graphics cafe seminar · november 2026

Agents for Reasoning and Verification in Mathematics Research

Three systems: Rethlas, Danus, Pharos, and OSVerify



Shurui Liu

Stanford University

Research mathematics

Competitions (IMO, Putnam) are like LeetCode or Chess. Research is a different regime.

Open scope

No fixed set of moves, and no guarantee that a solution exists.

Long horizon

Hundreds pages; wrong partial results or routes may waste compute.

Sparse, noisy reward

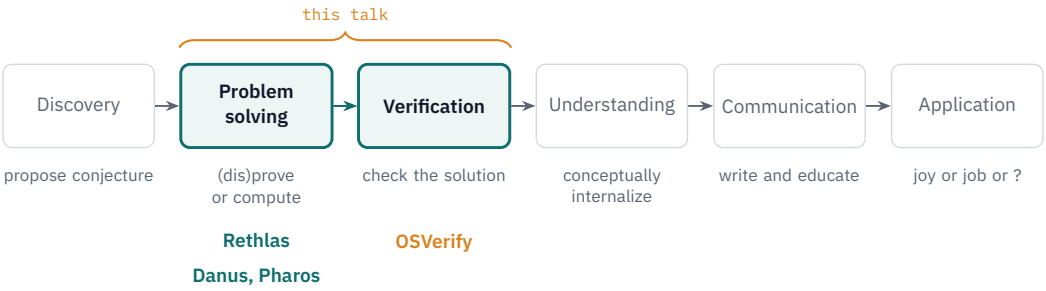
Hard to determine significance of partial results, or obtain a correct solution to a difficult problem, or confirm a solution is fully correct.

Evolving, noisy Data

arXiv and journals contain errors; the frontier is not in the training data. Need test-time learning.

Techniques could transfer to other tasks; research math is a cleaner “benchmark”.

Math research loop



This talk: **problem solving** (Rethlas, Danus, Pharos) and **verification** (OSVerify).

Three systems

1 Rethlas
One agent proves, a fresh one referees, in a loop.
arXiv:2604.03789

2 Danus
A team of agents over a verifier-gated fact graph DAG.
arXiv:2607.06447

3 Pharos
Improved planning, writing, context management, etc. upon Danus.
frenzymath/Pharos

4 OSVerify
One-sided verification: propose errors, then validate.
arXiv:2610.18089

TODAY · ONE SYSTEM AT A TIME: RESULTS, THEN DESIGN, THEN CASE STUDIES

I. Rethlas
a prover and a referee

II. Danus and Pharos
a research group of agents

III. OSVerify
scaling the referee

IV. Lessons
what transfers

Three systems

1 Rethlas
One agent proves, a fresh one referees, in a loop.
arXiv:2604.03789

2 Danus
A team of agents over a verifier-gated fact graph DAG.
arXiv:2607.06447

3 Pharos
Improved planning, writing, context management, etc. upon Danus.
frenzymath/Pharos

4 OSVerify
One-sided verification: propose errors, then validate.
arXiv:2610.18089

Mathematicians cannot count: the "top four" journals of mathematics are *Annals*, *Acta*, *Inventiones*, *JAMS*, and *Publ. IHÉS*.

TODAY · ONE SYSTEM AT A TIME: RESULTS, THEN DESIGN, THEN CASE STUDIES

I. Rethlas
a prover and a referee

II. Danus and Pharos
a research group of agents

III. OSVerify
scaling the referee

IV. Lessons
what transfers

Design principles

FROM MATHEMATICS

Mimic the human workflow

How to distill experts' workflow and use it to design harness?

Separation of roles

How to meaningfully decompose the task into separated ones (without over-harness)?

FROM COMPUTER SCIENCE

Context engineering

How to provide the context exactly needed to fullfil the task, neither lack nor excess?

Scaling test-time compute

How to meaningfully scale the compute with no waste?

