

KNOWING WHEN TO STOP: IGP-BENCH AND THE MULTI-AGENT SYSTEM GALOIS FOR THE INVERSE GALOIS PROBLEM

Shaowu Zhang* Jan Safrata Brent Kong Eric Lu
Sathvik Redrouthu Caiman Moreno-Earle Tony Yue Yu†

California Institute of Technology

ABSTRACT

Frontier agents can now sustain hours of autonomous work. Greater capability may bring greater progress, but it may also carry an agent farther along a wrong route. Existing long-horizon mathematics benchmarks are difficult to build, saturate quickly, or rely on language models to grade partial progress, making it difficult to measure how much effort an agent spends on an unproductive route. We introduce **IGP-Bench**, a benchmark of 200 long-horizon inverse Galois problems from computational algebraic geometry, only 39 of which have published explicit solutions. Each problem has intermediate results that are difficult to find but easy to check with deterministic GAP and Magma verifiers. Agents record their partial results and attempts in an artifact graph, allowing us to measure reasoning, instruction following, tool use, resource allocation, and, most importantly, analyze when agents stop or redirect their work. Across five problems, 27 intermediate subproblems, and more than 32,000 possible routes, we show that detailed analysis of long-horizon work reveals valuable insights that success rates alone do not. Claude Opus 5 achieves the highest mean progress score (81.0), followed by Claude Fable 5.1 (62.4), GPT-6 Sol (62.1), and GPT-6 Astra (59.4); newer models can spend substantial effort on unproductive routes. To address this failure mode, we introduce **GALOIS, an open-source multi-agent system** designed to identify promising routes and stop unproductive work. It comprises a lead agent, a strategy reviewer, specialist agents equipped with 36 general algebraic geometry skills, and a route-rejection agent that checks for possible obstructions in parallel. In reference runs led by Claude Opus 5.5, GALOIS obtains Magma-verified realizations for all five tasks in 10–126 minutes, including polynomials for M_{23} and M_{24} that no single agent realizes in our evaluation. These results highlight **knowing when to stop** as an important ability that current benchmarks rarely measure. We will release the tasks, verifiers, evaluation code, and GALOIS.

1 INTRODUCTION

AI agents can now work autonomously for hours on tasks of increasing difficulty and duration (Kwa et al., 2025a), but greater capability does not necessarily bring greater awareness of when they are doing things wrong. For example, OpenAI reported a long-running model mistakenly sustained an intrusion into Hugging Face’s infrastructure for two and a half days, illustrating how persistence on a wrong route could be dangerous (OpenAI, 2026b; Hugging Face, 2026; OpenAI, 2026a; Greenblatt et al., 2026). Similar issues have been observed in AI-assisted mathematical research. In recent realization of M_{23} as a Galois group over $\mathbb{Q}$ by Huang et al. (2026), the most capable models were also the most persistent on routes that eventually failed. According to the authors’ AI disclosure and reflection (Zhang, 2026; Chiou, 2026), agents keep suggesting going further on a wrong route, but eventually it’s humans that decided when to stop unproductive computations and identified the

*Corresponding author: szhang7@caltech.edu

†Principal investigator

essential obstruction which led to the final success. This raises a question: do AI agents know when to stop pursuing a wrong route, and how well can they make this judgment?

Inverse Galois theory provides a particularly suitable setting for evaluating whether research agents know when to stop. These problems have recently become a focus of AI-assisted mathematical research: SAIR’s IGP24 competition drew 256 participants over two months to search for explicit degree-24 polynomials realizing transitive groups of degree 24 over $\mathbb{Q}$ (SAIR Foundation, 2026); AIM’s 2026 workshop on AI and number theory posed the question of realizing the last sporadic simple group M_{23} (American Institute of Mathematics, 2026); Epoch AI features the M_{23} problem in FrontierMath: Open Problems (Epoch AI, 2026); and *Scientific American* also recently covered the development of this field (Chiou, 2026). As pointed out by experts in inverse Galois theory (Antieau, 2026), it is of essential difficulty of deciding when to stop a computation and move to another candidate.

To introduce IGP-Bench, we first introduce some mathematical background. For a finite group G , the inverse Galois problem asks whether there is a polynomial $f \in \mathbb{Q}[V]$ whose Galois group is isomorphic to G . We call an approach to constructing such a polynomial a *route*. Each route is specified by a list of conjugacy classes of G , called a *class vector*, for example $(2A, 23A, 23B)$ of M_{23} . Our literature survey provides a reference set of known routes, while historical attempts suggest that a generic route is unlikely to succeed. The difficulty is that a class vector may pass the initial group-theoretic tests yet lead to a cover that cannot be defined over $\mathbb{Q}$. Simply choosing a class vector and pushing its construction further therefore need not bring the target polynomial within reach. This makes route selection and stopping decisions a substantive part of the mathematics. The case of M_{23} illustrates the challenge (Figure 2): its 16 nontrivial conjugacy classes give 816 three-point and 3,876 four-point routes, while only $(2A, 23A, 23B)$ is currently known to yield a realization over $\mathbb{Q}$ (Huang et al., 2026). An agent must therefore decide which candidates justify further computation and which should be set aside as new evidence merges.

IGP-Bench uses these problems to evaluate agents. Each of its 200 tasks asks for a realization of a finite group over $\mathbb{Q}$. We divide the work into three stages: *screening*, which identifies candidate routes and tests their group data; *route identification*, which verifies the group data for a literature-supported route; and *construction*, which follows a selected route through its intermediate objects toward a polynomial. A task comes with a *checkpoint ladder*: a sequence of intermediate objects checked by computer-algebra verifiers. For each problem, agents record intermediate results and attempted routes in a labelled *artifact graph*, with resource costs such as CPU hours and output tokens attached to artifacts.

From these artifacts, we compute a progress score P and assess instruction following and tool use. We quantify stopping behavior with the knowing-when-to-stop metric K_r and resource use with the resource efficiency metric U_r , without an LLM judge (Appendix E).

We evaluated nine frontier-model systems on IGP-Bench, spanning open-weight and proprietary models: Claude Opus 5, Claude Fable 5.1, Claude Sonnet 5, GPT-6 Sol, GPT-6 Astra, GPT-5.6 Sol, GPT-5.6 Terra, DeepSeek V4 Pro, and GLM-5.3 Flash. We ran each system twice on each of five tasks targeting the Mathieu groups M_{11} , M_{12} , M_{22} , M_{23} , and M_{24} , for 90 single-agent runs in total. Each run had a 10-hour wall-clock limit, a 500,000-output-token cap, and access to eight CPUs. Mean progress scores ranged from 2.6 to 81.0, with Claude Opus 5 at the top (81.0), outperforming the newer Claude Fable 5.1 (62.4), GPT-6 Sol (62.1), and GPT-6 Astra (59.4) (Figure 1). Among many other findings in Section 6, the following two are particularly interesting.

1. **Going deeper does not always imply better score.**
2. **Knowing when to stop and going deeper are largely distinct abilities.**

We also introduce GALOIS, an open-source multi-agent system. Its lead agent plans the work, while specialist agents follow stage-specific procedures based on general methods from the literature. In

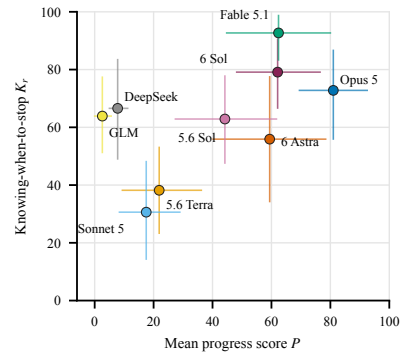


Figure 1: Progress and knowing when to stop by system, means with 95% bootstrap intervals.

parallel, a route-rejection agent checks candidate routes for known obstructions. It can remove a route from consideration only when an exact obstruction is proved.

In separate reference runs led by Claude Opus 5.5, GALOIS obtained verified realizations for all five tasks within 126 minutes each, including polynomials for M_{23} and M_{24} that no single agent realized in our evaluation. Our contributions are IGP-Bench and its metrics (Section 3), GALOIS (Section 4), and the initial evaluation (Sections 5–7).

2 RELATED WORK

Mathematics benchmarks and partial credit. FrontierMath and Riemann-Bench score final answers to difficult problems without partial credit, and HorizonMath ranks solutions that improve a published bound (Glazer et al., 2024; Garre et al., 2026; Wang et al., 2026). IMProofBench pairs research-level proof problems, graded by human experts, with automatically graded follow-up sub-questions, while TheoremBench evaluates machine-verified theorems in Lean (Schmitt et al., 2025; Pham et al., 2026). An IGP-Bench task instead specifies a group and a checkpoint ladder. The agent produces intermediate mathematical objects, and a computer-algebra verifier checks them. The M_{23} realization is prior work (Huang et al., 2026; Zhang, 2026), not a contribution of this paper.

AI systems, research agents, and multi-agent systems. FunSearch and AlphaEvolve have used search to find new constructions (Romera-Paredes et al., 2024; Novikov et al., 2025). QED, an open-source multi-agent system, separates planning, proving, and verification and has produced original proofs on research-level projects (An et al., 2026); Aletheia and the AI co-mathematician also target research-level mathematics (Feng et al., 2026; Zheng et al., 2026). PaperBench and EXP-Bench evaluate parts of research workflows using learned judges (Starace et al., 2025; Kon et al., 2026). MetaGPT, AutoGen, and CAMEL assign agents different roles (Hong et al., 2024; Wu et al., 2023; Li et al., 2023). GALOIS also uses a lead agent and specialists, but adds two requirements: mathematical results must pass verification, and explicit rules govern when to drop a route. Appendix A lists further work.

Cost, overthinking, and stopping. Kapoor et al. (2025) argue that accuracy-only benchmarks can reward systems that are unnecessarily expensive. Earlier work shows that reasoning models can spend computation where it is not needed (Chen et al., 2025b; Sui et al., 2025), and that extended reasoning in agentic tasks can displace useful interaction with the environment (Cuadron et al., 2025). Adapting test-time computation can work better than allocating it uniformly (Snell et al., 2024), while longer reasoning can reinforce distractions and mistaken strategies (Gema et al., 2025). The abstention benchmark AgentAbstain scores one decision per task: on paired everyday tasks, the agent should either act or abstain, and each task is scored pass or fail (Liu et al., 2026). IGP-Bench scores many decisions per run: over hours, the agent repeatedly decides whether to continue a route, and K_r measures what wrong routes cost, with routes labeled by a literature ledger (Table 1).

Table 1: Related benchmarks. *Long-horizon*: an agent works on one task for hours ($\triangle$: multi-step runs that are shorter or of unreported length). *Partial credit*: intermediate progress earns credit. *Deterministic*: grading avoids human or learned judges. *Task-derived*: intermediate states come from solving the task, not from a list written in advance. *Stopping*: whether and how stopping decisions are measured. $\checkmark$ yes, $\triangle$ partly, $\times$ no; N/A: no partial credit is given.

Benchmark	Long-horizon	Partial credit	Deterministic	Task-derived	Stopping
FrontierMath	$\times$	$\times$	$\checkmark$	N/A	$\times$
Riemann-Bench	$\triangle$	$\times$	$\checkmark$	N/A	$\times$
HorizonMath	$\times$	$\times$	$\triangle$	N/A	$\times$
IMProofBench	$\triangle$	$\checkmark$	$\triangle$	$\times$	$\times$
TheoremBench	$\times$	$\checkmark$	$\checkmark$	$\times$	$\times$
PaperBench	$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$
EXP-Bench	$\triangle$	$\checkmark$	$\triangle$	$\times$	$\times$
AgentAbstain	$\times$	$\times$	$\triangle$	N/A	binary per task: act or abstain
IGP-Bench	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	continuous per route: time, tokens, CPU hours

3 IGP-BENCH

3.1 BACKGROUND

The goal is to find a polynomial with rational coefficients and Galois group G . A common approach starts with a *cover*, a map from an algebraic curve to the line, and uses it to build a polynomial family $F(T, V)$. The family is *regular* if $\mathbb{Q}$ is algebraically closed in its splitting field over $\mathbb{Q}(T)$. For a regular family, infinitely many rational choices of T give a polynomial over $\mathbb{Q}$ with Galois group G (Serre, 1992; Malle & Matzat, 2018). Each route is specified by a class vector, a list of conjugacy classes. Group-theoretic tests can rule out some routes before the agent computes any equations. Passing those tests, however, is not the same as constructing the polynomial. Neither is proving that a realization exists. We therefore score route identification and construction separately. Appendix B gives the definitions and construction steps.

3.2 TASKS

Three tiers. Each of the 200 tasks specifies a finite group G and asks for a polynomial $f \in \mathbb{Q}[V]$ with $\text{Gal}(f/\mathbb{Q}) \cong G$. Domain experts designed the tasks and assigned their tiers from the cited literature. On 25 September 2026, we checked every assignment against Zywin’s census of small simple groups not known to occur over $\mathbb{Q}$, the LMFDB number-field tables, and Magma’s Klüners–Malle database (Zywin, 2025; The LMFDB Collaboration, 2026). The 39 *Tier 1* tasks have a published polynomial or regular family. The 55 *Tier 2* tasks have an existence proof but no explicit polynomial in the cited sources. The 106 *Tier 3* tasks remain open. Six are the small simple groups left open by the census. Like the Tier 1 and Tier 2 tasks, these six have a registered permutation action and a checkpoint ladder. The others are listed in a research registry and use the open-case scoring profile (Appendix H.2). Appendix E explains scoring without reference answers. The evaluation in Section 6 used the 19 September 2026 build. A later audit revised tier assignments and replaced two tasks.

Checkpoints. Most tasks have five checkpoints: a generating tuple, a braid orbit, a cover, a regular family, and a final polynomial. The first two check the group data describing candidate covers; the last three check the construction itself. A generating tuple contains one element from each class in the class vector. The elements must generate G and multiply to the identity. These tuples, up to conjugation, form the route’s *Nielsen class*. A *braid orbit* in this class corresponds to a connected family of covers. Six tasks use different checkpoint sequences based on known constructions (Appendix E). GAP or Magma checks the required mathematical properties at each checkpoint. Credit comes from passing these checks, not from matching a reference artifact. Agents cannot access the reference artifacts.

Route ledger. The *route ledger* lists class vectors supported by the literature and any known obstructions. We use it to score route identification and resource use. A route is *correct* when the literature supports it for the target group or its automorphism group, or when the run itself produces a verified polynomial on it with stored answers blocked. All other routes are assumed *wrong*, including those whose status is unknown. Wrong does not mean mathematically impossible. Appendix E gives the registration rules, and Section 7 discusses the limits of these labels.

What the agent sees. The agent receives the target group, objective, checkpoint descriptions, and a fixed budget. It works in a persistent sandbox with a shell, Python, GAP, Magma, and an offline corpus of 17 documents (Appendix D). We exclude documents that reveal the task’s realization and disable network access. The agent submits intermediate results in a required format. A formatter checks the syntax, but verifier results are hidden until the run ends. During the run, the agent must assess its own mathematical progress.

3.3 EXAMPLE: M_{23}

The M_{23} task illustrates the three stages and kinds of progress we score. In *screening*, the agent lists candidate routes and tests their group data. There are 816 three-point and 3,876 four-point class vectors, and verified facts about them earn screening credit. In *route identification*, the agent must

verify the group data for the literature-supported route $(2A, 23A, 23B)$, not simply name it. This route has seven candidate covers, exactly one of which is defined over $\mathbb{Q}$ (Huang et al., 2026). In *construction*, the agent follows the selected route by computing the cover’s equations, expressing them over $\mathbb{Q}$, and obtaining and specializing a regular family (Appendix C). No single agent in our evaluation reached the exact cover equations on this route. Instead, some constructed covers over larger number fields on routes already ruled out over $\mathbb{Q}$ (Figure 2).

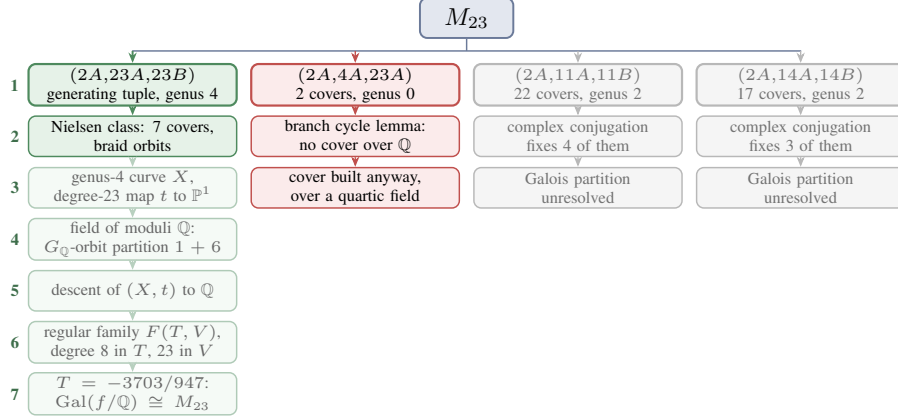


Figure 2: **Routes on M_{23} .** Green: the correct route (Huang et al., 2026), one box per checkpoint stage (Appendix C); full boxes were verified by at least one single-agent run, faded boxes by none, and the GALOIS reference runs complete the column. Red: a genus-0 route ruled out by the branch cycle lemma, on which several agents still built a cover over a quartic field. Grey: genus-2 routes with neither a known cover over $\mathbb{Q}$ nor a proved obstruction.

3.4 METRICS

Artifact graph. A *run* is one system’s attempt at one task. The agent records its work as a directed graph. Each node is an *artifact*: a declared unit of work containing tool calls, outputs, and checkpoint submissions. The agent also records which artifacts depend on which others. The evaluator assigns model and tool costs to the active artifact and reports unattributed costs separately. Mathematical verifiers determine progress.

Progress score. The progress score adds three components:

$$P = S + R + C, \quad 0 \leq S \leq 30, \quad R \in \{0, 30\}, \quad 0 \leq C \leq 40. \quad (1)$$

The *screening score* S rewards verified facts about candidate routes and constructions, up to 30 points. Each fact earns credit once, with a weight based on the strength of its evidence. The *route-identification score* R gives 30 points for verifying a correct route. The *construction score* C depends on the deepest construction checkpoint the agent verifies. A verified polynomial earns all 40 construction points, regardless of how the agent found it. Appendix E gives the acceptance rules and point schedules.

