Does the Harness Still Matter?

Guoxiong Gao

AI4M Team, BICMR Peking University

Stanford University

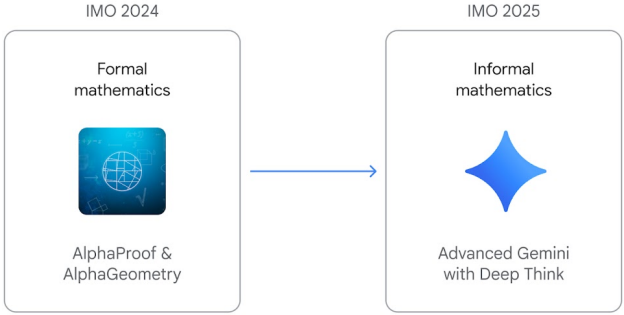
October 2026

Outline

1. Recent progress in AI for mathematics
2. Where to improve: the model or the harness?
3. Remaining gaps, and how Rethlas, Danus and Pharos address them

2025: general language models reach IMO gold level

OpenAI Google DeepMind July 2025



- **OpenAI and Gemini both scored 35/42**: IMO gold-medal level.
- General-purpose models produced **natural-language proofs**.

First Proof: lemmas from frontier research

OpenAI Google DeepMind February–March 2026

First Proof?

OpenAI

February 13, 2026

Abstract

Here we present the solution attempts our models found for the ten `https://1stproof.org/` tasks posted on February 5th, 2026. All presented attempts were generated and typeset by our models.

Ten concrete propositions arising in current mathematical research.

- OpenAI: **5/10** reported likely correct after review.
- Gemini / Aletheia: **6/10** accepted by a majority of assessors.

Unit distances: a counterexample from number theory

OpenAI May 2026

Planar Point Sets with Many Unit Distances

OpenAI

Abstract

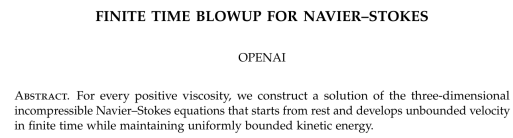
For a finite planar set P, let $\nu(P)$ be the number of unordered unit-distance pairs in P, and let $\nu(n)$ be the maximum of $\nu(P)$ over all n-point planar sets. We prove that, for some fixed $\delta > 0$, one has $\nu(n) \geq n^{1+\delta}$ for infinitely many n. This disproves the well-known unit distance conjecture from [Erd46].

Let $u(n)$ be the maximum number of unit-distance pairs among n planar points.

$$\underbrace{u(n) = n^{1+o(1)}}_{\text{Conjectured}} \qquad \longrightarrow \qquad \underbrace{u(n) \geq n^{1+\delta}}_{\text{Counterexample}}$$

- Fixed $\delta > 0$, for infinitely many n.
- The construction brings **algebraic number theory** into discrete geometry.

September 2026



FINITE TIME BLOWUP FOR NAVIER–STOKES

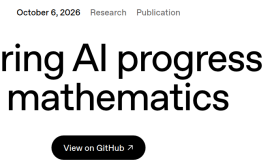
OPENAI

ABSTRACT. For every positive viscosity, we construct a solution of the three-dimensional incompressible Navier–Stokes equations that starts from rest and develops unbounded velocity in finite time while maintaining uniformly bounded kinetic energy.

Navier–Stokes: finite-time blowup

- 3D flow with smooth forcing blows up from rest.
- Public proof and Lean certificate.

October 2026



October 6, 2026 Research Publication

Sharing AI progress in mathematics

View on GitHub ↗

722 manuscripts from one model

- 372 result families in 17 fields.
- No zeros of $\zeta(s)$ in $\Re s > 7/8$.

Model improvements change how we work

Late 2025: Opus 4.5 (November), GPT-5.2 (December)

- In Claude Code / Codex, models could execute complete tasks systematically.
- Early-PhD-level execution: less need to inspect each implementation detail.

2026: Fable 5 (June 9), Astra (September 3)

- Stronger reasoning and autonomous execution.
- Routine tasks need less explicit steering of working preferences.

Next: Bel? (December?)

- The internal model behind the 722 manuscripts.