Base assumption: the foundation model is strong enough for meaningful search, and strong enough for verification with reasonable error rate.

part I

Rethlas

A generate–verify loop that persistently attacks one problem

Rethlas H. Ju*, G. Gao*, J. Jiang*, B. Wu*, Z. Sun*, S. Liu*, . . . , L. Xiao, R. Liu, B. Dai, B. Dong

*equal contribution

01

Same model, with and without the loop

Problem	Subject	Importance	Model	Model alone	With Rethlas
Rational cocharacters	algebraic groups	Lemma	GPT-5.5-Pro	✗ wrong proof	✓ correct proof, 44 min
KL polynomials	matroids	Publishable	GPT-5.6 Sol	✗ incomplete	✓ counterexample, 4 h 59 min
Automorphic commutator relation	relative geometric Langlands	Publishable	GPT-6-Astra	✗ incomplete	✓ proof, 1h

45 min
counterexample to an open question of Anderson (2014)

19,448
lines of Lean 4 certifying it, via Archon

For each case, only the statement was provided.

Results across fields

Area	Results	System
Commutative algebra	Anderson's question; eight CFFG conjectures; Boij–Söderberg questions (Erman–Sam)	Rethlas
Algebraic groups	rational conjugacy of cocharacters	Rethlas
p-adic Hodge theory	refined lift-independence; a genus-separation counterexample	Rethlas
Analysis, probability	Brezis problems; injectivity of phase retrieval	Rethlas
Combinatorics	matroid KL polynomials are not real-rooted; tangent classes of matroids; Matryoshka asymptotics	Rethlas, Danus
Algebraic geometry	foliations (bend-and-break, Shokurov index); Cartier indices; rational curves via jets	Danus, Pharos
Number theory, Langlands	Manin constant; automorphic commutator relation	Rethlas + OSVerify

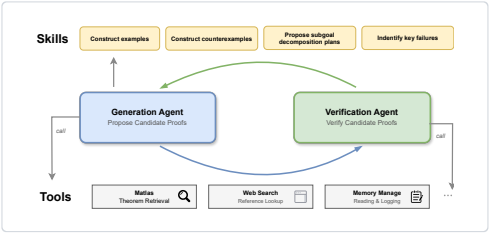
frenzymath/

Rethlas_results

A partial, growing list, checked by human experts before release.

Rethlas in one slide

- A **generation agent** (Codex or Claude Code) writes an informal proof *blueprint*, with skills, a search tool, and a notebook.
- A **verification agent** referees it: a fresh process per submission returns a strict JSON verdict and repair hints.
- Repeat until acceptance: zero critical errors *and* zero gaps.



Principles: **mimic the human workflow**, and **engineer the context**; above all, separate the author from the referee.

Skills, tools, and memory

Skills = habits

Toy examples, counterexamples, literature search, decomposition plans, direct and recursive proving, identifying key failures.

Loop: *assess, choose a skill, act*. Competing plans run as parallel subagents that share their failures.

Tool = the library

Matlas (`matlas.ai`): semantic search over ~13.6M theorem statements from arXiv, with dependencies unfolded.

Think, then look it up: odd iterations block search.

Memory = the notebook

Raw logs are scratch paper. Structured notes are append-only JSONL channels (`toy_examples, failed_paths, . . .`), retrieved by BM25.

Failed paths are mandatory

Every dead end stays queryable.

The author is not the referee

- Authors believe their own proofs. So do models.
- The math community's answer is the **referee**: someone with no stake in the proof being true.
- **Generator: continuous context.** Intuition, partial progress, and dead ends persist.
- **Verifier: fresh context.** A new process per submission sees the blueprint, not how it was produced.

Separate roles also get separate tools and skills.

Smallest deduction steps

Check each step for validity, applicability, and missing hypotheses.

Reference correctness

Does the cited theorem exist, and does its *statement* give what this proof needs, under these hypotheses?

An open problem of D. D. Anderson

question (anderson, 2014)

Does weak quasi-completeness imply quasi-completeness for Noetherian local rings?

An implication; the answer is an explicit counterexample.