Tables report S , R , and C as percentages of their respective maxima. **POLY** is the fraction of runs that produce a verified polynomial.

Knowing-when-to-stop and resource efficiency metrics. We define knowing when to stop as ending or redirecting work on an unproductive route before it uses up the budget needed for other routes. We quantify this with the *knowing-when-to-stop metric* K_r and resource use with the *resource efficiency metric* U_r , both on a 0–100 scale and using output tokens unless stated otherwise.

To compute K_r , we group artifacts by class vector and separate initial screening from later research on individual routes. Let B_r be the total screening cost, and let $\bar{C}_r^+$ and $\bar{C}_r^-$ be the mean post-

screening costs per correct and wrong route. Then

$$K_r = 100 \frac{B_r + \bar{C}_r^+}{B_r + \bar{C}_r^+ + \bar{C}_r^-}. \quad (2)$$

A higher K_r indicates a lower wrong-route cost relative to screening plus an average correct route. Including screening allows the score to account for useful preliminary work even before the agent reaches a correct route. We report the number of researched routes alongside the score.

U_r is the share of recorded cost spent on credited artifacts and the artifacts they depend on. Together, K_r and U_r distinguish choosing routes selectively from making verified progress on the routes chosen. Figure 6 illustrates both scores, and Appendix E provides the accounting rules.

4 GALOIS

GALOIS builds stopping decisions into its research process through a background route rejection agent and resource-limited experiments.

Control loop. At each step, a lead mathematician agent reads the plan, journal, and artifacts, then proposes tool calls, specialist work, or completion. A deterministic checker validates the proposal’s format. An optional strategy reviewer checks whether the proposal is supported by the evidence, advances the work, and uses appropriate tools. The runner launches resource-limited subprocesses or specialist sessions and records their outputs for the next step (Figure 7). Each skill is a written procedure for one pipeline stage. The revision used for these runs contains 36 skills and 122 registered tools, covering candidate enumeration, cover reconstruction, descent, and specialization (Table 7).

Stopping before and during construction. Initial screens and ranking select candidates using generating counts, class compatibility, Nielsen count, and genus. Gröbner calls have time limits. When exact recognition fails, the next step must be a justified precision increase, a coordinate change, another resource-limited test, or deferral, since more digits do not fix unsuitable coordinates (Section 1). For deferred work, the evidence ledger records what needs repair and when the work should be revisited. Exact field-of-moduli tests, audited passport obstructions, and verified descent or regularity negatives can justify rejection, but only within the scope of their certificates. Once all retained candidates on a route have been rejected or deferred, the scheduler moves on (Tables 5 and 6).

Reference-run protocol. The ten reference runs use a Claude Opus 5.5 lead at medium effort, with strategy review disabled and no spend cap. All ten reach a verified polynomial (Table 2). Appendix G describes the endpoints and scoring details.

Table 2: **GALOIS reference runs**, two per task. *Rejected*: the three-point route excluded by an exact field-of-moduli test before the run finished the four-point route shown. Numbers are run 1 / run 2. *The polynomial is the $T = 1$ specialization of the run’s regular family $F(T, V)$. *Steps*, *Lead*, *Obstr.*: pipeline steps, lead-agent calls, and route-rejection-agent calls; *Min*: wall-clock minutes; K_r uses output tokens over pipeline steps.

Task	Rejected	Finished route	Steps	Lead	Obstr.	Min	USD	Tokens	K_r
M_{11}	(4A, 4A, 5A)	(2A, 2A, 3A, 4A)	46 / 45	27 / 24	10 / 10	78 / 78	26.40 / 22.33	128k / 119k	73.5 / 73.0
M_{12}	none	(3A, 3A, 6A)	13 / 13	6 / 9	6 / 5	13 / 10	6.23 / 7.27	38k / 38k	100 / 100
M_{22}	none	(2A, 11A, 11B)*	22 / 25	13 / 12	5 / 8	126 / 59	24.64 / 14.51	252k / 114k	100 / 100
M_{23}	none	(2A, 23A, 23B)	16 / 17	12 / 9	3 / 4	43 / 41	7.97 / 7.27	38k / 41k	100 / 100
M_{24}	(3A, 3A, 12B)	(2A, 2A, 2A, 12B)	37 / 31	16 / 11	8 / 11	79 / 43	15.34 / 12.69	88k / 74k	35.1 / 46.7

5 EXPERIMENTAL SETUP

Tasks and systems. We evaluate the five Mathieu groups M_{11} , M_{12} , M_{22} , M_{23} , and M_{24} , all Tier 1 tasks. The August 2026 realization of M_{23} (Huang et al., 2026) was excluded from the agents’ corpus (Appendix D). Each task has at least five checkpoints. Counting unordered class multisets

provides 4,752 three-point and 28,227 four-point class vectors, for 32,979 candidate routes in total. The ledger lists 13 correct routes, roughly one in 2,500. We test nine models: Claude Opus 5, GPT-6 Sol, GPT-6 Astra, Claude Fable 5.1, GPT-5.6 Sol, GPT-5.6 Terra, Claude Sonnet 5, DeepSeek V4 Pro, and GLM-5.3 Flash, all at effort *xhigh* except Fable, which runs at *max* in two later batches with the same budgets. A *system* means a model together with its harness and effort setting. Each model uses its native Claude or Codex harness, or an OpenRouter-backed adapter. All systems access the sandbox through the same metered, offline Model Context Protocol *bridge*, which records every call. Each system completes two main-campaign runs per task (Table 4; Appendix H describes the eligibility rules). Appendix H also documents each cohort’s configuration, software versions, and restrictions on stored answers.

Sandbox, budgets, verification. Each run uses a network-isolated workspace with Python, GAP, and Magma (Appendix F). The budget per task is 10 hours, 500,000 output tokens, eight CPUs, and a task-dependent memory limit. Hitting a budget limit ends the run, but finalized artifacts can still earn credit. Verifiers run afterwards, outside the agent’s budget. A checkpoint counts only when its verifier returns *verified*. Appendix F shows the verifier limits.

6 RESULTS AND ANALYSIS

IGP-Bench has the right level of difficulty. Mean progress scores range from 2.6 to 81.0 (Table 3; Figure 3). Four systems, GPT-5.6 Terra, Claude Sonnet 5, DeepSeek V4 Pro, and GLM-5.3 Flash, do some screening correctly (S between 8.7% and 49.7% of its maximum) but rarely identify a correct route (R at most 20%) and make almost no construction progress (C at most 2.5%); none of their 40 runs produces a verified polynomial. The other five systems make measurable progress on routes, each identifying a correct route in at least half of its runs, so their 50 runs can be used to compare how agents choose and pursue routes; the next three paragraphs use these runs (Appendix H.6). No system solves all five tasks, although all five are Tier 1 tasks with a published realization. Only 16 of the 90 runs produce a verified polynomial, 3 on M_{11} , 5 on M_{12} , and 8 on M_{22} . No run does so on M_{23} or M_{24} .

Table 3: **Main results on IGP-Bench.** Means over eligible task-runs. S , R , and C are percentages of their maxima, 30, 30, and 40; P is in raw points. POLY is the percentage of runs ending with a verified polynomial. U_r is the resource efficiency metric and K_r the knowing-when-to-stop metric, both computed from output tokens. GALOIS[†] summarizes separate reference runs under a different protocol, and its U_r is a step-based proxy (Appendix G). Table 11 provides intervals, and Appendix E defines the other metrics.

System	Verified progress					Allocation		Protocol		Tool use		
	Screen	Route	Constr.	Total	POLY	Eff.	Stop	Format	Fidel.	Valid	Term.	Runs
	S	R	C	P	(%)	U_r	K_r	I_f	η	(%)	(%)	n
Claude Opus 5	100.0	100.0	52.5	81.0	50.0	57.7	72.8	96.9	99.0	99.8	91.2	10
GPT-6 Sol	97.0	80.0	22.5	62.1	20.0	31.4	79.1	99.2	98.4	100.0	98.7	10
GPT-6 Astra	78.0	50.0	52.5	59.4	50.0	60.3	55.9	98.2	100.0	99.8	96.7	10
Claude Fable 5.1	94.7	70.0	32.5	62.4	30.0	72.2	92.7	95.6	97.5	100.0	91.1	10
GPT-5.6 Sol	80.7	50.0	12.5	44.2	10.0	27.5	62.9	98.4	98.0	99.9	94.2	10
GPT-5.6 Terra	49.7	20.0	2.5	21.9	0.0	24.6	38.2	96.3	99.7	99.9	94.0	10
Claude Sonnet 5	45.0	10.0	2.5	17.5	0.0	26.9	30.6	93.9	97.3	99.8	95.5	10
DeepSeek V4 Pro	22.7	0.0	2.5	7.8	0.0	34.4	66.6	86.5	88.3	99.6	88.6	10
GLM-5.3 Flash	8.7	0.0	0.0	2.6	0.0	33.5	63.9	53.1	75.0	97.2	82.0	10
GALOIS [†]	100.0	100.0	100.0	100.0	100.0	82.8	82.8	N/A	N/A	N/A	N/A	10

Going deeper does not always imply better score. Overall performance is the progress score $P = S + R + C$, with maxima $100 = 30 + 30 + 40$, and we computed that across the 50 runs of the five systems, the correlation between the construction score C , which measures how far an agent goes on a route, and the rest of the score, $S + R$, is only +0.08 (Spearman, 95% bootstrap CI $[-0.20, +0.35]$; Appendix H.6). As a concrete example, GPT-6 Astra matches Claude Opus 5

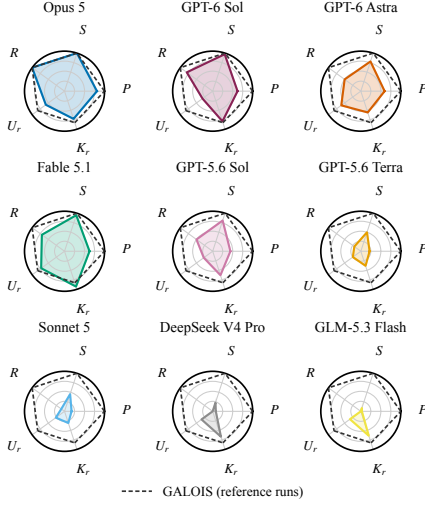


Figure 3: **Five-metric profiles of the nine systems** on a 0–100 scale. Dashed: GALOIS reference runs.

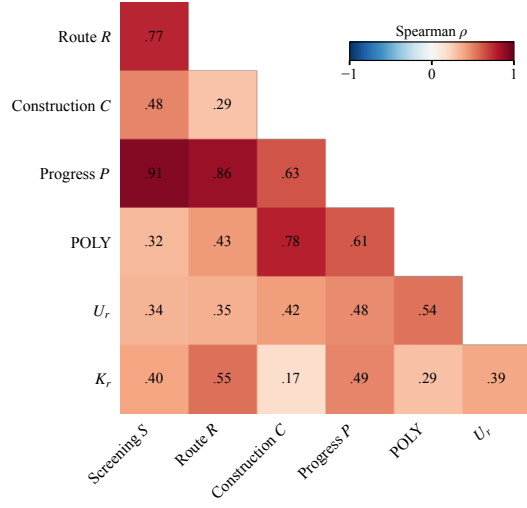


Figure 4: **Spearman correlations** over all 90 runs (K_r on 86); Appendix H.5 gives the full matrix.

on mean construction score ($C = 52.5$) and verified-polynomial rate (50%), yet ranks fourth overall with $P = 59.4$, compared with Opus’s leading score of 81.0 (Table 3); the 21.6-point gap comes entirely from screening (6.6 points) and route identification (15 points). Conversely, GPT-6 Sol and Astra have similar progress scores ($P = 62.1$ versus 59.4) but very different construction scores ($C = 22.5$ versus 52.5); Sol makes up the difference mainly through route identification, identifying a correct route in eight of ten runs against Astra’s five (Table 17).

Knowing when to stop and going deeper are largely distinct abilities. Over the same 50 runs, the correlation between the knowing-when-to-stop metric K_r and the construction score C is +0.01 (Spearman, 95% bootstrap CI $[-0.27, +0.30]$; +0.17 over all runs, Figure 4). Comparing runs within the same task gives +0.25 ($[-0.05, +0.51]$), so going deeper explains at most a small share of K_r (Appendix H.6). Opus and Astra, tied on construction, differ in K_r (72.8 versus 55.9; Figure 1).

Knowing when to stop is only weakly correlated with resource efficiency. Over the same 50 runs, the correlation between the knowing-when-to-stop metric K_r and the resource efficiency metric U_r is +0.29 (Spearman, 95% bootstrap CI $[-0.02, +0.56]$). A high K_r alone therefore does not show that an agent converted its recorded work into verified progress.

GALOIS knows when to stop. In every reference run on M_{11} and M_{24} , GALOIS starts with a three-point route, (4A, 4A, 5A) for M_{11} and (3A, 3A, 12B) for M_{24} , rejects it after exact field-of-moduli tests rule out its covers, and moves to a correct four-point route, (2A, 2A, 3A, 4A) and (2A, 2A, 2A, 12B) respectively, which leads to a certified polynomial (Table 2). On M_{11} the rejection is cheap in computation, under one minute of tool time over 6 steps, though it still takes about 27% of output tokens; the correct route then takes 36–37 steps. On M_{24} the rejection is the expensive part, with 12–18 steps, 53–65% of output tokens, and 52–63 minutes of tool time, after which the correct route needs 16 steps and about 15 minutes of tool time.

Tier 2 and 3 tasks of IGP-Bench also measure agents’ ability. We run the best-performing system, Claude Opus 5, once on each of ten tasks listed as open in the campaign build, with 2 h 48 min of effective agent time per task. Five of them are Tier 2 tasks, four are Tier 3 tasks, and one is Tier 1. It solves none of them, but it makes meaningful progress, even proving some candidate routes impossible, and averages $P = 45.1$ under the open-case profile (Table 8). A proved obstruction removes a route from future work.

7 DISCUSSION AND LIMITATIONS

Knowing when to stop and going deeper are largely distinct abilities. A research agent must decide both how to carry out the next calculation and whether that calculation is still worth doing. In our runs, going deep on a route is only weakly related to knowing when to stop and to the rest of the score. A capable model can keep overcoming local difficulties on a route that cannot reach a realization over $\mathbb{Q}$, such as a cover built over a larger number field (Section 3.3); under a fixed budget, that work displaces screening and viable construction.

Where computation is spent matters. Earlier work shows that more computation does not always mean better performance. Snell et al. (2024) find that allocating test-time computation according to problem difficulty is more efficient than spending it uniformly, and Cuadron et al. (2025) find that excessive internal reasoning, at the expense of interacting with the environment, is associated with lower completion rates in interactive software tasks. We see the same pattern in how agents choose research routes. Agents with higher construction scores are not clearly better at knowing when to stop (Section 6). In all three settings, the value of extra computation depends on where it is spent. IGP-Bench records what each computation costs and whether it produces a verified result, so it shows how much of the budget each route used. GALOIS builds on this: before investing more in a route, it runs a cheap test that can rule the route out, and it leaves the route when the test does.

Limitations. Our evaluation covers five tasks with 27 checkpoints. Each run may take up to ten hours, and the 90 runs used 540 agent-hours and 28.7 million output tokens, so evaluating more tasks or more runs per system was beyond our resources. With ten runs per system, the leading systems’ bootstrap intervals overlap, and the comparisons establish neither a causal effect of stopping nor that greater capability systematically lowers scores. GALOIS draws on 36 skills that give it expertise in the inverse Galois problem, so its reference runs are not a like-for-like comparison with the single agents. We use them as a reference for system design, and our longer-term aim is to apply GALOIS to Tier 3 tasks; a matched ablation of its route rejection is still needed (Appendix G).

In IGP-Bench, a route counts as correct only if the literature supports it or the run itself produces a verified polynomial on it. A working route unknown to the literature therefore counts as wrong unless the agent completes a polynomial on it, and identifying it earns no route credit. We expect such routes to be rare: these groups have been studied for decades, historical attempts suggest that generic routes seldom succeed, and only 13 of the 32,979 candidate routes are known to work (Section 5). K_r summarizes how costs are distributed across routes; it does not judge whether each stopping decision was correct.

8 CONCLUSION

We asked whether AI agents know when to stop pursuing a wrong route and, if so, how well they do it. IGP-Bench answers this question with verified intermediate results and a record of what each route costs, and its tasks are hard enough to separate systems: no system solves all five Mathieu tasks, and Tier 2 and 3 tasks still register partial progress. Knowing when to stop is largely distinct from going deep on a route and only weakly correlated with resource efficiency. We emphasized knowing when to stop and introduced GALOIS, a multi-agent system whose route-rejection agent actively searches for obstructions and steers the system toward correct routes, realizing all five tasks, and we expect it to extend to the open Tier 3 cases.

REPRODUCIBILITY STATEMENT

Upon publication, we will release the 200 tasks and checkpoint definitions, GAP and Magma verifiers, route ledger, scoring profiles, evaluation and metric code, agent manual, task briefs, MCP benchmark bridge, and full GALOIS source. The fixed corpus consists of cited, published documents, which we redistribute only where their licenses permit. Section 5 and Appendices F–H describe the software requirements, sandbox, budgets, system settings, and run eligibility decisions. Appendices C–G document the M_{23} checkpoint ladder, corpus, and GALOIS mechanisms, with references to the source files. We will also publish the preserved run directories, including artifact graphs, checkpoint submissions, tool logs, and abandoned artifacts. If a system produces a verified realization of a Tier 3 group, which would be a new mathematical result, we will disclose the system, model, and configuration that produced it. Reproduction has two practical limits: Magma-based verifiers require a Magma license, and frontier-model outputs are nondeterministic and may change with provider updates. Agent trajectories can therefore be reproduced only in distribution, not exactly.