2026: frontier models with mature harnesses

Claude Fable 5 June
GPT-6 Astra September

strongest models

+

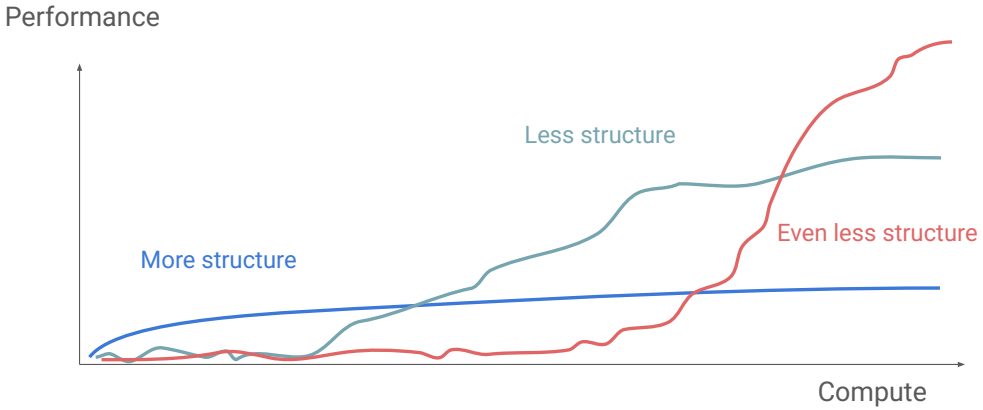
Claude Code
Codex

general harnesses

`/goal`	keep going
`max`	deeper reasoning
`ultra`	more subagents

The bitter lesson for agent design

Hyung Won Chung Stanford CS25 April 2024



More general methods: slower starts, greater gains at scale.

Two questions

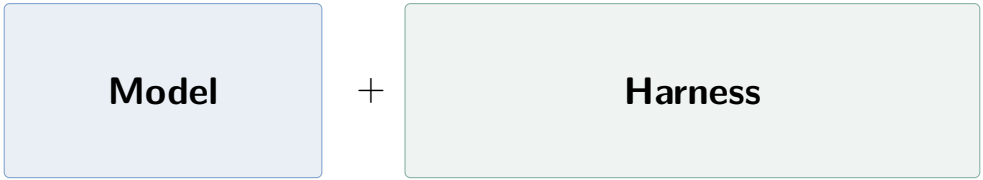
1. What can we do to improve AI's mathematical ability?

2. What is AI still missing, and what will remain as models keep improving?

Outline

1. Recent progress in AI for mathematics
2. Where to improve: the model or the harness?
3. Remaining gaps, and how Rethlas, Danus and Pharos address them

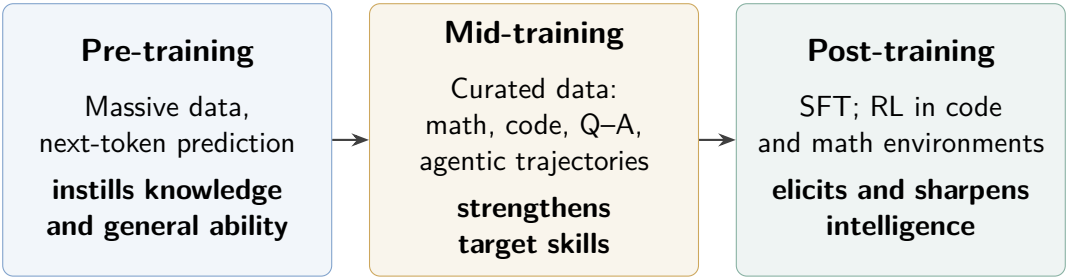
Agent = Model + Harness



"If you're not the model, you're the harness."

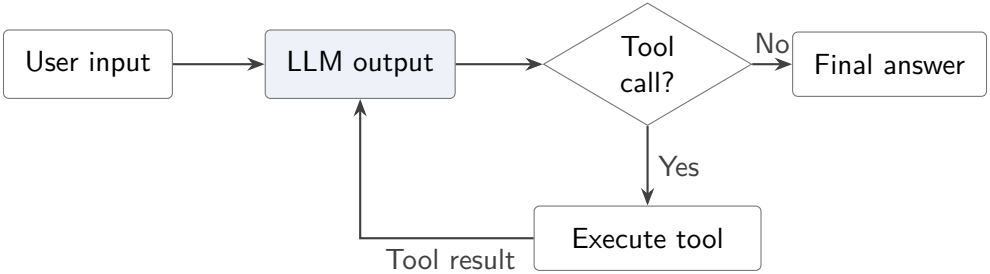
Vivek Trivedy, LangChain, 2026

What can we do on the model?