45 min
Rethlas: counterexample, no proof hints

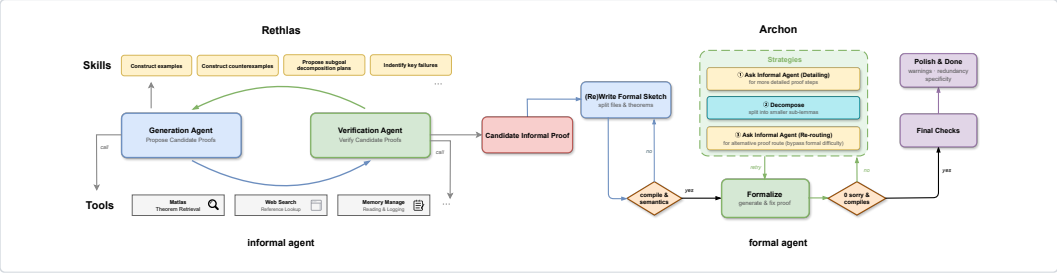
19,448
lines of Lean 4, 42 files

~80 h
Archon agent runtime

26 lines
human-reviewed specification

- **Cross-field retrieval.** The key ingredient, Jensen's 2006 theorem on completions of local UFDs, came through Matlas from the generic-formal-fiber literature. (Later: OpenAI's unit distance problem uses algebraic number theory to solve discrete geometry problem)
- **A small trusted base.** Comparator checks that the Lean project proves exactly the 26-line specification, with only the standard axioms. Humans review 26 lines, not 19,448.

From informal to formal: Rethlas–Archon



Rethlas's generate–verify loop produces the informal proof; Archon turns it into a kernel-checked Lean 4 project.

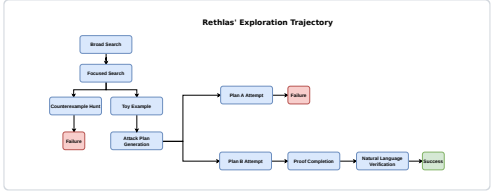
Rational conjugacy classes of cocharacters

question (suggested by brian conrad)

G a connected smooth affine group over a field k, $\mathbb{X}_*(G) = \mathrm{Hom}_{k\text{-gp}}(\mathrm{GL}_1, G)$. For K/k, is $\mathbb{X}_*(G)/G(k) \to \mathbb{X}_*(G_K)/G(K)$ injective?

Known for split reductive, reductive, or perfect k; the general case was not known to experts.

- ✗ GPT-5.5 Pro (web): a wrong proof, no meaningful progress.
- ✓ Rethlas (GPT-5.5 xhigh): a correct proof in 44 minutes, through the “quasi-reductive” case.
- › One “mistake”, a **citation substitution**: the needed result is in a copyrighted book the agent could not retrieve; so the agent cited a survey.



Kazhdan–Lusztig polynomials of matroids

conjectures

$P_M(t)$ is real-rooted (Gedeon–Proudfoot–Young, 2017) and log-concave (Elias–Proudfoot–Wakefield, 2016); evidence from uniform, q-niform, fan, wheel, whirl matroids.

Prompt: the definition of P_M, and "prove or disprove that all zeros of $P_M(t)$ lie on the negative real axis."

✔ Rethlas (GPT-5.6 Sol, max): explicit counterexample in 4 h 59 min, by deleting points from a finite projective geometry. The method itself is interesting.

✖ GPT-5.6 Sol Ultra (web), two prompts: incomplete.

Humans understand the proof

Cheng and Liu developed the method (Theorem 1), gave a geometric interpretation (Section 3), and proved that P_M need not be **unimodal**. This refutes both conjectures at once.

CS: model missed the highlights

ChatGPT failed to generalize the construction to a method or develop a geometric explanation or disprove another conjecture. (This happened to unit distance problem, too.)

R. Cheng, S. Liu, *Kazhdan–Lusztig polynomials of matroids need not be unimodal*, arXiv:2607.24186.

Why one loop is not enough

Search is a single DFS

One hypothesis at a time; no decomposition of the problem and no mixture of expertise.