AI USE STATEMENT

We used generative AI to help draft and revise the text, write $\text{\LaTeX}$, develop scripts for processing run data and generating tables and figures, search the literature, and check the manuscript against the released data and code. We did not use it to originate research ideas, generate experimental results, or assign scores. The evaluated agents are LLM systems, and their behavior is the subject of the experiments. Deterministic verifiers and evaluation code compute all benchmark scores. The description-fidelity judge in Section 3.4 does not contribute to any score, and the LLM-assisted failure taxonomy is used only for qualitative analysis. The authors reviewed the AI-assisted work, checked every reported number against the released tables and data, checked mathematical statements against the cited sources, and ran generated code and inspected its outputs. We take responsibility for the final text, claims, and artifacts, including all AI-assisted content.

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*, 2026. arXiv:2602.05192.
- Tudor Achim, Alex Best, Alberto Bietti, Kevin Der, Mathis Fédérico, Sergei Gukov, Daniel Halpern-Leistner, Kirsten Henningsgard, Yury Kudryashov, Alexander Meiburg, et al. Aristotle: IMO-level automated theorem proving. *arXiv preprint*, 2025. arXiv:2510.01346.
- American Institute of Mathematics. AI and number theory. Workshop, 27–30 May 2026, <https://aimath.org/pastworkshops/aint26.html>, 2026. Accessed 2 October 2026.
- Chenyang An, Qihao Ye, Minghao Pan, and Jiayaun Zhang. QED: An open-source multi-agent system for generating mathematical proofs on open problems. *arXiv preprint*, 2026. arXiv:2604.24021.
- Benjamin Antieau. J"urgen kl"uners and gunter malle on the inverse galois problem and computation. Interview, <https://antieau.github.io/2026/08/27/gunter-malle-juergen-klueners-ai.html>, 2026. Published 27 August 2026; accessed 2 October 2026.
- Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Math-Arena: Evaluating LLMs on uncontaminated math competitions. *arXiv preprint*, 2025. arXiv:2505.23281.
- Gennadii V. Belyi. On Galois extensions of a maximal cyclotomic field. *Mathematics of the USSR-Izvestiya*, 14(2):247–256, 1980.
- Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, 2025.

- François Charton, Jordan S. Ellenberg, Adam Zsolt Wagner, and Geordie Williamson. PatternBoost: Constructions in mathematics with a little help from AI. *arXiv preprint*, 2024. arXiv:2411.00566.
- Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, et al. Seed-Prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint*, 2025a. arXiv:2507.23726.
- Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint*, 2023. arXiv:2308.10848.
- Xiaoyang Chen and Xiang Jiang. Moonshine: An autonomous mathematical research agent centered on conjecture generation. *arXiv preprint*, 2026. arXiv:2606.10806.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for $2+3=?$ on the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b. arXiv:2412.21187.
- Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, 2025c.
- Lyndie Chiou. Mathematicians use AI to find mysterious symmetries, solving decades-old problem. *Scientific American*, September 2026. 22 September 2026, <https://www.scientificamerican.com/article/mathematicians-use-ai-to-find-mysterious-symmetries-solving-decades-old-problem/>.
- Kevin Coombes and David Harbater. Hurwitz families and arithmetic Galois groups. *Duke Mathematical Journal*, 52(4):821–839, 1985.
- Jean-Marc Couveignes. Tools for the computation of families of coverings. In *Aspects of Galois Theory*, volume 256 of *London Mathematical Society Lecture Note Series*, pp. 38–65. Cambridge University Press, 1999.
- Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. In *Conference on Language Modeling*, 2025. arXiv:2502.08235.
- Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, Marc Lackenby, Geordie Williamson, Demis Hassabis, and Pushmeet Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021.
- Pierre Dèbes and Jean-Claude Douai. Algebraic covers: field of moduli versus field of definition. *Annales Scientifiques de l'École Normale Supérieure*, 30(3):303–338, 1997.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint*, 2023. arXiv:2305.14325.
- Epoch AI. The inverse Galois problem for the Mathieu group M_{23} . FrontierMath: Open Problems, <https://epoch.ai/frontiermath/open-problems/inverse-galois>, 2026. Listed as unsolved by January 2026; marked “solved (human + AI)” as of 25 September 2026.
- Moran Feldman and Amin Karbasi. Gödel test: Can large language models solve easy conjectures? *arXiv preprint*, 2025. arXiv:2509.18383.
- Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang-hyun Kim, Federico Pasqualotto, et al. Towards autonomous mathematics research. *arXiv preprint*, 2026. arXiv:2602.10177.

- Michael D. Fried. Fields of definition of function fields and Hurwitz families—groups as Galois groups. *Communications in Algebra*, 5(1):17–82, 1977.
- Michael D. Fried and Helmut Völklein. The inverse Galois problem and rational points on moduli spaces. *Mathematische Annalen*, 290:771–800, 1991.
- Suhaas Garre, Erik Knutsen, Sushant Mehta, and Edwin Chen. Riemann-Bench: A benchmark for moonshot mathematics. *arXiv preprint*, 2026. arXiv:2604.06802.
- Aryo Pradipta Gema, Alexander Hägele, Runjin Chen, Andy Ardit, Jacob Goldman-Wetzler, Kit Fraser-Taliente, Henry Sleight, Linda Petrini, Julian Michael, Beatrice Alex, Pasquale Minervini, Yanda Chen, Joe Benton, and Ethan Perez. Inverse scaling in test-time compute. *arXiv preprint*, 2025. arXiv:2507.14417.
- Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv preprint*, 2025. arXiv:2511.02864.
- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järvinen, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI. *arXiv preprint*, 2024. arXiv:2411.04872.
- 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. *arXiv preprint*, 2025. arXiv:2502.18864.
- Louis Granboulan. Construction d’une extension régulière de $\mathbb{Q}(T)$ de groupe de Galois M_{24} . *Experimental Mathematics*, 5(1):3–14, 1996.
- Ryan Greenblatt, Ajeya Cotra, and Hjalmar Wijk. Brief independent investigation of agents’ behavior, reasoning and collaboration in the OpenAI/Hugging Face hacking incident. METR, <https://metr.org/blog/2026-08-26-openai-hugging-face-incident-investigation/>, 2026. Published 26 August 2026.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.
- Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MAgentBench: Evaluating language agents on machine learning experimentation. *arXiv preprint*, 2023. arXiv:2310.03302.
- Xiaoyu Huang, Blake Jackson, Kyu-Hwan Lee, Bjorn Poonen, Rachel Pries, and Shaowu Zhang. The Mathieu group M_{23} is a Galois group over $\mathbb{Q}$. *arXiv preprint*, 2026. arXiv:2608.08538.
- Thomas Hubert, Rishi Mehta, Laurent Sartran, et al. Olympiad-level formal mathematical reasoning with reinforcement learning. *Nature*, 2025. doi: 10.1038/s41586-025-09833-y.
- Hugging Face. Anatomy of a frontier lab agent intrusion: A technical timeline of the july 2026 incident. <https://huggingface.co/blog/agent-intrusion-technical-timeline>, 2026. Published 27 July 2026.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations*, 2024.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint*, 2022. arXiv:2207.05221.

- Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025. arXiv:2407.01502.
- Michael Klug, Michael Musty, Sam Schiavone, and John Voight. Numerical calculation of three-point branched covers of the projective line. *LMS Journal of Computation and Mathematics*, 17(1):379–430, 2014.
- Patrick Tser Jern Kon, Qiuyi Ding, Jiachen Liu, Xinyi Zhu, Jingjia Peng, Jiarong Xing, Yibo Huang, Yiming Qiu, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, Matei Zaharia, and Ang Chen. EXP-Bench: Can AI conduct AI research experiments? In *International Conference on Learning Representations*, 2026.
- Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Dietrich, Elizabeth Hasin, Josh Isaacs, Grant Jones, et al. Measuring AI ability to complete long tasks. *arXiv preprint*, 2025a. arXiv:2503.14499.
- Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, et al. Measuring AI ability to complete long software tasks. *arXiv preprint*, 2025b. arXiv:2503.14499.
- Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for “mind” exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Yong Lin, Shange Tang, Bohan Lyu, Ziran Yang, Jui-Hui Chung, Haoyu Zhao, Lai Jiang, Yihan Geng, Jiawei Ge, Jingruo Sun, et al. Goedel-Prover-V2: Scaling formal theorem proving with scaffolded data synthesis and self-correction. *arXiv preprint*, 2025. arXiv:2508.03613.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations*, 2024.
- Xun Liu, Yi Evie Zhang, Vira Kasprova, Parisa Rabbani, Pardis Sadat Zahraei, Tianyu Zhang, Ali Ebrahimpour-Borojeny, and Varun Chandrasekaran. AgentAbstain: Do LLM agents know when not to act? *arXiv preprint*, 2026. arXiv:2607.10059.
- 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*, 2024. arXiv:2408.06292.
- Han Luo, Bingbing Wen, and Lucy Lu Wang. Agentic abstention: Do agents know when to stop instead of act? *arXiv preprint*, 2026. arXiv:2606.28733.
- Gunter Malle. Polynomials with Galois groups $\text{Aut}(M_{22})$, M_{22} , and $\text{PSL}_3(\mathbb{F}_4) \cdot 2_2$ over $\mathbb{Q}$. *Mathematics of Computation*, 51(184):761–768, 1988.
- Gunter Malle and B. Heinrich Matzat. *Inverse Galois Theory*. Springer Monographs in Mathematics. Springer, 2nd edition, 2018.
- B. Heinrich Matzat. *Konstruktive Galoistheorie*, volume 1284 of *Lecture Notes in Mathematics*. Springer, 1987.
- B. Heinrich Matzat and Andreas Zeh-Marschke. Realisierung der Mathieugruppen M_{11} und M_{12} als Galoisgruppen über $\mathbb{Q}$. *Journal of Number Theory*, 23(2):195–202, 1986.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint*, 2025. arXiv:2501.19393.
- Fan Nie, Ken Ziyu Liu, Zihao Wang, Rui Sun, Wei Liu, Weijia Shi, Huaxiu Yao, Linjun Zhang, Andrew Y. Ng, James Zou, et al. UQ: Assessing language models on unsolved questions. *arXiv preprint*, 2025. arXiv:2508.17580.

- Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint*, 2025. arXiv:2506.13131.
- OpenAI. The hugging face incident and the road ahead. <https://openai.com/index/hugging-face-incident-and-the-road-ahead/>, 2026a. Published 26 August 2026.
- OpenAI. Safety and alignment in an era of long-horizon models. <https://openai.com/index/safety-alignment-long-horizon-models/>, 2026b. Published 20 July 2026.
- QuocViet Pham, Elvir Karimov, Andrey Galichin, and Ivan Oseledets. TheoremBench: Evaluating LLMs on theorem proving in formal mathematics. *arXiv preprint*, 2026. arXiv:2606.09450.
- Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity’s last exam. *arXiv preprint*, 2025. arXiv:2501.14249.
- Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. ChatDev: Communicative agents for software development. *arXiv preprint*, 2023. arXiv:2307.07924.
- Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanxia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, et al. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint*, 2025. arXiv:2504.21801.
- Matthieu Romagny and Stefan Wewers. Hurwitz spaces. In *Groupes de Galois arithmétiques et différentiels*, volume 13 of *Séminaires et Congrès*, pp. 313–341. Société Mathématique de France, 2006.
- Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625:468–475, 2024.
- SAIR Foundation. Inverse galois problem (IGP24). <https://competition.sair.foundation/competitions/igp24/overview>, 2026. Accessed 2 October 2026.
- Johannes Schmitt, Gergely Bérczi, Jasper Dekoninck, Jeremy Feusi, Tim Gehringer, Raphael Apenzeller, Jim Bryan, Niklas Canova, Timo de Wolff, Filippo Gaia, Michel van Garrel, Baran Hashemi, David Holmes, Aitor Iribar Lopez, Victor Jaeck, Martina Jørgensen, Steven Kelk, Stefan Kuhlmann, Adam Kurpisz, Chiara Meroni, Ingmar Metzler, Martin Möller, Samuel Muñoz-Echániz, Robert Nowak, Georg Oberdieck, Daniel Platt, Dylan Possamaï, Gabriel Ribeiro, Raúl Sánchez Galán, Zheming Sun, Josef Teichmann, Richard P. Thomas, and Charles Vial. IMProofBench: Benchmarking AI on research-level mathematical proof generation. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025. arXiv:2509.26076.
- Jean-Pierre Serre. *Topics in Galois Theory*, volume 1 of *Research Notes in Mathematics*. Jones and Bartlett, 1992.
- Zhihong Shao, Yuxiang Luo, Chengda Lu, Z. Z. Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. DeepSeekMath-V2: Towards self-verifiable mathematical reasoning. *arXiv preprint*, 2025. arXiv:2511.22570.
- Zachary S. Siegel, Sayash Kapoor, Nitya Nadgir, Benedikt Stroebl, and Arvind Narayanan. CORE-Bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. *arXiv preprint*, 2024. arXiv:2409.11363.

- Jeroen Sijsling and John Voight. On computing Belyi maps. *Publications Mathématiques de Besançon. Algèbre et Théorie des Nombres*, pp. 73–131, 2014.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint*, 2024. arXiv:2408.03314.
- Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI’s ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025.
- Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025. arXiv:2503.16419.
- The LMFDB Collaboration. The L-functions and modular forms database. <https://www.lmfdb.org>, 2026. Number-field tables by transitive Galois group, queried 25 September 2026.
- John G. Thompson. Some finite groups which appear as $\text{Gal } L/K$, where $K \subseteq \mathbb{Q}(\mu_n)$. *Journal of Algebra*, 89(2):437–499, 1984.
- Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625:476–482, 2024.
- George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amityay Thakur, and Swarat Chaudhuri. PutnamBench: Evaluating neural theorem-provers on the Putnam mathematical competition. *arXiv preprint*, 2024. arXiv:2407.11214.
- Helmut Völklein. *Groups as Galois Groups: An Introduction*, volume 53 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1996.
- Adam Zsolt Wagner. Constructions in combinatorics via neural networks. *arXiv preprint*, 2021. arXiv:2104.14516.
- Erik Y. Wang, Sumeet Motwani, James V. Roggeveen, Eliot Hodges, Dulhan Jayalath, Charles London, Kalyan Ramakrishnan, Flaviu Cipcigan, Philip Torr, and Alessandro Abate. Measuring progress in reasoning toward mathematical discovery with automatic verification. *arXiv preprint*, 2026. arXiv:2603.15617.
- Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, et al. Kimina-Prover preview: Towards large formal reasoning models with reinforcement learning. *arXiv preprint*, 2025a. arXiv:2504.11354.
- Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like LLMs. *arXiv preprint*, 2025b. arXiv:2501.18585.
- Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *arXiv preprint*, 2024. arXiv:2407.18418.
- Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, et al. RE-Bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. *arXiv preprint*, 2024. arXiv:2411.15114.
- Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint*, 2023. arXiv:2308.08155.

- Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint*, 2025. arXiv:2504.08066.
- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint*, 2025. arXiv:2504.15895.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint*, 2024. arXiv:2406.12045.
- Jie Zhang, Cezara Petrucci, Kristina Nikolić, and Florian Tramèr. RealMath: A continuous benchmark for evaluating language models on research-level mathematics. *arXiv preprint*, 2025. arXiv:2505.12575.
- Shaowu Zhang. Where to point the telescope: AI, mathematical judgment, and M_{23} . Proofs and Prompts, <https://proofsandprompts.com/2026/08/21/where-to-point-the-telescope/>, 2026. Published 21 August 2026.
- Daniel Zheng, Ingrid von Glehn, Yori Zwols, Iuliya Beloshapka, Lars Buesing, Daniel M. Roy, Martin Wattenberg, Bogdan Georgiev, Tatiana Schmidt, Andrew Cowie, et al. AI co-mathematician: Accelerating mathematicians with agentic AI. *arXiv preprint*, 2026. arXiv:2605.06651.
- 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.
- David Zywina. Inverse Galois problem for small simple groups. Note, <https://pi.math.cornell.edu/~zywina/papers/smallGalois.pdf>, 2025. Last updated August 2025.

A ADDITIONAL RELATED WORK

Mathematics benchmarks. Mathematics benchmarks range from competition and formal problems, such as miniF2F, PutnamBench, and MathArena (Zheng et al., 2022; Tsoukalas et al., 2024; Balunović et al., 2025), to expert- and research-level questions, such as Humanity’s Last Exam, RealMath, First Proof, UQ, and the Gödel Test (Phan et al., 2025; Zhang et al., 2025; Abouzaid et al., 2026; Nie et al., 2025; Feldman & Karbasi, 2025).

AI systems for mathematics. Machine learning has guided mathematical intuition (Davies et al., 2021) and found new constructions through search, as in FunSearch, AlphaEvolve, and Pattern-Boost (Wagner, 2021; Romera-Paredes et al., 2024; Novikov et al., 2025; Georgiev et al., 2025; Charton et al., 2024). AlphaGeometry, AlphaProof, Aristotle, Seed-Prover, DeepSeek-Prover-V2, Kimina-Prover, Goedel-Prover-V2, and DeepSeekMath-V2 target competition-level and formal theorem proving (Trinh et al., 2024; Hubert et al., 2025; Achim et al., 2025; Chen et al., 2025a; Ren et al., 2025; Wang et al., 2025a; Lin et al., 2025; Shao et al., 2025). Research agents such as Aletheia, the AI co-mathematician, QED, and Moonshine work on research-level mathematics (Feng et al., 2026; Zheng et al., 2026; An et al., 2026; Chen & Jiang, 2026).

Research agents and multi-agent systems. MLE-bench, RE-Bench, MAgentBench, ScienceAgentBench, CORE-Bench, and SWE-bench evaluate agents on engineering and scientific tasks (Chan et al., 2025; Wijk et al., 2024; Huang et al., 2023; Chen et al., 2025c; Siegel et al., 2024; Jimenez et al., 2024). AgentBench and τ -bench on general tool use (Liu et al., 2024; Yao et al., 2024), and Kwa et al. (2025b) measure the length of tasks agents can complete. The AI Scientist, the AI Scientist-v2, and the AI co-scientist automate parts of the research process (Lu et al., 2024; Yamada et al., 2025; Gottweis et al., 2025). ChatDev and AgentVerse assign agents different roles, and multi-agent debate lets agents critique one another (Qian et al., 2023; Chen et al., 2023; Du et al., 2023).