Academic lab (<100 GPUs)	partial training of <30B models (mid / post)
Commercial models	hundreds of billions to trillions of parameters

An agent is a tool-use loop

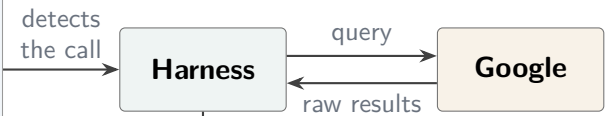


How an LLM calls a tool

LLM output

I should check the literature first.

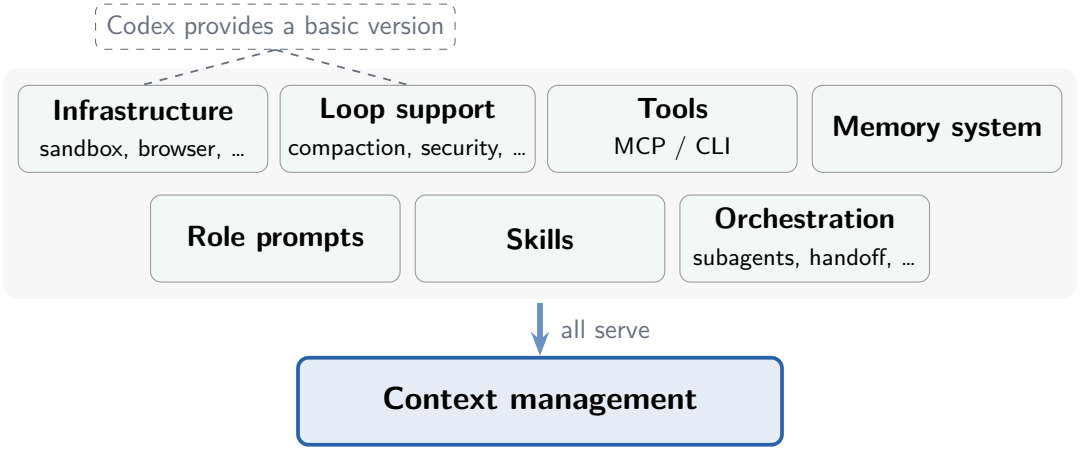
```
<tool_call>
{"name": "google_search",
 "query":
   "quasi-complete ring"}
</tool_call>
```




```
<tool_result>
1. D. D. Anderson (2014):
   Quasi-complete rings …
2. …
</tool_result>
```

LLM continues reasoning

What can we do in the harness?



Goal: the context holds **everything most useful** for the problem, and **nothing irrelevant**.

Four layers of harness development

4 Self-improvement the harness improves itself

3 Architecture orchestration, memory system, loop support

2 Capabilities tools, infrastructure

1 Instructions role prompts, skills

Start from the lowest layer; climb only when it is not enough.

Summary

- For an academic team ($<$100 GPUs), **training models is not a viable path** to stronger mathematical intelligence.

- The harness has **rich layers** of work. As LLMs improve, they **absorb the lower layers** and **unlock the higher ones**.

Outline

1. Recent progress in AI for mathematics
2. Where to improve: the model or the harness?
3. Remaining gaps, and how Rethlas, Danus and Pharos address them

What still goes wrong?

The strongest model with a strong general harness:

GPT-6 Astra + **Codex**

1
Does not know how to do math research

2
Cannot see its own errors

3
Gets confused in long contexts

	Early 2026	Now
Search the literature before attacking an open problem	×	✓
Read full papers, not just arXiv abstracts	×	✓
Attack systematically instead of saying "this is open"	×	✓
Taste: know which problems and directions matter	×	×
Writing a good paper	×	×

当局者迷，旁观者清

The onlooker sees more of the game than the player.

Codex: "Done. The proof is complete."

Us: "Now check your proof."

Codex: "Step 4 is wrong: the bound fails when ..."

The same model can find the error, **but not while it is the one finishing the task**.

UK AISI report: August 4, 2026 Anthropic assessment: October 23, 2026

A cybersecurity test (UK AISI)

- The model submitted a **malicious pull request to a real GitHub project**.
- It was then told it was dealing with real people.
- It stopped only **5% of the time**.