Context polluted

Every iteration reloads the `blueprint.md`, the verified partial results *with their proofs*, and the notes, even when only statements are needed.

Parallel edits collide

Two agents cannot safely improve the same Markdown document, so the loop cannot simply be parallelized.

What is needed: a modular deliverable, a single shared truth, and rules about who may write what. In other words, orchestration.

part II

Danus and Pharos

An organization of agents over a verifier-gated fact graph, rebuilt as Pharos for stronger models

Danus J. Liu, G. Gao, Z. Sun, B. Wu, S. Liu, . . . , B. Dong

Pharos G. Gao*, Z. Sun*, B. Wu*, S. Liu*, J. Jiang, B. Dong

*equal contribution

02

Tangent classes of matroids: an ablation

problem (main results of a paper by ronnie cheng)

Construct a tangent class $T_{M,\mathcal{G}}$ in the combinatorial K-ring of any loopless matroid, extending the tangent bundle of the wonderful compactification, with HRR and Chern-number properties. Cheng's own preprint was withheld from all systems.

Same verbatim prompt		Outcome
GPT-5.5-pro (web)	model	✗ no solution
Rethlas, 3 runs	loop	✗ 0/3 passed the verifier
Danus	organization	✓ verified solution

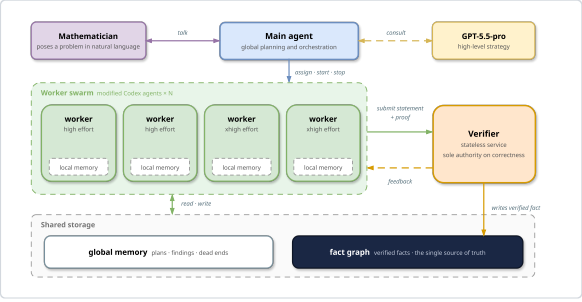


3,157
verified facts

54
deepest chain

R. Cheng, S. Liu, G. Gao, *Tangent classes of matroids and wonderful compactifications*, arXiv:2607.05835.

A research group of agents



- **Main agent** = group leader (Claude Code, Opus 4.8): plans, assigns, monitors, reports. No mathematics, no submission tool.
- **Workers** (Codex, GPT-5.5): Rethlas generation agents, one claim at a time; low effort for breadth, high for depth.
- **Senior expert** (GPT-5.5-pro): consulted every 1–2 h on distilled state.
- **Verifier**: fresh per claim; the only way into the fact graph.

Spend the most expensive intelligence at the lowest frequency.

Separation of powers, enforced by tools

At scale, separation of roles becomes separation of powers. It is enforced by the MCP tool table, not by instructions.

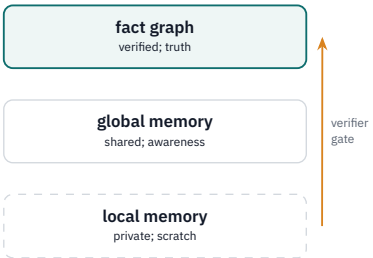
	main	worker	verifier
read facts	✓	✓	✓
write memory	✓	✓	✗
`fact_submit`	✗	✓	✗
decide truth	✗	✗	✓

The human expert and the main agent only steer direction and strategy. Worker and verifier skills are inherited from Rethlas essentially unchanged; what changed is the orchestration.

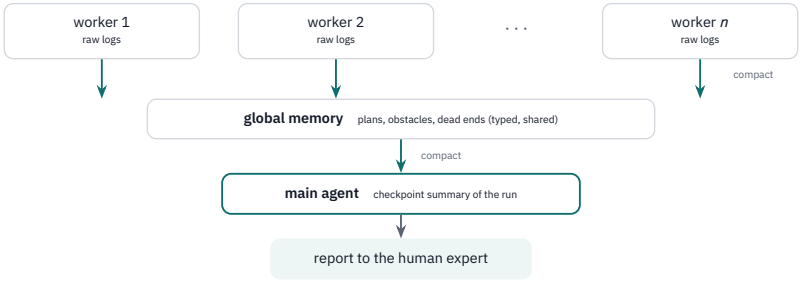
Three tier memory

Mimicing research group and context engineering:

- **Local memory**: each worker's own scratch paper.
- **Global memory**: plans, examples, counterexamples, dead ends, obstacles. Shared awareness, explicitly *not* truth.
- **Fact graph** (a DAG): node = statement + checked proof; edge = logical dependency.
  1. **decomposes** the problems into small steps;
  2. provides **communication** channel of verified facts among workers;
  3. **cleaner context**: A worker loads only the facts it needs, by statement, and reads a proof only when its technique is needed.



Compaction at every boundary



What any single agent reads stays bounded, while what the system knows keeps growing.

A worker starting its tenth task inherits the lessons of the first nine by reading a summary, not the full logs.

Tangent classes: inside the run

- **7 workers** (3 `xhigh`, 4 `high`); ~5 days, ~50 h active; 13 GPT-5.5-pro consultations.
- **3,157 verified facts**, 8,616 edges; chains up to **54** deep; **499** facts in the theorem's closure.
- Global memory: 636 proof attempts, 151 counterexamples, 25 dead ends.
- **Why Rethlas failed**: it conflated nearby notions (realizable matroid, wonderful model, toric variety; integral vs. rational K-theory).

What exploration costs, and buys

The facts established are about 6× more than needed in the surviving proof.

Takeaway?

Scaling compute and decomposing task can resolve subtlety of the math task. (Later GPT-6-Astra can do geometric Langlands problems.)

Scaling in width and depth

↔ Width

Modular deliverables (nodes of the fact graph); the main agent diversifies directions.

↕ Depth

Per-fact verification and the memory hierarchy extend the reasoning horizon beyond one context window: chains of 54 facts.

$$0.99^{499} < 1\%$$

probability that a 499-fact closure is sound at a 1% per-fact error rate

Verifier precision is load-bearing

A worker trusts every existing fact without re-deriving it. With a fallible verifier, errors compound along the chains and long-horizon exploration collapses; with a reliable one, accumulation is safe.

The organization deceives itself

FAILURE MODES

- Progress reports drift optimistic, and each one summarizes the last.
- The writer introduces mistakes when translating DAG into a paper.
- Effort feels like progress: seven agents are always busy, but is the work done?
- One false fact poisons everything downstream of it.

MECHANISMS

- Summaries under **prohibitions**: no progress estimates, no “almost done”.
- **The paper must pass the verifier.**
- **Citations are data**, recorded at proof time.
- **Result-gated termination**: stop only when the target is a verified fact.
- **Revocation** removes a wrong fact and all its dependents.

From Danus to Pharos

	Danus (July 2026)	**Pharos v3.1** (Sept. 2026)
Main agent	Claude Code, Opus 4.8	resident Codex session, `ultra` effort
Strategy	GPT-5.5-pro consulted every 1–2 h; reply broadcast to all workers	**consultant and broadcast removed;** one `TASK.md` briefing per worker
Planning	periodic summaries	prior-work survey; **route board**; 4-hour checkpoints graded by the expert's criteria
Human input	ad hoc	recorded verbatim; a "do not cite" list
Workers	per-round rhythm	"a fact is a lemma": no quota; multi-round work on one result
Verifier	one shared service	per project; probes "exhaustive" case claims; demands version and hypotheses of cited papers
Writing	one writing loop	outline ⟶ per-section writer–verifier pairs ⟶ whole-paper verifier + distillation of human expert writing style
Operations	shared homes; full scans of memory; -	per-actor homes; incremental index for memory search; compute caps; monitor

Rational curves via jets (for reproving Hartshorne conjecture)

question (du-guo-xie, appendix f)

For smooth projective X in characteristic zero and a curve $f : C \to X$, does $-K_X \cdot f_*[C] > 0$ alone give a rational curve through every point of $f(C)$, under a strict degree bound?