Reasoning length and abstention. Reasoning models can also underthink (Wang et al., 2025b), and budget forcing and early exit control how long a model reasons (Muennighoff et al., 2025; Yang et al., 2025). Language models partly know what they know (Kadavath et al., 2022), and abstention has been studied for language models and agents (Wen et al., 2024; Luo et al., 2026).

B THE CONSTRUCTIVE PIPELINE

B.1 MATHEMATICAL BACKGROUND

Regular realizations. Fix a finite group G . Almost every known realization of a nonabelian simple group over $\mathbb{Q}$ starts from a *regular* realization over $\mathbb{Q}(T)$. This is a polynomial $F(T, V) \in \mathbb{Q}[T, V]$ with $\text{Gal}(F/\mathbb{Q}(T)) \cong G$ such that $\mathbb{Q}$ is algebraically closed in its splitting field. Hilbert irreducibility then yields specializations over $\mathbb{Q}$ with the same Galois group (Serre, 1992; Völklein, 1996; Malle & Matzat, 2018).

Covers and route screening. Constructing such an F is a geometric problem. By Riemann’s existence theorem, a G -cover of the line branched over r points corresponds to a *generating tuple* $(g_1, \dots, g_r)$ with $g_1 \cdots g_r = 1$ and $\langle g_1, \dots, g_r \rangle = G$, up to simultaneous conjugation. Fixing the conjugacy classes $C = (C_1, \dots, C_r)$ of the g_i yields a *Nielsen class* $\text{Ni}(G; C)$. The braid group acts on this class, and its orbits are the components of the Hurwitz space parametrizing the covers (Fried, 1977; Fried & Völklein, 1991; Romagny & Wewers, 2006).

A G -cover whose branch divisor is defined over $\mathbb{Q}$ can be defined over $\mathbb{Q}$ only if two conditions hold: its class vector is stable under the absolute Galois group acting on classes and branch points together, and its braid orbit is Galois-invariant. At the real place, the orbit must be fixed by complex conjugation. Rational classes over rational branch points yield the simplest case. For M_{23} , the classes $23A$, $23B$ and the branch points $\pm\sqrt{-23}$ are swapped together by $\text{Gal}(\mathbb{Q}(\sqrt{-23})/\mathbb{Q})$. Group-theoretic data can therefore test necessary conditions early. They do not, in general, settle sufficiency, as the M_{23} fixed point illustrates.

Rigidity and explicit construction. Suppose G has trivial center, $r = 3$, the classes are rational, and $|\text{Ni}(G; C)| = 1$. The class vector is then *rigid*, which proves the existence of a regular realization over $\mathbb{Q}$ without computing its equations. This criterion, due to Belyi (1980), Thompson (1984), and Matzat (1987), together with its braid-orbit generalizations, settled 25 of the 26 sporadic groups. An explicit polynomial still requires constructing the cover. Such constructions are known for M_{11} , M_{12} , M_{22} , and M_{24} (Matzat & Zeh-Marschke, 1986; Malle, 1988; Granboulan, 1996). The modern approach uses numerical uniformization, recognition of exact equations, descent to $\mathbb{Q}$, and specialization (Couveignes, 1999; Klug et al., 2014; Sijtsling & Voight, 2014). The early screens in the pipeline below eliminate most class vectors.

B.2 PIPELINE STAGES

Figure 5 shows the twelve construction stages followed by GALOIS and the IGP-Bench checkpoints along the way.

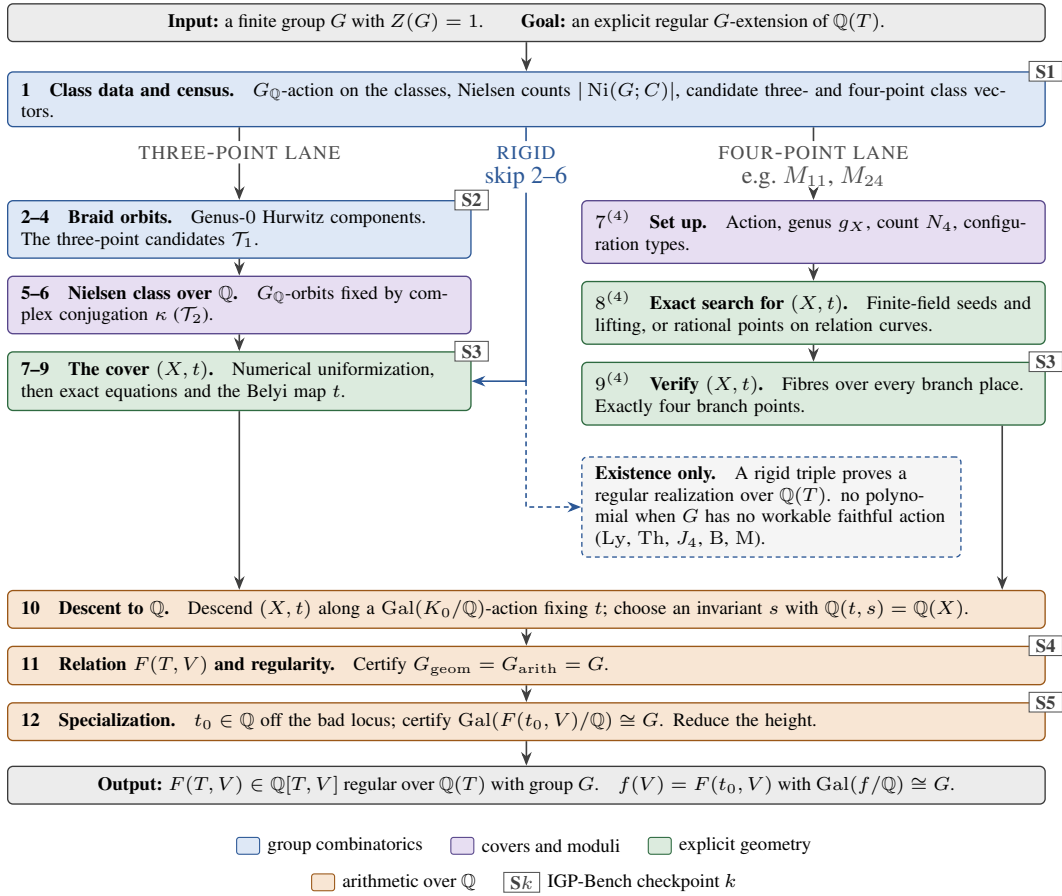


Figure 5: The construction pipeline used by GALOIS and checked by IGP-Bench. Stage 1 is shared by all groups. The process then splits according to the number of branch points and rejoins at Stage 10, where both paths use the same descent to $\mathbb{Q}$ and certification steps. Only Stage 7 is numerical. A rigid triple ($|\text{Ni}(G; C)| = 1$) skips Stages 2–6. If the group has no faithful action of workable degree, this proves existence but does not produce a polynomial. White tags mark the five generic IGP-Bench checkpoints: S1 generating tuple, S2 braid orbit, S3 geometric cover, S4 regular family over $\mathbb{Q}(T)$, and S5 final polynomial over $\mathbb{Q}$.

C THE M_{23} LADDER AND WORKED ROUTE

The task-specific ladder. The M_{23} task uses seven checkpoints rather than the usual five. For deepest verified stage $0, \dots, 7$, the cumulative construction score C is $0, 0, 5, 15, 20, 25, 30, 40$ points, respectively: (1) a generating tuple of degree-23 permutations with product one and a valid genus (S1). (2) a complete list of pairwise inequivalent Nielsen representatives for an ordered class vector, enumerated in GAP under a cap of 2×10^6 (S2, 5 points). (3) a geometric cover, either a $\mathbb{Q}$ -equation with branch cycles or a number-field cover with marking data (S3, 15). (4) a certificate that the field of moduli is $\mathbb{Q}$ (S4, 20). (5) a descended cover over $\mathbb{Q}$ with its descent isomorphisms (S4, 25). (6) a regular family over $\mathbb{Q}(T)$, verified using exact branch data and a good finite-characteristic reduction with group M_{23} . The verifier avoids Magma’s generic geometric-Galois-group computation in degree 23 because it does not finish in practical time (S4, 30). (7) a degree-23 polynomial, irreducible with square discriminant and Galois group M_{23} certified with Magma’s `GaloisProof`, or a specialization certificate for a verified family (S5, 40). On M_{23} , the route-identification score R is the *key case*: the agent earns $R = 30$ by verifying a braid orbit, Nielsen count, or Nielsen-representative result for $(2A, 23A, 23B)$. Stage 2 credit, however, is available for a complete representative list on *any* class vector. Several systems therefore earn $C = 5$ on M_{23} without R (Table 12). The screening score rewards verified facts in six categories: finite counts and structure for three- and four-point vectors, rationality, and arithmetic. Weights range from 1 for product one to 6 for a reduced Hurwitz component, a Hurwitz model, or an existence theorem. The total is capped at 30.

The worked route. The correct route uses the four-point class vector $(23A, 23A, 23B, 23B)$ or its three-point degenerations $(2A, 23A, 23B)$. A solver must complete the following steps.

1. **Find candidates.** M_{23} has 16 nontrivial conjugacy classes, giving $\binom{18}{3} = 816$ class triples up to order. No Galois-stable triple has a Nielsen class of size one, so rigidity does not apply. The smallest positive Nielsen count is 7, for $\{2A, 23A, 23B\}$ (Huang et al., 2026). The benchmark’s negative-case census records routes previously tried in the literature, including $(2A, 8A, 8A)$.
2. **Compute braid orbits.** Start from seeds (a, a, b, aba^{-1}) with $a \in 23A$, $b \in 23B$, and $ab \in 2A$. Apply the pure braid generators $\beta_{12}, \beta_{13}, \beta_{14}$ in GAP until each orbit closes. The seven seeds give seven orbits in the four-point Nielsen class.
3. **Compute genus and cusps.** View each orbit Z_i as a degree- $|Z_i|$ cover of the λ -line branched at $0, 1, \infty$. The generators have cycle counts $(c_1, c_2, c_3) = (2, 1, 1)$. Riemann–Hurwitz gives $g = 1 + \frac{1}{2}(|Z_i| - c_1 - c_2 - c_3) = 1 + \frac{1}{2}(2 - 4) = 0$. The cusps record the degenerate triples $(23A, 23B, 23B)$ and $(2A, 23A, 23B)$ obtained when branch points collide.
4. **Verify route identification.** Check that every orbit representative has the form (a, a, b, aba^{-1}) with $\langle a, b \rangle = M_{23}$, that $\{2A, 23A, 23B\}$ is Galois-stable, and that the number of verified orbits equals $|\text{Ni}(M_{23}; 2A, 23A, 23B)| = 7$. These checks establish the route-identification score R .
5. **Choose branch points over $\mathbb{Q}$.** The classes $23A$ and $23B$ are swapped by $\text{Gal}(\mathbb{Q}(\sqrt{-23})/\mathbb{Q})$. Choose the branch locus as the zero set of $x(x^2 + 23)$, with $0 \mapsto 2A$ and $\pm\sqrt{-23} \mapsto 23A, 23B$. The class map is then Galois-equivariant; it would not be with $\{0, 1, \infty\}$.
6. **Determine the Galois action on the covers.** The absolute Galois group acts on the seven covers. Although a transitive action might be expected, the action has a fixed point. This cover has field of moduli $\mathbb{Q}$ and descends to a cover over $\mathbb{Q}$ by the results of Coombes & Harbater (1985) and Dèbes & Douai (1997) (Huang et al., 2026). Complex conjugation acts by $[(g_1, g_2, g_3)] \mapsto [(g_1^{-1}, g_3^{-1}, g_2^{-1})]$.
7. **Build an analytic model.** The curve X is non-hyperelliptic of genus 4. A uniformization $X_{\mathbb{C}} \cong \Gamma \backslash \mathbb{D}$ gives four holomorphic differentials as power series at a suitable point.
8. **Recover exact equations.** The four series define the canonical map to $\mathbb{P}^3$. Its image is cut out by a quadric P and a cubic Q with coefficients in $\mathbb{Q}(\sqrt{-23})$, which must be recognized from numerical data and checked exactly. This is where the precision failure in Section 1 occurred. In triangle-group coordinates, recognition failed at 40, 60, and 90 digits. After switching to Galois-equivariant coordinates defined through the Belyi map, 40 digits sufficed for P . A separate reduction algorithm recovered Q from the same 40-digit data (Zhang, 2026).

9. **Find the cover.** Recover the degree-23 map $t: X \rightarrow \mathbb{P}^1$ from the points above the branch locus.
10. **Descend the cover.** Rewrite the cover over $\mathbb{Q}$ and choose a second function s with $\mathbb{Q}(X) = \mathbb{Q}(t, s)$.
11. **Find the relation and prove regularity.** Compute a polynomial $F(T, V) \in \mathbb{Z}[T, V]$ of degree 8 in T and 23 in V with $F(t, s) = 0$. Certify irreducibility, branch behavior, and $G_{\text{geom}} = G_{\text{arith}} = M_{23}$, using a reduction modulo 31 checked in Magma for the last equality.
12. **Specialize.** Use rational values t_0 off the bad locus to obtain $f(V) = F(t_0, V)$ with $\text{Gal}(f/\mathbb{Q}) \cong M_{23}$. Reduce the polynomial with `polredabs` and certify it in Magma. The published example is unramified outside $\{2, 3, 23\}$.

In this evaluation, systems that identify the route complete steps 1–4 and parts of 5–6. None completes step 8.

D LITERATURE CORPUS

The offline corpus (`igpbench-general-math-v4`) contains 17 documents and is searched through a deterministic offline BM25 index. It provides background rather than answers: textbooks, course notes, and expository articles on Galois theory, Hurwitz spaces, and the relevant groups, plus three evaluator-written guides.

We manually check every document before admitting it. It must contain no final polynomial, coefficient vector, family, or specialization for an active task; no task-specific branch-cycle tuple, Belyi map, or calibration fixture; no worked construction that can be copied by transcription; and no link or appendix containing such data. We explicitly exclude the M_{11} polynomial papers and König’s M_{11} family, the M_{23} realization (Huang et al., 2026) and its supplements, the $\text{SL}_2(13)$ and $\text{SL}_2(19)$ constructions, the “Rubik’s cube as a Galois group” preprint, and any A_5 polynomial. One section of Crespo’s article is redacted for this reason. An audit script also checks for known title and phrase markers from excluded sources.

1. J. S. Milne, *Fields and Galois Theory*, course notes, v5.10 (2022).
2. J. S. Milne, *Group Theory*, course notes, v4.01 (2025).
3. K. Conrad, *Galois Correspondence*, expository note.
4. *Outline of Galois Theory Development*, Stanford Math 210B course handout.
5. L. Chariker, *The Inverse Galois Problem, Hilbertian Fields, and Hilbert’s Irreducibility Theorem*, REU paper (2007).
6. M. D. Fried, *Nielsen Classes and Hurwitz Spaces*, web notes.
7. The Stacks Project Authors, *Algebraic Curves* (chapter 0BRV), 2026 snapshot.
8. R. A. Wilson, *Sporadic Groups*, chapter 5 of the finite simple groups course notes.
9. P. M. Bradley, *Counting G -Orbits on the Induced Action on k -Subsets*, PhD thesis, Manchester (2014).
10. T. Crespo, *Galois Representations, Embedding Problems and Modular Forms*, Collectanea Math. 48 (1997), with the example polynomials of Section 3.2 redacted.
11. M. Kuschowitz, *Quadratische Formen, Einbettungsprobleme und die Hasse–Witt-Invariante*, Master’s thesis, TU Braunschweig (2010).
12. D. Sen, K. Bauer, P. Zvengrowski, *A Generalized Goursat Lemma*, J. Algebra Appl. 12 (2013).
13. R. Fernando, *Structure and Generation Properties of the Rubik’s Cube Group*, expository notes (2017).
14. The GAP Group, *GAP Tutorial*, release 4.16.1 (2026).
15. IGP-Bench, *Magma, GAP, and Polynomial-Galois Workflow Guide*, evaluator documentation.
16. IGP-Bench, *Composita, Central Lifts, and Discriminant Square Classes*, evaluator documentation.
17. IGP-Bench, *IGPBench Task Literature Map*, an agent-searchable reading map.

E METRIC DEFINITIONS

Checkpoint sequences. The generic ladder has five stages: generating tuple (S1), braid orbit (S2), geometric cover (S3), regular family over $\mathbb{Q}(T)$ (S4), and final polynomial (S5). Six tasks use ladders adapted to known constructions: M_{11} , M_{23} , $\text{SL}_2(13)$, $\text{SL}_2(19)$, and two Rubik’s cube groups. The M_{11} campaign runs were scored on the generic ladder. Appendix C gives the seven-stage M_{23} ladder. At every checkpoint, GAP or Magma checks the required properties; the submitted object need not match a stored reference artifact.

Research and construction credit. The screening score S rewards independently verified facts about tuples, orbits, families, and polynomials. Each fact earns credit once, up to a total of 30. On the generic ladder, the weights range from 0.5 for statistical evidence to 3 for an existence theorem. Task-specific schedules can use different weights, as the M_{23} schedule in Appendix C does. The construction score C depends on the deepest verified checkpoint. The generic ladder gives 15 points for a cover, 30 for a regular family, and 40 for a final polynomial; task-specific ladders distribute intermediate credit differently. A verified polynomial earns $C = 40$, whatever method produced it.

Route ledger and milestone. The ledger lists 141 distinct literature routes for 64 groups, along with known obstructions. A correct route must support a realization over $\mathbb{Q}(T)$ for the target group itself. For example, a route in $\text{Aut}(M_{22})$ is not correct for M_{22} . On the generic ladder, earning the route-identification score $R = 30$ requires a single artifact containing both a generating tuple on a registered correct route and a verified braid-orbit computation on that route. The score is awarded at most once. Some tasks have their own acceptance rules. For M_{23} , Appendix C lists the accepted braid-orbit, Nielsen-count, and Nielsen-representative certificates.

Resource usage. We record the model and reasoning settings, input tokens with cache accounting where available, output tokens, elapsed benchmark time, and measured tool CPU hours. We include separately reported reasoning-token counts when available, but do not count them twice when they are already included in output tokens. Tool CPU usage excludes inference on the model server. A missing measurement is not treated as zero usage.