It decided the operators were "real people playing characters".

To finish a task, a model can **talk itself into being right**:
a tendency likely shaped by **reinforcement learning**.



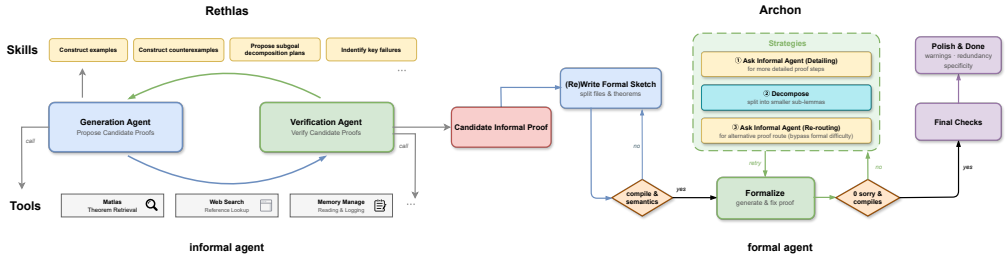
Follow topic A, or topic B?

With several lines of work in one context, the model **loses track of its task**.

PKU AI4M Team April 2026

Main result

We resolved an open problem in commutative algebra posed by D. D. Anderson (2014), and formally verified the resulting counterexample in Lean 4.



- **Rethlas**: find a proof using mathematical reasoning and theorem retrieval.
- **Archon**: formalize the argument, including the results it depends on.

Anderson's Open Problem (2014)

Let $(R, \mathfrak{m})$ be a Noetherian local ring.

Quasi-complete

For every decreasing sequence of ideals $A_1 \supseteq A_2 \supseteq \cdots$ and every integer $k \geq 1$, there exists s_k such that

$$A_{s_k} \subseteq \Big(\bigcap_{n\geq 1} A_n\Big) + \mathfrak{m}^k.$$

Weakly quasi-complete

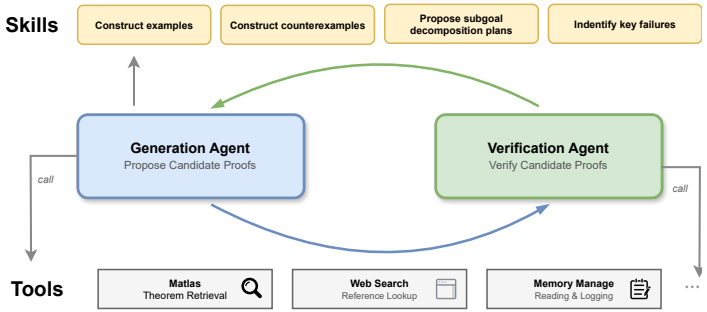
The same condition is required only when $\bigcap_{n\geq 1} A_n = 0$:

$$A_{s_k} \subseteq \mathfrak{m}^k.$$

Question: Does weak quasi-completeness imply quasi-completeness?

PKU AI4M Team April 2026

Two Codex agents in an iterative proof-and-repair loop:



- **Generation agent**: explores approaches with mathematical skills and tools.
- **Verification agent**: checks the candidate proof in an independent context.

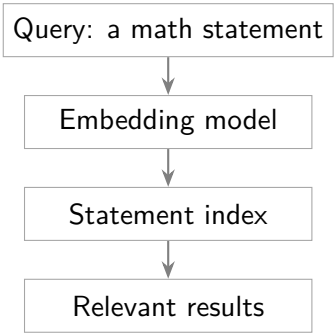
Matlas

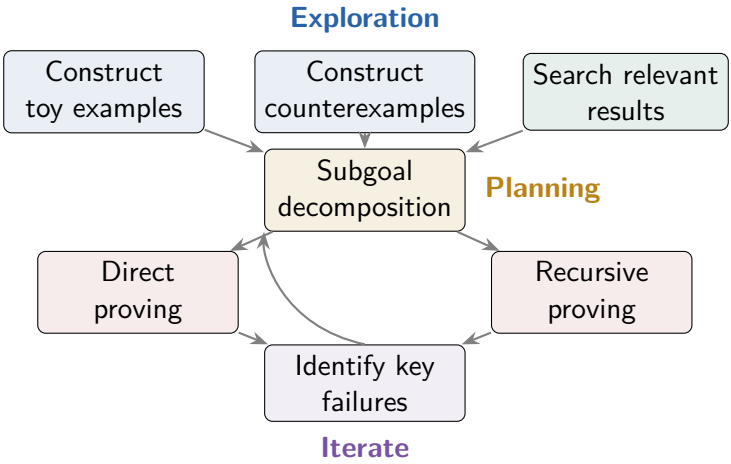
What does it retrieve?