19 h 57 min
to the verified target, no human intervention

273
verified facts; 1 revoked and re-proved

11 h 38 min
with expert hints at launch; same proof

>300k
lines of Lean 4, within a week

B. Dong, G. Gao, B. Guo, Z. Sun, B. Wu, S.-Y. Xie, arXiv:2609.28465, Appendix B.

The route board

Route board — Example: the Miyaoka–Mori run, hour 18

Route	Key idea	Origin	Status	Record
R3-6	Weighted jets instead of Frobenius	new	Active	Tested on special models
R3	Ruled surface; seed curve as a section	known	Active	Fact A at 1:27
R6	Foliated tubes, then adjoint contraction	known	Active	Fact B at 5:42, uses fact A
R2	Finite covers and cuts, glued back	known	Active	Four sub-routes so far
R1	Ricci reserve, area-decreasing disks	known	Stopped	Stalled; experience kept

Known routes first

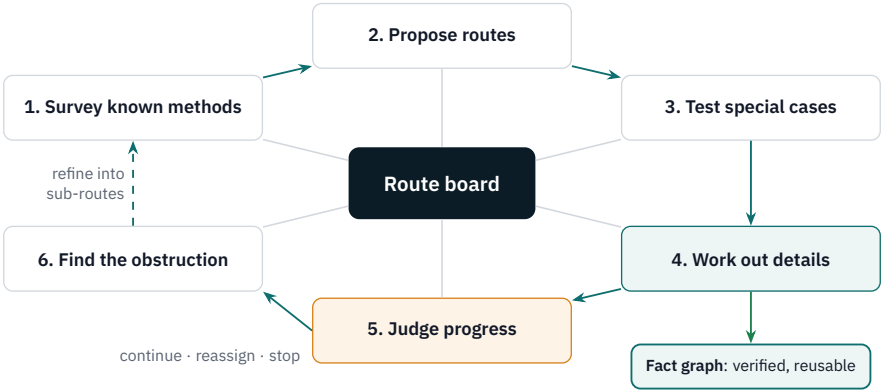
Every technique applied in the literature is tried before new routes are invented.

One board for the whole run

Where each route stands, and what to try next.

Adapted from G. Gao's talk; Dong–Gao–Guo–Sun–Wu–Xie, arXiv:2609.28465, Appendix B.

How the route board is updated

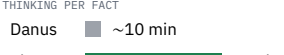


Adapted from G. Gao's talk; Dong–Gao–Guo–Sun–Wu–Xie, arXiv:2609.28465, Appendix B.

Pharos: other improvements

Deeper facts

Suited to GPT-6 Astra.



Helper agents

Digest literature, scout routes, assess progress.

Never part of a proof.

Computation

Workers run code and numerical experiments.

Working with experts

Follow a run, steer it, get **shorter and clearer papers**.

Efficiency

Lossless context compression: lower cost, higher throughput.

Adapted from G. Gao's talk; Dong–Gao–Guo–Sun–Wu–Xie, arXiv:2609.28465, Appendix B.

part III

OSVerify

Verification is now the expensive step

OSVerify F. Nie*, H. Ye*, S. Liu, Z. Wang, S. Ermon, J. Zou

*equal contribution

03

OSVerify as a referee

OpenAI preprints

11 recent OpenAI preprints on central problems. Mistakes found in each, overlooked by OpenAI's internal model, confirmed by experts.

Human papers

Issues confirmed by the authors; same backbone, reviewing directly vs. inside OSVerify:

	direct	OSVerify
paper1	11	**45**
paper2	3	**88**

Also found mistakes in another several published papers on top journals.

Used by a mathematician

An AI-assisted paper hexagon:2611.00006v1 on Kamnitzer's ICM 2022 conjecture.