Resource efficiency. Let A be the finalized artifacts, $r(v)$ the cost of artifact v , and $U \subseteq A$ the credited artifacts together with all their ancestors in the dependency graph. Then

$$U_r = 100 \frac{\sum_{v \in U} r(v)}{\sum_{v \in A} r(v)}.$$

We compute U_r and K_r from output tokens by default, and also from tool CPU time and accumulated worker time; with parallel work, the latter differs from elapsed wall time.

Format compliance. A deterministic formatter checks artifact records and final submissions against public formatting requirements, allowing at most two repair reminders. We define

$$I_f = \frac{100 n_0 + 50 n_1 + \frac{100}{3} n_2}{n_0 + n_1 + n_2 + n_{\text{fail}}},$$

where n_i is the number of submissions accepted after exactly i reminders, and n_{fail} is the number still not accepted. Retries are not new submissions. The score is undefined when there are no recorded submissions. Passing the format check does not establish mathematical correctness.

Description fidelity. A judge compares finalized result and negative-result records with the execution log. It labels each record correct, incorrect, or undetermined, with counts N_c , N_i , N_u . We define $\eta = 100 N_c / (N_c + N_i)$ and report all three counts alongside η . The score is undefined when every judgment is undetermined. We separately report where objects came from, declared dependencies, completeness of the record, and consistency of the final submission. Description fidelity does not affect the progress score.

Tool calling. Let T be all tool calls and $T_v \subseteq T$ those with parseable actions and valid arguments. Validity is $|T_v|/|T|$. Termination is the fraction of valid calls that finish running; a process that exits with a nonzero status still counts as terminated. This measures tool reliability, not how the overall run ended. Figure 6 illustrates U_r and K_r with a small example graph.

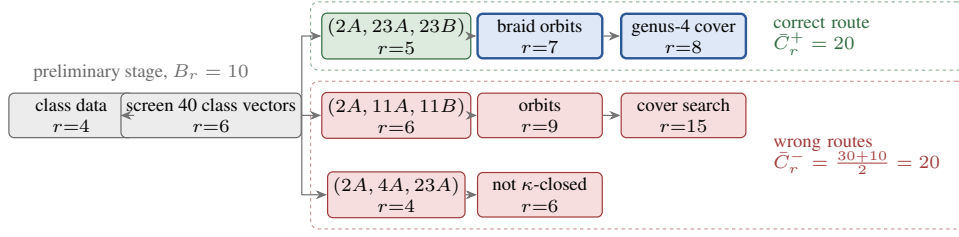


Figure 6: U_r and K_r on an example graph. Each finalized artifact has cost $r(v)$. Bold blue boxes mark verified checkpoints. Grey is screening ($B_r = 10$), green is a correct route from the literature, and red is research on wrong routes. Only one red route has the displayed conjugation obstruction. Credited artifacts and their ancestors cost 30 out of 70 in total, so $U_r = 100 \cdot 30/70 \approx 43$. The two wrong routes cost 30 and 10, giving $\bar{C}_r^- = 20$ and $K_r = 100(10 + 20)/(10 + 20 + 20) = 60$. When defined, $K_r = 100$ signifies no post-screening cost on wrong routes. It does not mean that every stopping decision was correct.

Tasks without a known answer. Tier 2 tasks have no reference polynomial, and Tier 3 tasks have no correct route in the literature. We can still measure progress on both. The screening score depends on verified properties: a verified Nielsen count or rationality obstruction earns the same credit on every task. For Tier 2, the ledger records the rigid class vector that proves existence. Route identification and checkpoints up to the cover can therefore be scored as in Tier 1, without a reference polynomial. For Tier 3, a class vector with a generating tuple that passes the structural verifier but is absent from the ledger is recorded as a *new route*. It earns screening credit immediately and route-identification credit after independent verification. A verified realization of an open group would be a new theorem, which we would report together with the system that produced it.

In this release, the route-identification score is enabled only where correct routes have been registered: the five Mathieu tasks and four other tasks with custom ladders. Tier 3 tasks instead use the open-case profile in Appendix H.2, which awards R for an exact obstruction to a candidate passport or a verified cover on one. The remaining tasks have a maximum score of 70 until registration is complete.

Route classification and undefined cases. A literature route counts as correct after structural verification against a registered ledger case. New routes require independent verification. Let $n_{\pm}$ and $C_r^{\pm}$ be the number and total post-screening cost of correct and wrong researched routes. We set $\bar{C}_r^{\pm} = C_r^{\pm}/n_{\pm}$ when $n_{\pm} > 0$, and 0 otherwise. If the agent researches only wrong routes, $K_r = 100 B_r / (B_r + \bar{C}_r^-)$. We report K_r as N/A when no route reaches the research stage or the denominator is zero. A class vector examined only during screening contributes to B_r but not to either route count. Screening and research costs are never counted twice.

When the evaluator did not resolve every route’s status, we compute K_r by classifying routes retrospectively: a route is correct only if its class vector appears in the literature census. The released data list these values per run, and Appendix H identifies which value each run uses. Unattributed cost is excluded from the denominator of U_r . A run that records only a small part of its work in artifacts is therefore scored only on that part.

F EXPERIMENTAL SETUP DETAILS

Sandbox. Each run uses an evaluator-owned workspace with Bubblewrap filesystem isolation and a seccomp policy blocking all agent network access except Magma’s license check. The generic environment description lists Python 3.12 with SymPy and mpmath, GAP 4.15.1 with ctblib, atlasrep, tomlib, and mapclass, Magma V2.29-9, and ripgrep. The GPT-6 Sol captures instead record Python 3.11.16 and Magma runtime paths naming V2.26-5. Their Magma version-probe field is null. We therefore cannot claim a single verified software version across all cohorts.

Jobs run on an institutional HPC cluster, with one problem per Slurm array task and 8 cores per job. Three separate limits apply. Agent shell commands have a per-command timeout of 7,200 s in the main campaign, though many coded runs, including Opus and campaign-02 runs, record an 1,800 s

tool limit. Verifier Magma calls have a 21,600 s limit. Braid-orbit and Nielsen enumerations are capped at 2×10^6 classes.

Verification. A call that exceeds its cap returns `unsupported`. The description-fidelity judge in Appendix E is Claude Sonnet 5, called through the Claude Code CLI with tools disabled.

Reporting. Tasks are sampled deterministically from the catalog using the tier and seed. Every run in a campaign uses the same configuration file.

Table 4: System configurations for the eligible runs. Every run has the main-campaign budget: 10 hours, a 500k output-token cap, eight CPUs, and task-dependent memory of 64–192 GB. Each system has two runs per task. Claude Fable 5.1’s two runs per task come from two later batches, on 23 and 29 September 2026. Earlier 6-hour GPT-5.6 Sol and Terra repetitions and the Claude Haiku campaign are excluded (Appendix H).

System	Model id	Harness	Effort	Runs per task	n
Claude Opus 5	claude-opus-5	Claude native (Claude Code)	xhigh	2	10
GPT-6 Sol	gpt-6-sol	Codex native	xhigh	2	10
GPT-6 Astra	gpt-6-astra	Codex native	xhigh	2	10
Claude Fable 5.1	claude-fable-5-1	Claude native (Claude Code)	max	2	10
GPT-5.6 Sol	gpt-5.6-sol	Codex native	xhigh	2	10
GPT-5.6 Terra	gpt-5.6-terra	Codex native	xhigh	2	10
Claude Sonnet 5	claude-sonnet-5	Claude native (Claude Code)	xhigh	2	10
DeepSeek V4 Pro	deepseek-v4-pro-0813	OpenRouter adapter	xhigh	2	10
GLM-5.3 Flash	z-ai/glm-5.3-flash	OpenRouter adapter	xhigh	2	10

G GALOIS DETAILS

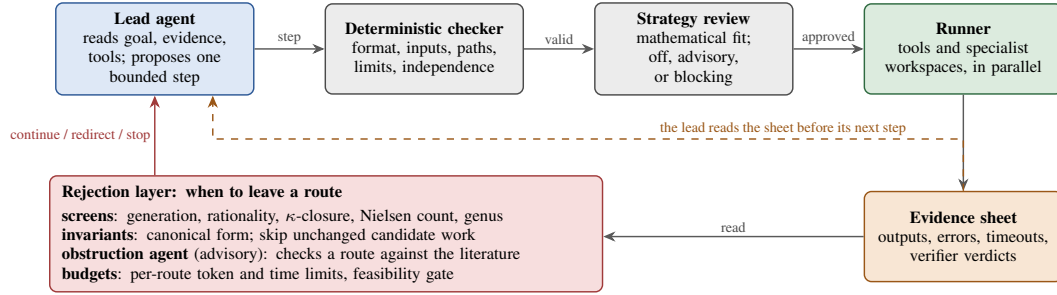


Figure 7: **The GALOIS control loop.** The lead proposes one resource-limited step. A deterministic checker validates it, an optional strategy reviewer checks its mathematical fit, and a runner executes it using registered tools or specialist workspaces. Outputs, timeouts, and errors go into an evidence sheet that the lead reads before its next step. The rejection layer uses cheap screens to admit and rank candidates, normalization invariants to avoid repeating unchanged work, and an obstruction agent to review the literature. The obstruction agent only advises; rejection requires an exact certificate (Table 6). Budget rules may defer work without proving the route impossible.

Skills by stage. Table 7 lists the 36 skills in the run commit and the pipeline stage served by each. Every skill is a written procedure for a stage in Appendix B, intended for any registered group with trivial centre and any class vector. The procedures and their twelve worked examples use symbolic descriptions or self-test instances such as A_5 , S_3 , and synthetic passports. GAP tools take the group from the engine registry.

At the run commit, that registry held 27 groups, 22 with a fixed permutation representation: the five Mathieu groups, J_1 , J_2 , HS , Co_3 , $Sz(8)$, $PSU_3(3)$, $PSU_4(3)$, $PSU_5(2)$, and $PSL_2(p)$ for nine primes. The tool registry contains 38 three-point tools, 72 four-point tools, and 12 legacy tools.

Table 5: When GALOIS defers or leaves a route.

Rule	Effect
First-success stop	a certified polynomial ends the run
No-progress guard	the run halts after repeated steps with no new candidate or artifact
Typed precision ladder	precision rises only for a stated reason; otherwise the cover is deferred
Evidence ledger	numerical covers are classified to reorder work, never to reject it
Route transition	once every retained cover of a route is rejected or deferred, the next route starts
Four-point feasibility	a seed search is skipped when its projected coverage is negligible

Table 6: Who may reject a route in GALOIS. Only the first row may reject a cover or passport, and only within the proved scope. A passport records a cover's degree and branch-cycle classes. All other checks return `may_reject = false`.

May reject	exact field-of-moduli obstruction in Magma (invariants of a bound genus-0 candidate, symmetrized over branch permutations. Verdict <i>not defined over</i> $\mathbb{Q}$ on a selected cover). Audited degree-11 passport template. Elementary passport necessities replayed in GAP (Riemann–Hurwitz, dihedral bound, branch-cycle lemma). Verified descent or Stage-11 regularity negatives replayed independently
May only advise	entry screens and route ranking (Nielsen count, genus, real-pairing heuristics). Cheap precheck. The 14-method obstruction package (genus-one and hyperelliptic moduli, ramification-divisor invariants, field-of-moduli degree, normalization-invariant search, and others). The parallel obstruction agent (at most two 15-minute assessments per candidate, one 5-minute experiment, eight 10-minute research rounds). The candidate reviewer (records repair, obstruction experiment, defer, reject-with-validation, escalate precision, switch lane, or continue). The strategy reviewer

Table 7: The 36 GALOIS skills in the run commit, grouped by pipeline stage (Appendix B). Each skill gives a procedure for a stage and can be applied to any registered group and class vector. Stage numbers marked (4) belong to the four-point path.

Stage	Skill	Procedure
<i>Group data and census</i>		
1	group-setup	Verify the group's pinned permutation representation from the engine registry, its order and class count against the ATLAS table, and emit the labelled class table with power maps and rationality flags.
1	group-setup-research	Register a group the ATLAS libraries do not know (names that Atlas-Rep or ctblib know are registered without an agent).
1	class-vector-census	Enumerate every unordered multiset of three and of four nontrivial classes with the exact product-one count by the Frobenius character formula, the genus, and a feasibility flag.
1	dossier-extension	Group dossier: the three-point census with exact Nielsen counts and the admissible $\{M, M, N, N\}$ four-point vectors consumed by Stages 2 to 6.
1	real-pairing-filter	Necessary-condition filters on class vectors (real-pairing conditions, genus cap) and a plausibility ranking. Advisory, never a route certificate.
1	candidate-ranking	Exploratory ranking of census rows by raw Nielsen mass and source genus. Advisory input to the route portfolio.
<i>Orbits and Nielsen classes</i>		
2–4	orbit-decomposition	Decompose the ordered product-one fibre of a four-class vector into pure-braid orbits with witnesses (GAP job).
2–4	genus-screen	Independently re-derive the λ -line genus of every pure-braid orbit component from its degree and cycle counts.
2–4	targeted-orbit-closure	Close seed tuples under targeted pure-braid moves, compute λ -line genus and cusps, and recognise Häfner forms.
2–4	hafner-bridge	Häfner's bridge from three-point triples to length-two genus-zero four-point pure orbits (optional).
2–4	three-point-orbits	Exact braid-orbit inventory of a three-point class vector on generating product-one tuples (optional).
5–6	nielsen-representatives	$\mathbb{Q}$ -rational branch configuration with the equivariance check, exact Nielsen count, one certified representative per orbit, and the action of complex conjugation with the real orbits.
6	cover-galois-orbits	Galois-orbit partition of a cover set by simultaneous lattice recognition, the separating invariant of Stage 6.

continued on next page

Table 7 continued

Stage	Skill	Procedure
<i>Numerical cover, exact curve, covering map</i>		
7	belyi-numerical-cover	Numerical differentials and triangle data of one cover in the Belyi-package disc coordinate.
7	belyimap-frame-transform	Coordinate-transform certificate from the package uniformiser to the working frame, with the echelon and Weierstrass record.
8	curve-reconstruction	Exact curve model over a number field from the accepted coordinate series, with a case-by-case escalation route.
7-9	genus-zero-exact-reconstruction	Exact-first lane for genus-zero covers: the rational function whose factorisation the three ramification partitions prescribe.
7-9	certified-branch-cycles	Certify the monodromy permutations of an exact genus-zero map by ball arithmetic, tying it to the selected Nielsen representative.
9	covering-function	The exact covering map $t: X \rightarrow \mathbb{P}^1$ from the exact curve and its distinguished points (Magma).
<i>Descent, family, certificates</i>		
10	descent-and-auxiliary-function	Descent datum of the pair (X, t) to $\mathbb{Q}$, the fixed field, the auxiliary function s with $\mathbb{Q}(X) = \mathbb{Q}(t, s)$, and the primitivity certificate.
11	polynomial-construction	Build $F(T, V)$ from the $\mathbb{Q}$ -model bundle: solve, CRT, reduce, verify, certify.
11	monodromy-certificate	The regularity certificate, items 1 to 7, for an accepted $F(T, V)$.
12	specialization-certificate	Certified specialisation: sweep a height box, choose t_0 , prove irreducibility and the Galois group, reduce, and check field equivalence and ramified primes.
<i>Four-point lane</i>		
7(4)-12(4)	four-point-construction	Lane index for the four-point arithmetic lane and the legacy bounded exact rational-function construction.
7(4)	four-point-configuration	Bind a selected four-class vector to its construction problem: action, genus, normaliser and forced-rational-place checks, exact Nielsen recount, chamber counts, chart recommendations.
8(4)	four-point-ramification-seed	Compile a genus-zero ramification chart and find verified finite-field seeds with the target monodromy group.
8(4)	four-point-local-recovery	Hensel-lift a verified seed, search residue discs, reconstruct over $\mathbb{Q}$ or a small number field with held-out digits, and verify exactly.
8(4)	four-point-family-reconstruction	The p -adic one-parameter family through a verified seed, bidegree-bounded relations, Magma analysis of the relation curve, and exact evaluation at a rational point.
9(4)	four-point-exact-export	Export an exact four-point model as polynomial, model, and branch artifacts for the verification stages.
<i>Obstructions (rejection evidence)</i>		
any	inverse-galois-obstructions	Select and run bounded exact obstruction experiments: class compatibility, real structures, intrinsic four-point invariants, local points.
any	exchange-orbit-descent-obstruction	Criterion on the $G_{\mathbb{Q}}$ -action on the exchange orbits of a three-point passport, read through an intrinsic invariant of a bound exact cover.
any	three-point-passport-obstruction	Low-degree intrinsic invariants for unresolved three-point passports, separating marked real obstructions from branch-symmetric passport obstructions.
any	genus-one-moduli	Exact j -invariant diagnostics of a genus-one source curve.
any	hyperelliptic-moduli	Exact Igusa or Shioda invariants of genus-2 and genus-3 hyperelliptic source curves for nonrational-moduli obstructions.
any	ramification-divisor-invariants	Exact projective invariants of ramification-point blocks on a genus-zero source, for field-of-moduli obstructions without full cross-ratio traces.
any	invariant-relation-certification	Turn a conjectured relation for a geometric invariant into an exact open-chart identity.

Task-specific content in the runtime checkout. Each run starts from a checkout prepared by `scripts/make_blind_snapshot.sh`. It contains the engine, schemas, templates, knowledge files, skills without self-test folders, tools, scripts referenced by runtime code, and one obstruction reference. It excludes git history, documentation, reports, tests, and earlier runs.

We searched these files at the run commit for class vectors from the five tasks. None of the eight knowledge files names a Mathieu group. The development case cards, which record each task's certified route and queue order, were moved out of the knowledge directory on 25 September 2026 and are absent from the checkout.

Route names do appear elsewhere. $(2A, 23A, 23B)$ appears in 23 files: 19 scripts and templates for Stages 5 to 12 use it as an example class-vector tag, and four engine modules mention it in development notes. $(2A, 11A, 11B)$ appears in nine similar files. The finished routes for M_{11} , M_{12} , and M_{24} each appear in one to three engine docstrings as cases used to measure a tool. The exact field-of-moduli tool's docstring names the M_{11} three-point route it decides. One obstruction script is specialized to the ramification profile of the M_{24} three-point route.