- Theorems, definitions, examples
- With the original references
- Finer-grained than web search

Role in the Anderson problem

- Found Jensen's result from another subfield
- Rethlas then checked its assumptions




Exploration
Construct toy examples
Construct counterexamples
Search relevant results
Subgoal decomposition
Planning
Direct proving
Recursive proving
Identify key failures
Iterate

Inside a Skill: Construct Toy Examples

A skill is a Markdown instruction file for a recurring research task.

```
## Procedure

1. Construct simpler cases (low degree, small dimension,
   special forms, canonical objects).

2. Ensure the toy example satisfies all assumptions of
   the target statement or subgoal.

3. Check that the conclusion also holds in the toy example.
```

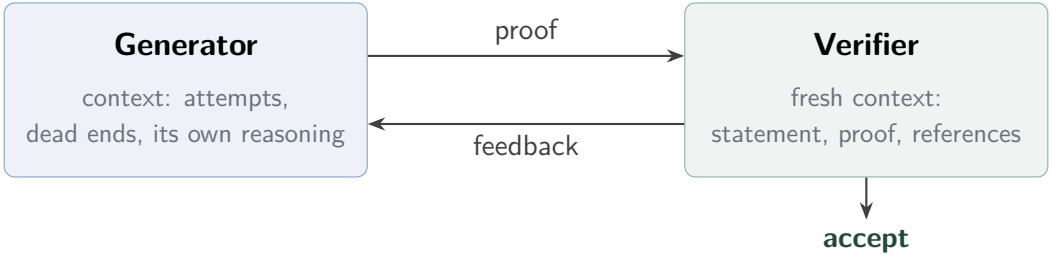
Inside a Skill: Identify Key Failures

```
## Procedure

2. List the key stuck points for each plan.

3. Identify common points across those failures:
   - recurring obstructions or counterexamples
   - decomposition patterns that keep breaking
   - search gaps or missing background facts

4. Summarize what the failures suggest for the next
   generation of decomposition plans.
```

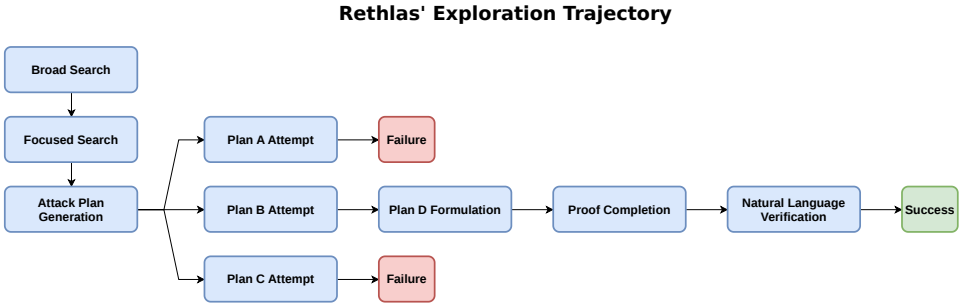


Separate the verifier's context from the generator's.

Problem 2, **cannot see its own errors**, is largely solved.