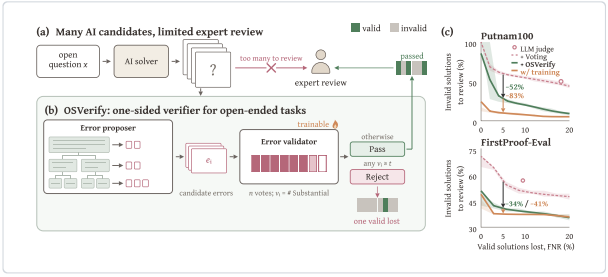
Round 1: GPT-6-Pro drafted, Codex reviewed; OSVerify caught a major error.

Round 2: GPT-6-Pro repaired; three ChatGPT reviews and Codex found nothing; OSVerify caught another proof-breaking error and suggested corrections.

F. Nie, H. Ye, S. Liu, Z. Wang, S. Ermon, J. Zou, arXiv:2610.18089 (Appendix: representative issues); S. Liu, arXiv:2610.05781; hexagonmath.org/2611.00006.

Verification is the bottleneck

- **Real life application**: generation now outpaces review, so we need to reduce the human effort spent refereeing papers.
- **Agentic system application**: once generation scales, a weak verifier will ruin the run. So scale test-time compute on the verifier too.
- **Objective** (Neyman–Pearson): maximize the rejection rate of invalid solutions (TNR) subject to a small false-rejection rate of valid ones (FNR $\leq \tau$).



Finding an error vs. confirming one

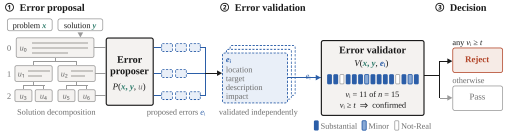
Mimic the referee: first suspect a step, then check it.

Proposer $\mathcal{G}$: decompose the solution hierarchically; propose errors at every granularity. Goal: coverage.

Validator C: is an error Substantial, Minor, or Not-Real? Repeated votes. Goal: precision.

0.68 ⟶ 0.90

AUROC: judging solutions vs. judging proposed errors; same model, FirstProof-Eval



$s(x, y) = \max_{e \in \mathcal{G}(x,y)} \Pr[C(x, y, e) = \text{Substantial}]$; reject iff $s \geq t$, with t set on valid solutions so that FNR $\leq \tau$.

The Manin constant

theorem (chi–liu–nie, arxiv:2610.25672)

The Manin constant conjecture, open for over 50 years and central to the study of the BSD conjecture, is true; so are Stevens's Conjectures I and II.

Base models, expert prompting

ChatGPT-6 Pro, Claude Fable 5.1, Codex 6-Astra first proved only $c_\varphi \mid \deg \varphi$. Reaching $c_E \mid 2$, then $c_E = 1$, took many rounds of prompts from the authors.

Review

12 independent checks (Codex 6-Astra, Claude Opus 5.5) passed. OSVerify found one mistake: the claim $L_E \in \mathbb{Q}[[w]]$ for every $E/\mathbb{Q}_p$. It suggested a fix for the three affected places.

Harness

Rethlas (GPT-6 Astra) with OSVerify, given only the statement, reproduced essentially the same proof, rejecting ineffective strategies automatically.

Key idea: modularity of the normalized lift via Calegari–Dimitrov–Tang. The logs show no access to the closest prior use (Yin, 2026), but pretraining exposure cannot be excluded.

A stricter referee, measured

TNR (%) at FNR ≤ 5%, backbone `gpt-oss-120b`

	Putnam	H2V	FirstProof
LLM judge	0	54.4	0
Voting (16 samples)	38.4	69.2	35.9
OSVerify	70.4	**84.7**	58.0
+ RL-trained validator	**89.6**	83.2	**62.3**

TNR: share of invalid solutions rejected; FNR: share of valid ones rejected. Reference solutions and Lean withheld; weak backbones on purpose; FirstProof-Eval (research level) unseen in training.