No file labels a vector correct, rejected, or preferred. The census and ranking tools compute the lead’s queue at run time. Nevertheless, a lead with file access could read these route names. We have not checked the read records of the runs in Table 2 against this list.

Run configuration. The runs in Section 4 use Claude Opus 5.5 at medium effort through the Claude CLI as lead. Strategy review is off, the obstruction agent is on, the system stops at the first certified specialization, and there is no spend cap. Each run begins with a fresh runtime-only checkout. Heavy tools run on the same 16-core, 110 GB workstation.

The skills contain methods from the constructive literature: census and rigidity screens, numerical uniformization, recognition, descent, and certification. This mathematical knowledge is part of the system. The runs therefore provide reference timings for the tasks, not a like-for-like comparison with single agents. An audit of every tool call in every agent session (`scripts/audit_lead_reads.py`) found no reads outside the checkout in any run in Table 2.

Scoring the GALOIS runs. These runs have neither an artifact graph nor an evaluator record. The GALOIS row in Table 3 is therefore computed from pipeline records by `scripts/collect_galois_runs.py`, with output in `data/galois_runs.csv`. We apply the paper’s definitions to pipeline steps rather than artifacts. Each step belongs to the class vector named in its plan entry. Steps 1 to 3—group setup, exact Nielsen census, and route selection—count as preliminary screening.

Every finished route is on the ledger’s list of correct routes: $(2A, 2A, 3A, 4A)$, $(3A, 3A, 6A)$, $(2A, 11A, 11B)$, $(2A, 23A, 23B)$, and $(2A, 2A, 2A, 12B)$. The two rejected source routes are not. C is 40 for all ten runs; for M_{22} , the polynomial is the $T = 1$ specialization of the run’s regular family, verified with Magma (`data/galois_verification.txt`). We assign full S and R : the computed exact census, Nielsen representatives, and orbit data exceed the research cap, and every run has a verified generating tuple on its correct route.

We compute U_r and K_r from the output tokens of agent calls assigned to each step. For U_r , we count screening and all steps on the finished route as useful. This is a step-based proxy, not the artifact-graph definition. Since each run has at most one wrong route, the proxy equals K_r . We do not report format compliance, description fidelity, tool-call validity, or termination for these runs.

The ten runs average $P = 100$, POLY 100, and $U_r = K_r = 82.8$ (Table 3). These scores are not directly comparable with the evaluator’s scores: the runs had no token cap, used a different lead model and effort, ran without the bridge, and were classified by us using their plan entries.

H EXPERIMENTAL RESULTS

H.1 RUN SELECTION, SOURCE RECORDS, AND KNOWN PROTOCOL ISSUES

The analysis uses the eligible summaries in the supplied paper data: 10 Opus, 10 GPT-6 Sol, 10 Astra, 10 Fable, 10 GPT-5.6 Sol, 10 Terra, 10 Sonnet, 10 DeepSeek, and 10 GLM task-runs. This gives 90 runs, with two per task for each system. We include only Opus runs at xhigh effort. We exclude the post-hoc Astra M_{11} replacement, which was chosen after the preregistered run scored 4; using it would raise Astra’s mean P from 59.4 to 69.0. We also exclude the earlier 6-hour GPT-5.6 Sol and Terra repetitions to keep two runs per task. Claude Haiku is absent from the aggregate because it has only five valid runs, all from the earlier 6-hour campaign. Its 10-hour batch lacked Magma on the agent path, and its verified progress was near zero.

Claude Fable 5.1 ran after the main campaign, at effort *max*, in two batches with one run per task: on 23 September 2026 at benchmark revision `f9d7282` and on 29 September at `99c1025`. Both used the Claude native harness, the same budgets, and the resolved model `claude-fable-5-1`. In the second batch, four first attempts stopped early on a subscription usage limit and were rerun under new run identifiers. We use the reruns. The 10-hour budget of the second M_{23} run ran out before its final evaluation, so its record is a partial evaluation at 9.86 hours. Its scores are carried forward from the last verifier pass, and its token and time totals come from its last 15-minute sample. The second batch has no retrospective census, so its route counts come from the evaluator’s route audit.

GPT-6 Sol’s runs come from a two-wave campaign. Because this campaign ran without the description-fidelity judge, we obtain η by re-running the pinned judge on archived records and recompute I_f from the format logs.

We use the same procedure to obtain η for DeepSeek and GLM, whose judge calls failed on the cluster, and for the campaign-02 M_{23} runs of GPT-5.6 Sol and Terra, where the judge was switched off. We also recompute I_f for the ten campaign-02 GPT-5.6 runs because their campaign summary lacks format fields. Five GLM runs finalized no record with a determinate judgment, so they have no η .

All runs use the literature-based correct-route rule in Section 3.4. The route-identification score and its screening top-up count only ledger routes in the target group whose class vectors the evaluator lists as correct. The top-up is the screening credit attached to route identification. Earlier benchmark versions also accepted some routes in an almost-simple overgroup. We replayed every archived run that had earned route-identification credit on a now-ineligible route through the current evaluator.

Under the rule in force during the campaign, two scores changed in that replay: the Sonnet M_{22} batch-2 run fell from 60 to 0 because its $\text{Aut}(M_{22})$ route was no longer eligible, and the GPT-5.6 Sol campaign-02 M_{22} run fell from 100 to 46, keeping only the construction score of its Magma-verified polynomial on $(2B, 5A, 12A)$. On 30 September 2026 we revised the rule (`positive-routes-v2`): a literature route whose class vector lies in the registered automorphism group of the target is correct, and a route on which a run produces a verified polynomial is correct for that run when the run enforced the known-answer-store block. The benchmark’s `rescore-routes` command applies the rule to an archived run from its stored records and re-runs no verifier. We rescored the 15 runs whose full records we hold: all ten Fable runs and the five GPT-5.6 Sol campaign-02 runs. Among them, Fable’s second M_{11} run rose from 46 to 100 (a polynomial through an M_{12} route) and the GPT-5.6 Sol campaign-02 M_{22} run from 46 to 100; three Fable runs gained K_r because their work on $\text{Aut}(M_{22})$ or $\text{Aut}(M_{12})$ routes now counts as correct; and the five GPT-5.6 Sol campaign-02 runs received evaluator K_r values in place of census values. Two runs without records are reconstructed from documented facts: Opus’s second M_{11} run (52 to 100, a polynomial through an M_{12} route with stored answers blocked) and the Sonnet M_{22} run (0 to 60, restored). The two Astra runs with polynomials on unlisted routes keep $R = 0$, because their records do not establish that stored answers were blocked. Every other run keeps its value, since the retrospective census already counted literature overgroup routes as correct. The released data hold the sheets under both rules and a log of every changed row.

We audited where the final polynomials came from. Every certified Opus polynomial—five in the eligible set—was constructed during its run. Opus records carry the enforced flag `no-known-answer-stores-v1`, which masks Magma’s polynomial database, stored number fields, and stored Belyi maps. All ten GPT-6 Sol configurations and all ten Fable configurations also enforce this flag. The supplied records do not establish uniform blocking for the other systems. Astra’s M_{12} and M_{22} polynomials were constructed on routes it chose, using Gröbner or finite-field search with p -adic lifting followed by GaloisProof. GPT-5.6 Sol’s one polynomial is the campaign-02 M_{22} construction on the $\text{Aut}(M_{22})$ route.

We reported two verifier issues to the maintainers. First, the verifier rejected a non-monic integral multiple of a correct M_{11} polynomial in the GPT-5.6 Sol campaign-02 run and the Astra batch-1 run. Both therefore lost terminal credit despite passing their own GaloisProof checks. Second, Opus’s second M_{11} run realized M_{11} through an M_{12} route but earned no route credit because the interface does not support a route recorded in an overgroup.

For K_r , we use the evaluator’s value wherever one was preserved in a run summary or census sheet: all Astra and Fable runs, the five rescored GPT-5.6 Sol campaign-02 runs, eight DeepSeek runs, six GLM runs, and one Opus run, or 40 runs in total. We use the census value for the other 46 runs. Four GLM runs recorded no route research and have no K_r .

H.2 TIER 2 AND 3 TASKS UNDER THE OPEN-CASE PROFILE

Tier 3 tasks have no correct route in the literature, so the route-identification score from Section 3.4 is undefined. The *open-case* profile keeps the 30/30/40 point split and generic construction ladder, but replaces R with a *passport resolution*.

A *candidate passport* for G is an unordered class vector $C = \{C_1, C_2, C_3\}$ satisfying three conditions. It is Galois-stable: closed under $C \mapsto C^k$ for k coprime to $\text{lcm ord } C_i$. It contains at least $|G|/|Z(G)|$ product-one triples, the size of one free conjugation orbit of a generating triple. And its Riemann–Hurwitz genus in the registered action is at most $g_{\max}$. We use $g_{\max} = 2$ by default. If there is no candidate of genus ≤ 2 , we raise the bound to the smallest candidate genus: 4 for $\text{Sz}(8)$ and 9 for $\text{PSL}_2(27)$. Every three-point regular realization of G over $\mathbb{Q}(T)$ with source genus at most $g_{\max}$ has branch-cycle data in a candidate passport.

The first artifact that *resolves* a candidate earns $R = 30$, once per run. A resolution can be either (i) an exact obstruction recomputed by the evaluator in GAP, or (ii) a verified cover over $\mathbb{Q}$ with that passport, at checkpoint S3 or higher. Two obstruction facts qualify. Fact N7 says that no inner Nielsen class of the passport is fixed by any of the four real structures of a conjugation-stable three-branch-point configuration. Thus no cover in that passport is defined over $\mathbb{R}$, and hence none over $\mathbb{Q}$. Fact N4 says that every product-one triple generates a proper subgroup.

As with the Tier 1 and Tier 2 route-identification score, a resolution brings S up to 30. In addition, S credits a correct passport census (C1) and correct counts of inner Nielsen classes and real classes (C2). Numerical results earn no credit, including unrecognized invariants computed from numerical covers.

Table 8 reports one Claude Opus 5 run on each of the ten tasks listed as Tier 3 in the campaign build, at maximum reasoning effort, with 500,000 output tokens and eight CPUs. Five of these tasks are Tier 2, four are Tier 3, and one is Tier 1. We selected them using a Magma screen of class triples: cyclotomically closed triples, structure constants, Riemann–Hurwitz genus, and sampled generating Nielsen classes. We excluded groups with a rational rigid generating triple and ranked the rest by the fewest generating classes and lowest genus. The runs launched in parallel with a five-hour wall budget. A harness restart used the first 2 h 12 min, and the agents were stopped gracefully at the original deadline. Each task therefore had 2 h 48 min of effective agent time. Finalized artifacts were scored as they would be after a wall-time stop. Six runs proved an obstruction resolving a candidate within the first hour and a half. All ten reached the braid-orbit checkpoint, and none claimed a cover.

The open-case scores do not depend on the tier label.

For comparison, three exploratory Claude Opus 5 runs from September 2026, with uneven budgets and manual stopping, score 15 on $\text{PSU}_3(4)$, 12 on $\text{PSL}_3(8)$, and 12 on $\text{PSU}_3(8)$ under the same profile. They computed censuses and Nielsen classes but resolved no candidate passport. Re-verifying the $\text{PSL}_3(8)$ run showed that its passport $(3A, 3A, 4A)$ has 24 real inner Nielsen classes, twice the 12 analyzed by the pipeline. The discrepancy arises because the pipeline checks only the all-real structure when every class is rational.

Table 8: Tier 2 and 3 tasks under the open-case profile. One Claude Opus 5 run per task at maximum effort, with 2 h 48 min of effective agent time, scored after a graceful stop. S , R , C , and P are raw points, with maxima 30, 30, 40, and 100. *level* is the deepest verified checkpoint. *cand.* counts candidate passports of genus at most $g_{\max}$ in the evaluator’s census; *res.* counts those the run resolved. *resolution* gives the passport, genus, and obstruction fact. *tok.* gives output tokens in thousands. Class names follow the GAP character-table library.

Group	$g_{\max}$	cand.	res.	S	R	C	P	level	resolution (passport, genus, fact). Tok.
Sz(8)	4	1	1	30	30	0	60	S2	$(2a, 5a, 5a)$, 4, N7. 371k
$\text{PSU}_3(4)$	2	1	1	30	30	0	60	S2	$(3a, 3a, 4a)$, 2, N7. 348k
$\text{PSU}_3(5)$	2	16	2	30	30	0	60	S2	$(2a, 5a, 6a)$, 0, N4. $(2a, 4a, 10a)$, 0, N7. 373k
$\text{PSL}_3(7)$	2	13	4	30	30	0	60	S2	$(2a, 4a, 7b)$, 0, N7. $(2a, 4a, 14a)$, 0, N7. 402k
$\text{PSL}_4(3)$	2	211	1	30	30	0	60	S2	$(3a, 5a, 8a)$, 2, N7. 363k
$\text{PSL}_2(16)$	2	2	2	30	30	0	60	S2	$(2a, 5a, 5b)$, 0, N4. $(3a, 5a, 5b)$, 1, N4. 288k
$\text{PSL}_2(125)$	2	1	0	23	0	0	23	S2	none. N7 only on $(7a, 7b, 7c)$ of genus 37. 344k
$\text{PSU}_3(8)$	2	4	0	23	0	0	23	S2	none. N7 only on passports of genus 22 and 64. 467k
Sz(32)	2	0	0	23	0	0	23	S2	none. No three-point candidate of genus ≤ 12 . 290k
$\text{PSL}_2(27)$	9	1	0	22	0	0	22	S2	none. 441k
mean				27.1	18.0	0.0	45.1		

H.3 UNCERTAINTY, STOPPING STATISTICS, THE M_{23} RUNS, AND HOW RUNS ENDED

Table 10 reports K_r and U_r in three units and under both route-classification procedures. Table 11 gives bootstrap intervals, Table 12 lists every M_{23} run, Table 9 compares the evaluator’s route-identification score with the retrospective census, and Table 13 records how the runs ended.

Table 9: Runs where earning the milestone ($R > 0$) differs from researching a route labeled correct by the retrospective census. Nine runs research a census-correct route without earning the milestone; six earn it without a correct census count. Fable’s second batch has no census and is not compared. The two tests need not agree because R also requires a verified certificate. This comparison is diagnostic, not a count of scoring errors.

System	Task	Run	R	Census correct	Census wrong	K_r tok.
GPT-5.6 Sol	M12	sol_A	0	1	18	78.3
Claude Sonnet 5	M22	sonnet_A	100	0	8	69.9
Claude Opus 5	M23	opus_A	100	0	1	17.6
GPT-5.6 Sol	M12	sol_A2	0	1	7	37.5
GPT-5.6 Sol	M22	sol_A2	100	0	6	93.0
GPT-5.6 Sol	M24	sol_A2	100	0	6	40.9
GPT-5.6 Terra	M11	terra_A2	0	1	3	67.0
GPT-6 Astra	M11	astra_A	0	1	1	28.6
GPT-6 Astra	M12	astra_A2	0	1	3	63.2
GPT-6 Astra	M22	astra_A2	0	1	1	47.9
GPT-6 Sol	M22	sol6_A	100	0	17	60.5
GPT-6 Sol	M12	sol6_A2	0	1	28	83.1
GPT-6 Sol	M22	sol6_A2	100	0	11	49.5
Claude Fable 5.1	M11	b1-fable-max-m11	0	2	0	54.1
Claude Fable 5.1	M23	b1-fable-max-m23	0	1	106	95.7

Table 10: Stopping statistics by system. Route counts come from retrospective classification of the artifact graphs. Claude Fable 5.1’s second batch has no census, so its route counts come from the evaluator’s route audit, and its census column averages the first batch only. We report K_r under the paper’s rule—use the evaluator’s value when recorded, otherwise the census—in three resource units, plus token-based K_r using only the census. We also report U_r in three units and mean tool CPU hours per run. The first K_r column is used in Table 3. Means exclude undefined values: four GLM runs in every unit and one Sonnet run in CPU hours.

System	n	Routes/run		K_r (paper rule)			K_r		U_r		
		all	non-pos.	tok.	CPU	time	census tok.	tok.	CPU	time	CPU h
Claude Opus 5	10	10.5	9.3	72.8	53.0	66.0	72.8	57.7	60.2	56.3	2.2
GPT-6 Sol	10	20.5	19.3	79.1	61.1	75.6	79.1	31.4	29.7	30.7	0.7
GPT-6 Astra	10	3.8	2.9	55.9	54.4	59.4	58.0	60.3	62.8	56.8	0.4
Claude Fable 5.1	10	22.9	21.9	92.7	68.8	84.9	98.0	72.2	67.8	61.0	5.7
GPT-5.6 Sol	10	9.3	8.8	62.9	38.8	53.3	57.8	27.5	23.3	23.7	2.6
GPT-5.6 Terra	10	6.9	6.6	38.2	34.2	31.6	38.2	24.6	13.8	16.9	3.8
Claude Sonnet 5	10	3.9	3.9	30.6	8.2	20.3	30.6	26.9	28.5	26.3	2.4
DeepSeek V4 Pro	10	2.7	2.7	66.6	44.5	51.2	66.6	34.4	20.7	25.9	2.5
GLM-5.3 Flash	10	1.4	1.4	63.9	48.9	55.4	63.9	33.5	35.9	31.9	0.5

Table 11: System means with 95% bootstrap intervals from 10,000 resamples. *Over runs*: percentile intervals from resampling task-runs with replacement. *Task-clustered*: resample the five tasks with replacement, retaining every run of each sampled task.