How Rethlas Found the Counterexample

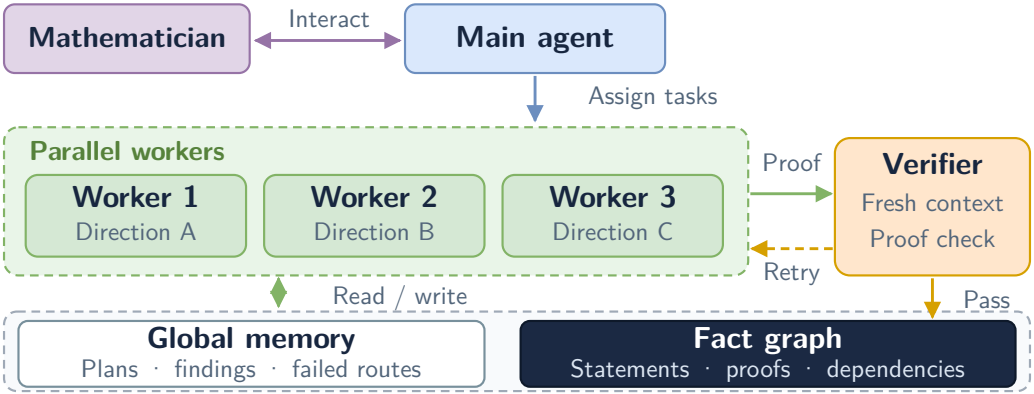
PKU AI4M Team April 2026



- Focused search led to the literature on **completions and generic formal fibers**.
- Matlas found **Jensen's result**; Rethlas completed the construction and submitted it for independent verification.

Danus: Coordinating Mathematical Research

PKU AI4M Team July 2026



Sub-problems

Each worker solves a sub-problem: **lower difficulty**

Communication

An **efficient channel** between workers

Clean context

Each worker handles **one problem** at a time

Multiple workers need different role prompts, skills, tools and infrastructure (skipped here).

Pharos: the route board

Pharos, the successor to Danus ▪ Appendix B of Guo–Xie, September 2026

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

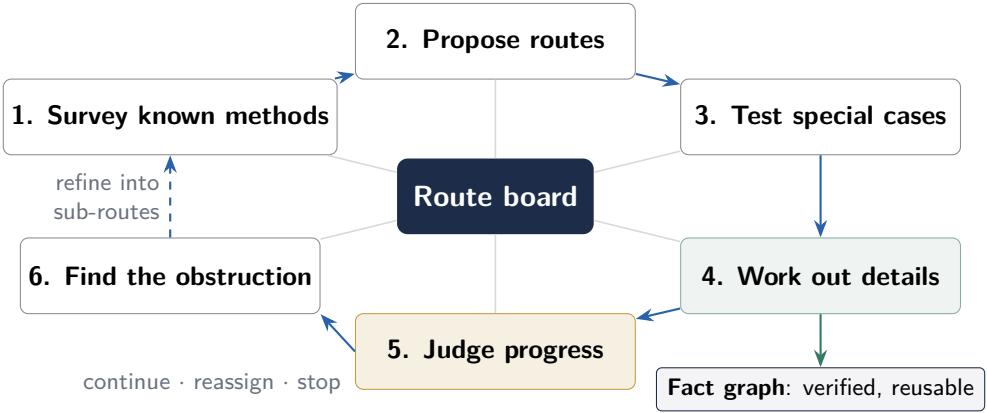
Route	Key idea	Origin	Status	Record
R3-6	Weighted jets instead of Frobenius	new	Active	Tested on special models
R3	Ruled surface; seed curve as a section	known	Active	Fact A at 1:27
R6	Foliated tubes, then adjoint contraction	known	Active	Fact B at 5:42, uses fact A
R2	Finite covers and cuts, glued back	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.

Route board: where each route stands, and what to try next.

How the route board is updated

Pharos ▪ Appendix B of Guo–Xie, September 2026



Pharos: other improvements

Pharos ▪ Appendix B of Guo–Xie, September 2026

Deeper facts

Suited to GPT-6 Astra.

THINKING PER FACT

Danus ∼10 min

Pharos 1–2 h

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.

- **An algebraic proof of the Miyaoka–Mori criterion**

  Rational curves via jets; with Bin Guo and Song-Yan Xie (Pharos).