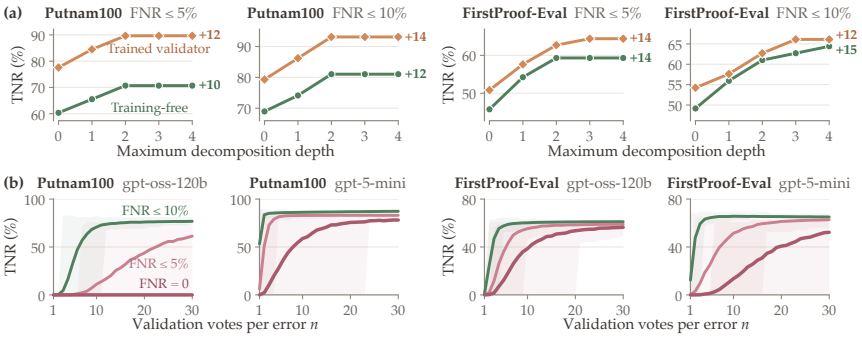
Benchmarks

Vs. voting at the same FNR: 46% fewer invalid solutions reach a human; 57% fewer with RL.

The Manin constant

Found the error that 12 AI reviews had passed (OSVerify on GPT-5.6 Sol).

Scaling compute on verification



(a) Proposer: deeper decomposition raises TNR, with most of the gain by depth two. **(b) Validator:** more votes raise TNR; stricter FNR constraints need more votes. **Matched budgets:** OSVerify beats voting at every token budget, and RL on the validator reaches the same TNR with ~4× less compute.

part IV

Lessons and furture directions

04

Design philosophy

Discovery → **Problem solving** → **Verification** → *Understanding* → Communication → Application.

MATHEMATICAL HEURISTICS

Mimic the human workflow

Habits become skills, resources become tools, the notebook becomes memory (Rethlas); at scale, mimic a research group (Danus, Pharos).

Separation of roles

The author is not the referee (Rethlas); separation of powers through capabilities (Danus, Pharos); suspecting an error is separate from confirming it (OSVerify).

CS HEURISTICS

Context engineering

Fresh context for judgment; bounded reads for workers; compaction at every boundary; one source of truth; one error per validation call.

Scaling test-time compute

Iteration (Rethlas); width and depth (Danus, Pharos); verification (OSVerify).

// future work

Simplifying the harness as models improve

Components that compensate for weaknesses

Skills, retrieval, tools.

Components that helps context management

Separation of roles (fresh verifier), memory, etc.

Open directions

Evaluating agentic systems

How to evaluate/benchmark agentic systems more rigorously?

From verification to understanding

Ongoing work: AI that helps humans understand mathematics.

Creativity and theory building

Systems work within the “convex closure of known techniques” (not well-defined!); a strict verifier favors small steps.

Self-improvement of harness

How to unlock self-improvement capabilities of harness?

References

SYSTEMS

H. Ju, G. Gao, J. Jiang, B. Wu, Z. Sun, S. Liu, et al. *Automated Conjecture Resolution with Formal Verification*. arXiv:2604.03789.

J. Liu, G. Gao, Z. Sun, B. Wu, S. Liu, et al. *Danus: Orchestrating Mathematical Reasoning Agents with Fact-Graph Memory*. arXiv:2607.06447.

F. Nie, H. Ye, S. Liu, Z. Wang, S. Ermon, J. Zou. *One-Sided Verification Under Limited Expert Review*. arXiv:2610.18089.

Code: `github.com/frenzymath/{Rethlas, Archon, Pharos}`

MATHEMATICS

R. Cheng, S. Liu, G. Gao. *Tangent classes of matroids and wonderful compactifications*. arXiv:2607.05835.

R. Cheng, S. Liu. *Kazhdan–Lusztig polynomials of matroids need not be unimodal*. arXiv:2607.24186.

B. Dong, G. Gao, B. Guo, Z. Sun, B. Wu, S.-Y. Xie. *Constructing rational curves via jets on projective varieties with non-nef canonical bundle*. arXiv:2609.28465.

Z. Chi, S. Liu, P. Nie. *On the Manin constant*. arXiv:2610.25672.

S. Liu. *Automorphic commutator relations in relative geometric Langlands*. arXiv:2610.05781.

Thank you

Shurui Liu · Stanford University