System	n	over runs			task-clustered	
		P	R	K_r (tokens)	P	K_r (tokens)
Claude Opus 5	10	81.0 [70, 92]	100.0 [100, 100]	72.8 [56, 87]	[66, 96]	[57, 89]
GPT-6 Sol	10	62.1 [48, 76]	80.0 [50, 100]	79.1 [67, 91]	[54, 72]	[64, 93]
GPT-6 Astra	10	59.4 [40, 78]	50.0 [20, 80]	55.9 [34, 77]	[46, 72]	[27, 82]
Claude Fable 5.1	10	62.4 [45, 80]	70.0 [40, 100]	92.7 [83, 99]	[45, 83]	[85, 99]
GPT-5.6 Sol	10	44.2 [27, 62]	50.0 [20, 80]	62.9 [48, 78]	[27, 58]	[47, 77]
GPT-5.6 Terra	10	21.9 [9, 36]	20.0 [0, 50]	38.2 [23, 53]	[9, 38]	[22, 54]
Claude Sonnet 5	10	17.5 [8, 29]	10.0 [0, 30]	30.6 [14, 48]	[7, 28]	[13, 48]
DeepSeek V4 Pro	10	7.8 [5, 11]	0.0 [0, 0]	66.6 [49, 83]	[6, 11]	[46, 84]
GLM-5.3 Flash	10	2.6 [0, 6]	0.0 [0, 0]	63.9 [51, 77]	[0, 5]	[49, 76]

Table 12: Every M_{23} run. Route counts use retrospective classification, except for Fable’s run 2, which uses the evaluator’s route audit. K_r is shown for output tokens and tool CPU hours; U_r uses output tokens. $C = 12.5$ is the percentage corresponding to 5 raw points at stage 2 of the task-specific ladder (Appendix C). N/A marks a run with no route research, for which K_r is undefined.

System	Run	P	R	C	Routes	Non-pos.	K_r tok.	K_r CPU	U_r tok.	Tokens (k)	Hours
Claude Opus 5	run 1	65	100	12.5	1	1	17.6	9.3	22.9	451	5.6
	run 2	65	100	12.5	10	9	78.6	46.1	35.7	460	10.0
GPT-6 Sol	run 1	35	0	12.5	25	25	43.8	4.3	31.2	158	10.0
	run 2	65	100	12.5	22	21	81.1	8.7	33.0	154	10.0
GPT-6 Astra	run 1	35	0	12.5	8	8	4.7	0.9	3.5	500	7.4
	run 2	35	0	12.5	13	13	6.9	3.5	6.1	504	8.2
Claude Fable 5.1	run 1	35	0	12.5	107	106	95.7	3.7	27.1	310	10.0
	run 2	35	0	12.5	53	53	88.3	8.6	21.8	360	10.0
GPT-5.6 Sol	run 1	35	0	12.5	11	11	39.7	4.1	20.0	83	3.5
	run 2	35	0	12.5	5	5	24.4	1.5	14.5	232	9.8
GPT-5.6 Terra	run 1	35	0	12.5	7	7	13.4	0.1	50.7	92	3.9
	run 2	65	100	12.5	14	13	62.4	0.8	23.2	290	10.0
Claude Sonnet 5	run 1	24	0	12.5	5	5	77.7	63.3	48.3	451	4.4
	run 2	35	0	12.5	2	2	40.1	0.0	64.0	411	5.8
DeepSeek V4 Pro	run 1	20	0	12.5	3	3	83.1	12.1	2.1	242	1.9
	run 2	10	0	12.5	11	11	98.2	96.4	15.5	211	2.6
GLM-5.3 Flash	run 1	0	0	0.0	0	0	N/A	N/A	0.0	509	2.4
	run 2	0	0	0.0	2	2	48.1	97.5	0.0	446	5.0

Table 13: How eligible runs ended, compared with how many reached 99% of a wall-time or token budget according to the counters. The recorded reason and counters do not always agree. One Opus M_{12} run is marked completed at 9.9 hours, and one Sonnet M_{22} run at 497k tokens. Seven GLM runs reached or exceeded the 500k cap by up to 5%, and one used the full 10 hours; all are marked completed.

System	Recorded termination				Derived from counters	
	Completed	Wall limit	Token limit	Other	Wall $\geq$ 99%	Tokens $\geq$ 99%
Claude Opus 5	7	2	1	0	3	1
GPT-6 Sol	4	3	3	0	3	3
GPT-6 Astra	7	1	2	0	1	2
Claude Fable 5.1	5	4	1	0	4	1
GPT-5.6 Sol	9	1	0	0	1	0
GPT-5.6 Terra	4	6	0	0	6	0
Claude Sonnet 5	10	0	0	0	0	1
DeepSeek V4 Pro	4	0	6	0	0	6
GLM-5.3 Flash	10	0	0	0	1	7

H.4 RESULTS BY TASK

Table 14 gives the scores of each system on each task.

Table 14: Verified progress by model and task. Components are percentages of their maxima. P is out of 100.

Model	Task	S	R	C	P	POLY	n
Claude Opus 5	M11	100.0	100.0	50.0	80.0	50.0	2
Claude Opus 5	M12	100.0	100.0	100.0	100.0	100.0	2
Claude Opus 5	M22	100.0	100.0	100.0	100.0	100.0	2
Claude Opus 5	M23	100.0	100.0	12.5	65.0	0.0	2
Claude Opus 5	M24	100.0	100.0	0.0	60.0	0.0	2
GPT-6 Sol	M11	100.0	100.0	0.0	60.0	0.0	2
GPT-6 Sol	M12	85.0	50.0	50.0	60.5	50.0	2
GPT-6 Sol	M22	100.0	100.0	50.0	80.0	50.0	2
GPT-6 Sol	M23	100.0	50.0	12.5	50.0	0.0	2
GPT-6 Sol	M24	100.0	100.0	0.0	60.0	0.0	2
GPT-6 Astra	M11	56.7	50.0	50.0	52.0	50.0	2
GPT-6 Astra	M12	60.0	50.0	100.0	73.0	100.0	2
GPT-6 Astra	M22	73.3	50.0	100.0	77.0	100.0	2
GPT-6 Astra	M23	100.0	0.0	12.5	35.0	0.0	2
GPT-6 Astra	M24	100.0	100.0	0.0	60.0	0.0	2
Claude Fable 5.1	M11	73.3	50.0	50.0	57.0	50.0	2
Claude Fable 5.1	M12	100.0	100.0	0.0	60.0	0.0	2
Claude Fable 5.1	M22	100.0	100.0	100.0	100.0	100.0	2
Claude Fable 5.1	M23	100.0	0.0	12.5	35.0	0.0	2
Claude Fable 5.1	M24	100.0	100.0	0.0	60.0	0.0	2
GPT-5.6 Sol	M11	100.0	100.0	0.0	60.0	0.0	2
GPT-5.6 Sol	M12	46.7	0.0	0.0	14.0	0.0	2
GPT-5.6 Sol	M22	56.7	50.0	50.0	52.0	50.0	2
GPT-5.6 Sol	M23	100.0	0.0	12.5	35.0	0.0	2
GPT-5.6 Sol	M24	100.0	100.0	0.0	60.0	0.0	2
GPT-5.6 Terra	M11	36.7	0.0	0.0	11.0	0.0	2
GPT-5.6 Terra	M12	56.7	50.0	0.0	32.0	0.0	2
GPT-5.6 Terra	M22	23.3	0.0	0.0	7.0	0.0	2
GPT-5.6 Terra	M23	100.0	50.0	12.5	50.0	0.0	2
GPT-5.6 Terra	M24	31.7	0.0	0.0	9.5	0.0	2
Claude Sonnet 5	M11	26.7	0.0	0.0	8.0	0.0	2
Claude Sonnet 5	M12	20.0	0.0	0.0	6.0	0.0	2
Claude Sonnet 5	M22	66.7	50.0	0.0	35.0	0.0	2
Claude Sonnet 5	M23	81.7	0.0	12.5	29.5	0.0	2
Claude Sonnet 5	M24	30.0	0.0	0.0	9.0	0.0	2
DeepSeek V4 Pro	M11	20.0	0.0	0.0	6.0	0.0	2
DeepSeek V4 Pro	M12	20.0	0.0	0.0	6.0	0.0	2
DeepSeek V4 Pro	M22	20.0	0.0	0.0	6.0	0.0	2
DeepSeek V4 Pro	M23	33.3	0.0	12.5	15.0	0.0	2
DeepSeek V4 Pro	M24	20.0	0.0	0.0	6.0	0.0	2
GLM-5.3 Flash	M11	16.7	0.0	0.0	5.0	0.0	2
GLM-5.3 Flash	M12	6.7	0.0	0.0	2.0	0.0	2
GLM-5.3 Flash	M22	0.0	0.0	0.0	0.0	0.0	2
GLM-5.3 Flash	M23	0.0	0.0	0.0	0.0	0.0	2
GLM-5.3 Flash	M24	20.0	0.0	0.0	6.0	0.0	2

H.5 CORRELATION AMONG METRICS

Figure 8 shows Spearman correlations between metrics across runs. Among the 38 runs that earned the route-identification score, the construction score correlates at 0.03 with K_r and 0.67 with U_r . Five of these runs reached the 500,000-token cap: Opus M_{24} , GPT-6 Sol M_{11} , both GPT-6 Sol M_{24} runs, and Fable’s second M_{24} run. All five have $C = 0$ and a mean K_r of 92. Four have U_r below 30, and Fable’s run has 72.6.

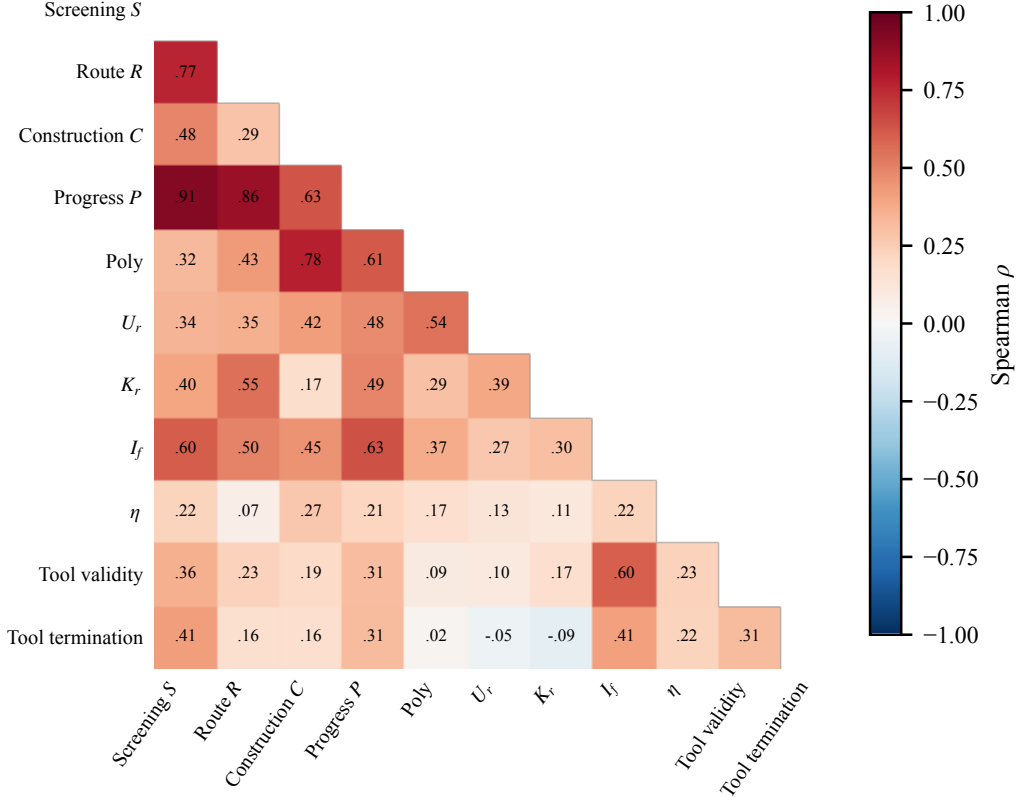


Figure 8: Spearman correlations between IGP-Bench metrics across the 90 eligible task-runs, using runs with both metrics defined for each pair. K_r is defined on 86 runs, I_f on 89, and η on 85.

H.6 CONSTRUCTION DEPTH AND ROUTE IDENTIFICATION

The five Mathieu tasks are hard enough to separate the systems into two groups. Ranking the nine systems by mean construction score and ranking them by mean route-identification score give the *same* five leaders: Claude Opus 5, GPT-6 Astra, Claude Fable 5.1, GPT-6 Sol, and GPT-5.6 Sol. Both components fall sharply after the fifth place, from 12.5 to 2.5 on the construction scale and from 50 to 20 on the route scale. The remaining four systems—GPT-5.6 Terra, Claude Sonnet 5, DeepSeek V4 Pro, and GLM-5.3 Flash—score at most 2.5 and 20 respectively, and across their 40 runs they produce no verified polynomial and earn route credit three times.

Those four systems carry no information about how the two abilities relate, because they are at the floor of both scales. Among the 50 runs of the five systems that make progress, $\rho(C, R) = +0.05$ (Table 15). Pooling all 90 runs instead gives +0.29, which is a between-group effect: the four systems at the floor supply the slope.

The grouping does not create the result. Because the same partition follows from ranking on either component, the grouping does not condition on one of the two correlated variables in pref-

erence to the other. Nor does it restrict their range in a way that would attenuate the correlation. Within the five systems the route score keeps almost its full dispersion ($\text{sd} = 13.9$ against 14.9 over all 90 runs, in raw points) and the construction score is *more* dispersed than in the full sample (18.3 against 15.1). A grouping that uses no score at all gives the same answer: the four systems of the current model generation (Claude Opus 5, Claude Table 5.1, GPT-6 Sol, GPT-6 Astra) give $\rho(C, R) = +0.02$ over 40 runs.

System means overstate the association. Correlating the two columns of Table 3 across the nine systems gives $\rho(C, R) = +0.85$ (exact permutation $p = 0.005$), far above the $+0.29$ over the 90 runs. Averaging ten runs per system removes the run-to-run variation while leaving the floor effect intact, so the system-mean coefficient mostly measures how far each system is from the floor. We report run-level coefficients throughout for this reason.

A contingency table is more informative than the coefficient. In this campaign C takes only the three values 0, 5, and 40, so the correlation is close to a comparison of route credit between runs that produced a verified polynomial and runs that did not. Among the 50 runs of the five systems, 14 of the 16 runs with a polynomial earned route credit (88%), against 21 of the 34 without (62%); Fisher’s exact test gives $p = 0.10$. The point estimates are thus weakly positive while the association remains unresolved at this sample size, and the claim in Sections 1 and 6 is a failure to detect an association, not a demonstration of independence.

Two further bounds on the claim. Within a single system, across its ten runs, the runs with a deeper construction earn route credit slightly more often: the point-biserial correlation between C and the route indicator averages $+0.20$ over the six systems whose runs vary on R . Separately, the correlation between S and R cannot be used as evidence about these abilities, because all 38 runs that earned R have S at its maximum of 30 through the route-completion top-up of Appendix E; after subtracting the floor that rule imposes on each system’s mean S , the residual relation with the route rate is -0.27 .

What holds without distributional assumptions. The two components order the systems differently. GPT-6 Sol identifies a correct route in eight of ten runs and has the lowest construction score of the leading four ($C = 22.5$), while GPT-6 Astra does so in five and ties Claude Opus 5 for the highest ($C = 52.5$); their progress scores are 62.1 and 59.4. The script `scripts/ability_separation.py` recomputes every number in this subsection from the released per-run data.

Table 15: Spearman correlation between the construction score C and the route-identification score R , with 95% bootstrap intervals from 10,000 resamples of runs. The first group is the five systems that lead on both components, which are the same five under either ranking; the second is the four systems of the current model generation, a grouping that uses no score. POLY counts runs ending with a verified polynomial, and R rate is the fraction earning route credit.

Group	Runs	POLY	R rate	$\rho(C, R)$	95% CI	$\text{sd}(R)$
The five systems that make progress	50	16	0.70	+0.05	$[-0.21, +0.30]$	13.9
Current model generation	40	15	0.75	+0.02	$[-0.25, +0.30]$	13.2
All systems pooled	90	16	0.42	+0.29	$[+0.07, +0.48]$	14.9

Construction and the progress score. Over the 50 runs of the five systems, construction correlates with the progress score at $+0.60$ ($[+0.32, +0.81]$). Because C is a summand of P , part of any such correlation is arithmetic. To size that part we permute $S + R$ across runs, which makes construction independent of the rest of the score, and recompute P : over 20,000 permutations the correlation averages $+0.69$ (95% range $[+0.51, +0.85]$), and 86% of permutations give a correlation at least as large as the observed one. The observed association is therefore no larger than the identity alone produces, consistent with the correlation of $+0.08$ between construction and the rest of the score, $S + R$ (the corrected item–total correlation of psychometrics). Pooled over all 90 runs, construction correlates at $+0.63$ with P and $+0.42$ with $S + R$, above the identity baseline of $+0.41$; as with route identification, the excess is a between-system effect of the four systems at the floor.

H.7 AN ARTIFACT GRAPH

Figure 9 shows the artifact graph of a Claude Opus 5 run that realizes M_{22} .

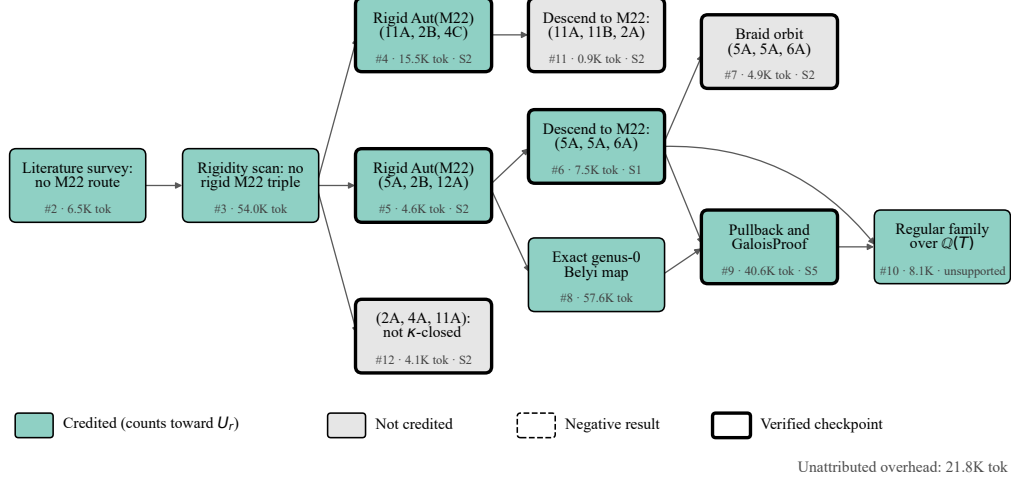


Figure 9: Artifact graph for a Claude Opus 5 run that realizes M_{22} ($P = 100$). Boxes are arranged by dependency depth and show output tokens and the strength of the certified checkpoint. Filled boxes count toward U_r ; bold borders mark verified checkpoint claims. The run finds no rigid rational triple in M_{22} , certifies two rigid triples in $\text{Aut}(M_{22})$, descends one to M_{22} , solves its Belyi map exactly, and certifies the pulled-back polynomial (S5).