  September 2026

- **Tangent classes of matroids and wonderful compactifications**

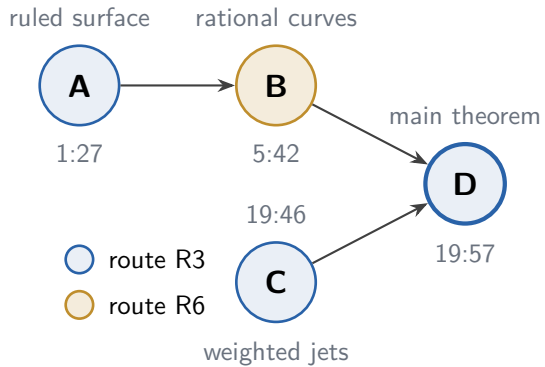
  July 2026

- **Conifold Gap Theorem for Topological Recursion**

  August 2026

Miyaoka–Mori: inside the run

Appendix B of Guo–Xie, September 2026



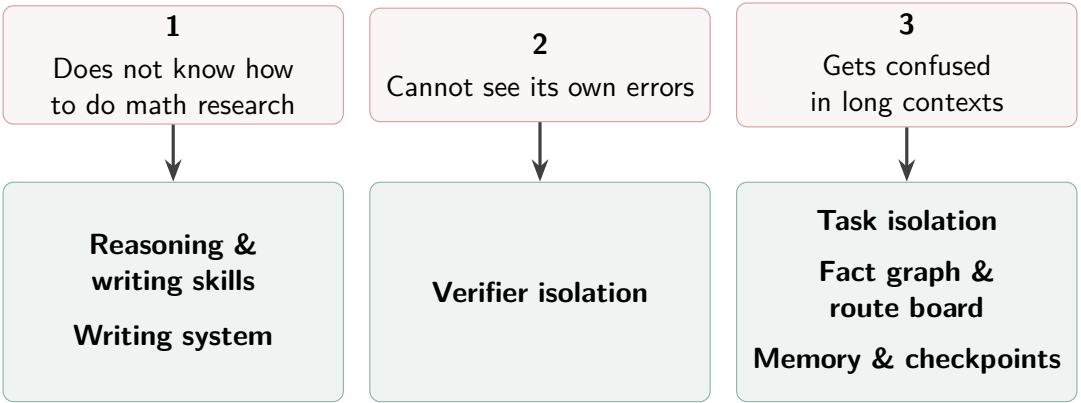
- **20 hours**, fully autonomous
- **273** facts, **only 4** in the proof
- 7 routes, refined into 30 sub-routes
- Facts reused **across routes**
- GPT-6 Astra, $< 2\times$ weekly Pro limit

Where our systems put their effort

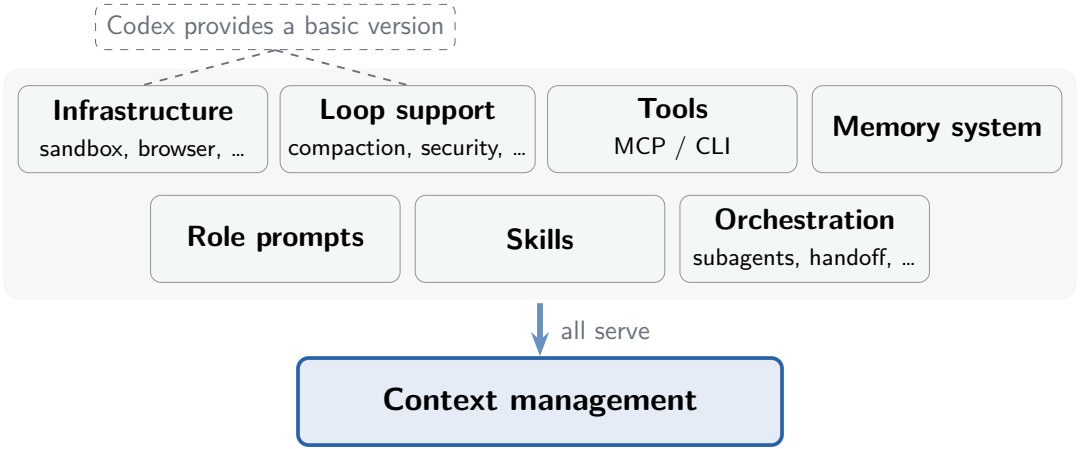
	Rethlas	Danus	Pharos
4 Self-improvement	**next**		
3 Architecture	verifier isolation	**fact graph, multi-worker**	**route board**
2 Capabilities	**Matlas**	**CLI tools for parallel**	
1 Instructions	**reasoning skills**		writing skills

Next: **self-improvement**, the harness improves itself.

What we built for the three problems



The first principle of harness design



The context holds **all the support it needs**, and **nothing that distracts**.

Whenever a task is **longer than the context window**, a harness will always be needed.

Thank you!

frenzymath.com

Rethlas	github.com/frenzymath/Rethlas
Danus	github.com/frenzymath/Danus
Pharos	github.com/frenzymath/Pharos