8.6	8.4	0.0	0.2
DeepSeek-V4-Flash	✓	96.0	2.09	47.4	40.6	4.0	58.5
	×	99.8	2.98	7.4	5.8	0.0	7.2
Gemma-4-31B	✓	94.0	2.10	35.4	32.2	6.0	12.4
	×	100.0	2.35	13.4	12.2	0.0	7.4
GLM-4.7-Flash	✓	27.0	0.57	0.6	0.6	72.2	91.5
	×	98.2	2.69	2.0	2.0	1.8	20.2

A.6 Evaluation Details

Protocol mechanics. Each instance is presented in two model calls. The first call carries the instance in the field's own description (as in Table 1) and the list of approach kinds available for its question type, and asks for up to five proposals, each a kind with the parameters the model would use, followed by an ordered list of at most three of them to commit to. A validator rejects any certificate content in this call, so proposals are made before any proof is attempted. The second call is made once per committed approach in order, asks the model to execute that approach and return a verdict and certificate in the format of Section 2.2, and stops at the first accepted certificate. A first call whose output cannot be parsed as a plan is scored as zero proposals, and an instance with no commitments has no second call. The full prompts and the parser are included with the code.

Serving and decoding. All seven models were served locally with vLLM through its OpenAI-compatible endpoint, at temperature 0.2 with a budget of 64,000 completion tokens per call, reasoning tokens included. Reasoning was enabled or disabled explicitly through the chat template for both arms of every model, so that the two arms differ only in that setting. One attempt was made per instance and no attempt was voided. The evaluation set is corpus draw `22adcfb2`, 500 instances, 100 per question type, 50 true and 50 false in each.

Run statistics. Table 9 gives, for every run, the fraction of first calls that produced a parsable plan, the mean number of commitments, the fraction of instances on which any certificate was accepted, establishment (an accepted certificate on the committed approach), and the fraction of first and second calls that reached the token budget. The difference between the third and fourth columns is the rate of off-plan certificates: accepted certificates that the model produced through an approach other than the one it committed to, which do not count toward establishment. These are 3 to 7 percent for the capable models with reasoning enabled and near zero otherwise. The two overcautious runs are visible in the plan column: with reasoning enabled, Qwen3.5-9B and GLM-4.7-Flash produced a parsable plan on 8 and 27 percent of instances, their first calls having reached the budget on 92 and 72 percent, which is why their fluency collapses in Table 3.

Comparison with a closed task. To check that the three-phase task does not itself depress the solve rate, three models were also run on the same 500 instances as a closed task, a single call that presents the instance and asks for a verdict and certificate with no proposals or commitments, at the same budget with reasoning enabled. At matched budget, one closed call against the first committed approach, the closed task certifies more instances: Qwen3.8-27B 56.4 against 41.0 percent, DeepSeek-V4-Flash 42.6 against 30.6, and Qwen3.5-9B 6.6 against 0.6. Allowing the three-phase

task its full commitment list closes the gap: the fraction of instances with any accepted certificate is 56.6 against 56.4 for Qwen3.8-27B and 47.4 against 42.6 for DeepSeek-V4-Flash. The three-phase task therefore settles as many questions as the closed task; what it costs, the difference between settled and establishment in Table 9, is the price of committing first to an approach that cannot succeed, which is the discrimination error the benchmark is designed to expose. The model ranking is the same under both tasks.