H.8 ALL TASK-RUNS

Table 16 lists every eligible task-run behind the averages of Tables 3 and 14, one row per run.

Table 16: Scores for all 90 eligible IGP-Bench task-runs before averaging. Columns follow Table 3. S , R and C are percentages of their maxima. POLY is 100 when a run produces a verifier-certified final polynomial. Every row is a run with 10 h and 500,000 output tokens. *run 1* and *run 2* are the two repetitions of a task. Appendix H gives eligibility and source details. N/A means the metric is undefined for that run (Appendix E).

Task	Run	Verified progress (%)					Allocation		Protocol		Tool use (%)	
		S	R	C	P	POLY	U_r	K_r	I_f	η	Valid	Term.
Claude Opus 5 (10 runs)												
M11	run 1	100.0	100.0	0.0	60.0	0.0	21.0	89.8	98.5	100.0	100.0	96.1
	run 2	100.0	100.0	100.0	100.0	100.0	58.8	89.8	100.0	100.0	100.0	95.6
M12	run 1	100.0	100.0	100.0	100.0	100.0	94.8	100.0	86.4	100.0	99.4	87.4
	run 2	100.0	100.0	100.0	100.0	100.0	95.5	95.1	95.0	100.0	100.0	83.8
M22	run 1	100.0	100.0	100.0	100.0	100.0	95.1	40.7	100.0	100.0	100.0	81.2
	run 2	100.0	100.0	100.0	100.0	100.0	96.2	76.7	100.0	90.0	99.3	88.2
M23	run 1	100.0	100.0	12.5	65.0	0.0	22.9	17.6	96.9	100.0	100.0	91.3
	run 2	100.0	100.0	12.5	65.0	0.0	35.7	78.6	97.7	100.0	100.0	94.9
M24	run 1	100.0	100.0	0.0	60.0	0.0	29.0	78.2	97.9	100.0	100.0	97.1
	run 2	100.0	100.0	0.0	60.0	0.0	28.2	61.4	96.7	100.0	99.6	96.1
GPT-6 Sol (10 runs)												
M11	run 1	100.0	100.0	0.0	60.0	0.0	2.4	92.8	99.1	100.0	100.0	98.8
	run 2	100.0	100.0	0.0	60.0	0.0	3.8	90.4	98.9	100.0	100.0	98.6

continued on next page

Table 16 continued

Task	Run	Verified progress (%)					Allocation		Protocol		Tool use (%)	
		<i>S</i>	<i>R</i>	<i>C</i>	<i>P</i>	POLY	U_r	K_r	I_f	η	Valid	Term.
M12	run 1	100.0	100.0	100.0	100.0	100.0	64.5	95.9	99.2	100.0	100.0	98.3
	run 2	70.0	0.0	0.0	21.0	0.0	19.0	83.1	100.0	100.0	100.0	97.9
M22	run 1	100.0	100.0	100.0	100.0	100.0	83.3	60.5	99.0	100.0	100.0	97.6
	run 2	100.0	100.0	0.0	60.0	0.0	64.5	49.5	98.8	88.9	100.0	99.5
M23	run 1	100.0	0.0	12.5	35.0	0.0	31.2	43.8	100.0	100.0	100.0	99.8
	run 2	100.0	100.0	12.5	65.0	0.0	33.0	81.1	100.0	97.6	100.0	99.5
M24	run 1	100.0	100.0	0.0	60.0	0.0	3.0	96.9	98.4	98.2	100.0	99.1
	run 2	100.0	100.0	0.0	60.0	0.0	9.6	97.1	99.0	98.9	100.0	98.3
GPT-6 Astra (10 runs)												
M11	run 1	13.3	0.0	0.0	4.0	0.0	99.3	28.6	100.0	100.0	100.0	95.7
	run 2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	97.4
M12	run 1	100.0	100.0	100.0	100.0	100.0	65.3	45.8	95.0	100.0	99.0	97.1
	run 2	20.0	0.0	100.0	46.0	100.0	41.5	63.2	95.0	100.0	99.0	99.0
M22	run 1	100.0	100.0	100.0	100.0	100.0	100.0	62.9	100.0	100.0	100.0	99.2
	run 2	46.7	0.0	100.0	54.0	100.0	77.4	47.9	100.0	100.0	100.0	97.3
M23	run 1	100.0	0.0	12.5	35.0	0.0	3.5	4.7	97.2	100.0	100.0	98.2
	run 2	100.0	0.0	12.5	35.0	0.0	6.1	6.9	97.2	100.0	99.8	98.4
M24	run 1	100.0	100.0	0.0	60.0	0.0	60.8	99.1	100.0	100.0	100.0	92.7
	run 2	100.0	100.0	0.0	60.0	0.0	49.0	100.0	97.6	100.0	99.8	91.8
Claude Table 5.1 (10 runs)												
M11	run 1	46.7	0.0	0.0	14.0	0.0	100.0	54.1	93.8	100.0	100.0	94.8
	run 2	100.0	100.0	100.0	100.0	100.0	91.9	100.0	100.0	100.0	100.0	89.9
M12	run 1	100.0	100.0	0.0	60.0	0.0	81.2	100.0	70.0	100.0	100.0	85.9
	run 2	100.0	100.0	0.0	60.0	0.0	100.0	100.0	100.0	75.0	100.0	79.6
M22	run 1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	89.7
	run 2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	85.0
M23	run 1	100.0	0.0	12.5	35.0	0.0	27.1	95.7	97.4	100.0	100.0	97.7
	run 2	100.0	0.0	12.5	35.0	0.0	21.8	88.3	97.4	100.0	100.0	98.7
M24	run 1	100.0	100.0	0.0	60.0	0.0	27.0	94.1	97.1	100.0	100.0	93.9
	run 2	100.0	100.0	0.0	60.0	0.0	72.6	94.7	100.0	100.0	100.0	96.1
GPT-5.6 Sol (10 runs)												
M11	run 1	100.0	100.0	0.0	60.0	0.0	46.6	83.4	97.9	100.0	100.0	93.8
	run 2	100.0	100.0	0.0	60.0	0.0	20.9	83.5	97.1	100.0	99.3	94.4
M12	run 1	43.3	0.0	0.0	13.0	0.0	26.7	78.3	99.0	100.0	100.0	96.6
	run 2	50.0	0.0	0.0	15.0	0.0	27.1	37.5	98.3	100.0	99.8	89.7
M22	run 1	13.3	0.0	0.0	4.0	0.0	5.8	58.4	98.4	100.0	100.0	92.8
	run 2	100.0	100.0	100.0	100.0	100.0	90.5	93.0	100.0	100.0	100.0	93.9
M23	run 1	100.0	0.0	12.5	35.0	0.0	20.0	39.7	97.7	100.0	100.0	96.3
	run 2	100.0	0.0	12.5	35.0	0.0	14.5	24.4	97.5	100.0	99.8	95.3
M24	run 1	100.0	100.0	0.0	60.0	0.0	8.3	89.6	100.0	100.0	100.0	95.5
	run 2	100.0	100.0	0.0	60.0	0.0	14.6	40.9	97.9	80.0	100.0	93.7
GPT-5.6 Terra (10 runs)												
M11	run 1	33.3	0.0	0.0	10.0	0.0	16.5	45.2	96.4	100.0	100.0	98.4
	run 2	40.0	0.0	0.0	12.0	0.0	30.9	67.0	92.9	100.0	100.0	93.7
M12	run 1	13.3	0.0	0.0	4.0	0.0	3.2	54.8	98.1	100.0	100.0	93.5
	run 2	100.0	100.0	0.0	60.0	0.0	82.7	62.8	92.9	100.0	99.1	95.3
M22	run 1	13.3	0.0	0.0	4.0	0.0	4.7	5.1	94.7	100.0	100.0	90.7
	run 2	33.3	0.0	0.0	10.0	0.0	7.6	8.3	97.4	97.1	100.0	93.8
M23	run 1	100.0	0.0	12.5	35.0	0.0	50.7	13.4	100.0	100.0	100.0	91.5
	run 2	100.0	100.0	12.5	65.0	0.0	23.2	62.4	98.8	100.0	99.8	94.5
M24	run 1	36.7	0.0	0.0	11.0	0.0	17.1	51.0	95.7	100.0	99.8	96.7
	run 2	26.7	0.0	0.0	8.0	0.0	9.1	11.9	96.2	100.0	100.0	91.7
Claude Sonnet 5 (10 runs)												
M11	run 1	33.3	0.0	0.0	10.0	0.0	1.0	1.0	88.9	80.0	99.6	93.6
	run 2	20.0	0.0	0.0	6.0	0.0	0.3	0.0	93.8	100.0	100.0	95.5

continued on next page

Table 16 continued

Task	Run	Verified progress (%)					Allocation		Protocol		Tool use (%)	
		S	R	C	P	POLY	U_r	K_r	I_f	η	Valid	Term.
M12	run 1	0.0	0.0	0.0	0.0	0.0	39.7	40.3	90.9	100.0	100.0	96.1
	run 2	40.0	0.0	0.0	12.0	0.0	16.8	0.0	94.4	92.9	99.6	92.9
M22	run 1	100.0	100.0	0.0	60.0	0.0	0.7	69.9	94.7	100.0	99.5	96.4
	run 2	33.3	0.0	0.0	10.0	0.0	40.1	19.9	95.7	100.0	100.0	96.4
M23	run 1	63.3	0.0	12.5	24.0	0.0	48.3	77.7	97.4	100.0	99.6	96.4
	run 2	100.0	0.0	12.5	35.0	0.0	64.0	40.1	93.8	100.0	100.0	95.8
M24	run 1	20.0	0.0	0.0	6.0	0.0	1.1	11.3	94.7	100.0	99.8	95.2
	run 2	40.0	0.0	0.0	12.0	0.0	56.5	46.0	94.4	100.0	99.7	97.0
DeepSeek V4 Pro (10 runs)												
M11	run 1	20.0	0.0	0.0	6.0	0.0	16.9	44.5	88.9	100.0	100.0	91.9
	run 2	20.0	0.0	0.0	6.0	0.0	81.1	82.6	80.0	66.7	100.0	88.4
M12	run 1	20.0	0.0	0.0	6.0	0.0	0.3	23.6	80.0	66.7	100.0	89.0
	run 2	20.0	0.0	0.0	6.0	0.0	16.5	26.4	90.0	75.0	98.5	94.8
M22	run 1	0.0	0.0	0.0	0.0	0.0	18.4	81.5	75.0	75.0	99.4	89.6
	run 2	40.0	0.0	0.0	12.0	0.0	95.3	87.7	87.5	100.0	99.4	90.5
M23	run 1	50.0	0.0	12.5	20.0	0.0	2.1	83.1	95.0	100.0	100.0	92.5
	run 2	16.7	0.0	12.5	10.0	0.0	15.5	98.2	100.0	100.0	100.0	76.5
M24	run 1	20.0	0.0	0.0	6.0	0.0	58.0	99.3	83.3	100.0	100.0	87.7
	run 2	20.0	0.0	0.0	6.0	0.0	40.0	38.8	85.7	100.0	98.4	85.2
GLM-5.3 Flash (10 runs)												
M11	run 1	0.0	0.0	0.0	0.0	0.0	83.2	81.8	75.0	100.0	98.6	88.9
	run 2	33.3	0.0	0.0	10.0	0.0	9.6	64.8	62.5	100.0	97.0	78.1
M12	run 1	0.0	0.0	0.0	0.0	0.0	0.0	N/A	0.0	N/A	96.6	87.8
	run 2	13.3	0.0	0.0	4.0	0.0	63.7	56.2	66.7	66.7	98.5	84.4
M22	run 1	0.0	0.0	0.0	0.0	0.0	0.0	N/A	N/A	N/A	100.0	75.4
	run 2	0.0	0.0	0.0	0.0	0.0	99.9	45.2	61.1	N/A	98.3	90.3
M23	run 1	0.0	0.0	0.0	0.0	0.0	0.0	N/A	33.3	N/A	94.9	86.5
	run 2	0.0	0.0	0.0	0.0	0.0	0.0	48.1	46.3	33.3	92.8	79.2
M24	run 1	40.0	0.0	0.0	12.0	0.0	78.1	87.1	72.2	75.0	97.3	78.7
	run 2	0.0	0.0	0.0	0.0	0.0	0.0	N/A	61.1	N/A	98.4	70.4

H.9 METRIC INTERPRETATION AND SENSITIVITY

Sensitivity to route labels and cost attribution. Using the retrospective ledger for every defined token K_r changes Astra’s mean from 55.9 to 58.0, Fable’s from 92.7 to 97.3, and GPT-5.6 Sol’s from 62.9 to 57.8. The means for the other systems remain unchanged at the reported precision (Table 10). The pooled K_r – R and K_r – C correlations become 0.49 and 0.10, compared with 0.55 and 0.17 under the paper’s mixed-source rule. Under the earlier correct-route rule (Appendix H), the model averages were Opus 76.2, GPT-6 Sol 62.1, Astra 59.4, and Fable 57.0, the pooled K_r – R and K_r – C correlations were 0.44 and -0.02 , and among the 34 runs with R the K_r – C and U_r – C correlations were -0.29 and 0.64 . This alternative classification still does not mean that wrong routes are mathematically impossible.

Two runs produce a verified polynomial without earning a route-identification score. That is not a contradiction: the two forms of credit have different acceptance rules. The resource efficiency metric U_r counts only finalized artifact costs. When some work is unattributed, multiplying that fraction by all run tokens does not give the credited token cost. Figure 1 therefore reports total tokens and U_r separately.

Task effects and ranking uncertainty. Correlations pooled across tasks do not establish task-independent effects. To examine task differences, we use pairwise complete records, rank each metric with average ranks for ties, subtract its mean rank within each task, and correlate the residuals. The resulting correlations with C are 0.35 for K_r and 0.52 for U_r , compared with pooled Spearman correlations with C of 0.17 and 0.42. For the 50 runs of the five systems in Section 6, the within-task correlation between K_r and C is 0.25 (95% bootstrap CI $[-0.05, 0.51]$), against a pooled 0.01. This

adjustment does not remove system effects, shared score definitions, or selection into the $R > 0$ subset. The pooled result should therefore be read as describing the particular mixture of tasks in this evaluation.

We also recompute means after leaving out each task in turn. Omitting M_{23} changes Sol’s and Astra’s scores from $P = 62.1$ and 59.4 to 65.1 and 65.5 , respectively. Opus keeps the highest observed mean in every leave-one-task-out comparison. Table 11 gives both run-level and task-cluster bootstrap intervals from 10,000 resamples. We do not use interval overlap as a hypothesis test. With only five related tasks, neither resampling method establishes a general ranking across the 200 listed problems, and we make no statistical-significance claim.

Scores of the reference runs. Under the generic construction schedule all ten runs have $C = 40$, where the two M_{22} polynomials are $T = 1$ specializations of the runs’ regular families. With assigned $S = R = 30$, the values in Table 3 follow: $C = 100$ on the percentage scale, $P = 100$, and verified-polynomial coverage of 100%.

The step-based $U_r = 82.8$ counts screening and every step on the finished route as useful. Without an artifact graph, this is an optimistic proxy. It is neither a certified bound on the benchmark’s metric nor a directly comparable measurement.

What the reproduction code checks. The revision code checks capture hashes, matches run identifiers, recalculates statistics, and regenerates tables and figures from the supplied snapshot. It does not rerun Magma, the description-fidelity judge, or any agent. Terminal scores and retrospective route counts remain the preserved source data. The capture-only attachment lacks the full terminal artifact graphs needed to reconstruct every U_r and retrospective K_r numerator. The released inventory distinguishes these terminal records from the independently checked capture records.

I EVIDENCE FOR THE STOPPING ANALYSIS

Source and scope. The comparisons below use the same eligible task-runs in `igpbench_paper_results.csv`. For K_r , they use the evaluator’s value when available and the census value otherwise. They add no model runs or causal interventions. The editorial bundle’s `verify_stopping_claims.py` reproduces all transformations.

Separating construction from total score. Table 17 shows the three leading systems’ raw point contributions alongside their normalized construction scores. Opus and Astra each contribute 21 construction points to P . Their total-score gap is exactly

$$81.0 - 59.4 = (30.0 - 23.4) + (30 - 15) + (21 - 21) = 21.6.$$

Astra contributes 12 more construction points than Sol, but earns 5.7 fewer screening points and 9 fewer route-identification points. Its total is therefore 2.7 points lower. This calculation breaks down the recorded scores; it does not identify a causal effect.

Table 17: **Construction and total research scores give different rankings.** S , R , and C are raw contributions to P , with maxima 30, 30, and 40. The final column shows C on the 0–100 scale used in Table 3.

System	S	R	C	P	Normalized C
Claude Opus 5	30.0	30.0	21.0	81.0	52.5
GPT-6 Sol	29.1	24.0	9.0	62.1	22.5
GPT-6 Astra	23.4	15.0	21.0	59.4	52.5

An obstruction recorded before reconstruction. For `astra_b1_M23`, the supplied qualitative archive `failures_long.csv` records that artifact 4 identifies both the inversion exchange $23A \leftrightarrow 23B$ and the branch-class obstruction for $(2A, 4A, 23A)$. Later annotations describe artifacts 15–25 reconstructing and certifying the corresponding cover over a quartic field. The archive also describes a second construction on $(2A, 8A, 8A)$ over a degree-28 coefficient field, followed by a passport obstruction.

The main text uses the first route because the annotation explicitly places its obstruction before reconstruction. It does not infer a general obstruction merely because an all-real branch configuration is absent. These claims are based on the supplied trace annotations, not on new independent replays of the proofs. Terminal scores and budgets come from the separate per-run CSV.