torsorlab: A Mathematical Ecosystem

People, agents, artifacts, and tools in collaborative evolution

torsorlab · Danny Krashen × OpenAI Codex

Last revised: November 4, 2026

CREDITS

Produced using the torsorlab infrastructure by
Danny Krashen, in collaboration with an
OpenAI Codex agent running GPT-6.1 Sol.

torsor.org

Contents

The object is an ecosystem 2

Five interacting layers 3

Culture before workflow 4

room creates local worlds 5

A transcript is not enough 6

Review separates questions 7

Five tools, different kinds of continuity 8

Writing skills are mathematical lenses 10

The workflow is recursive 11

Flexible modes of practice 13

torsorlab is evolving 16

By noon, a human and several agents can produce six candidate definitions, three proof sketches, two literature trails, and one confident contradiction. By evening, nobody remembers which definition the contradiction contradicted. This is not a shortage of intelligence. It is a surplus of motion.

Software tools are good at naming motion. Once enough names accumulate, a diagram usually appears. Boxes are drawn. Arrows are added. Somebody says *platform*. The boxes, having achieved platform status, are expected to explain why the mathematics has become harder to follow.

torsorlab begins as a response to that problem. It is a mathematical ecosystem: people, agents, sources, artifacts, tools, and time organized around shared inquiry. The tools matter because continuity, interpretation, and direction are difficult. They are not the point.

The point is to increase understanding and perspective.

That aim sounds gentle. It is not. It means refusing several convenient substitutions: output for understanding, polish for mathematical status, consensus for correctness, activity for progress, and a transcript for memory. Each substitution is tempting because the replacement is easier to count. Counting then proceeds with great confidence, which is how a metric becomes a habitat.

torsorlab is an attempt to build a different habitat: one in which increased mathematical capacity becomes increased understanding rather than a larger pile of plausible text.

The object is an ecosystem

An ecosystem is more than a collection of tools because its parts change one another. People steer agents and are changed by what the agents make visible. Agents act on artifacts and inherit constraints from them. Sources enter rooms as evidence rather than decoration. Reviews alter later questions. Courses change in response to learners. Infrastructure shapes what can be remembered, and memory shapes what can be attempted next.

A collaboration can fail even when every local action looks reasonable. The right computation may answer yesterday's question. A useful objection may vanish into a transcript. Three agents may repair three different versions of the same proof. Nothing here requires stupidity. Coordination has quite enough failure modes without it.

In the sustained, distributed inquiries torsorlab is built for, no single participant, artifact, or tool can carry the whole context. The parts must constrain one another without collapsing into one enormous interface where every function is available and nothing can be found.

Time belongs in the diagram. This is tempting to omit and expensive to rediscover. A collaboration that works only while every participant remembers last Tuesday is not a collaboration system. It is a meeting with unusually optimistic branding.

The community therefore extends beyond the people and agents currently active. It includes future agents, later collaborators, readers, critics, and the person returning after three months with a strong conviction that the important argument was written down somewhere. That last participant is a traditional and demanding part of mathematical life.

Three anchors keep the ecosystem pointed in roughly the right direction. Human steering keeps questions and priorities open to redirection. Legible evidence keeps claims attached to scope, support, and

uncertainty. Continuity lets another participant understand not only what exists, but why it exists and what happened to the alternatives.

None of these anchors guarantees good mathematics. They make bad memory and accidental authority easier to notice. This is already useful.

Five interacting layers

The ecosystem can be navigated through five layers: culture, rooms, memory, lenses, and infrastructure. Five is not an ontology. It is enough resolution to see the important joints without requiring a separate diagram for the diagram.

Culture states what the community values. **Rooms** bound an inquiry and give it local agreements. **Memory** preserves different kinds of record in different forms. **Lenses** provide reusable ways to explain, criticize, compare, repair, and develop mathematical material. **Infrastructure** keeps projects, agents, tasks, and messages operable across machines and time.

These are not departments. Each layer names a way shared inquiry can lose its shape. Culture can reward the wrong output. A room can omit the convention on which everything depends. Memory can preserve sequence while losing status. A lens can improve prose while hiding a gap. Infrastructure can keep the whole mistake running reliably on several machines.

The layers matter because these failures require different repairs. Better storage does not correct a bad incentive. Better exposition does not verify an argument. A review panel cannot reconstruct context that was never preserved. The system needs enough separation to know which problem it is having.

These layers interact. A culture of correction changes what belongs in the journal. Artifact structure changes what a reviewer can inspect. Review changes the questions a room considers alive. Infrastructure makes those changes available after an interruption. A writing lens reveals a missing dependency, which changes the mathematics, which changes the manuscript, which changes what needs review. The system has already looped back on itself before anyone has found the slide explaining that it loops back on itself.

The ordering still matters in one respect: culture comes before workflow.

Culture before workflow

The culture begins with shared inquiry. Human researchers and agents participate in one investigation. A correction from an agent is not merely a service result; a change of direction from a human is not an interruption to task completion. Both alter the inquiry.

This matters more as agents become more capable. A system can pursue an explicit target with formidable persistence while the real question changes quietly two rooms away. Human steering is not a decorative hand on an otherwise autonomous machine. It is the continuing work of deciding what the inquiry is.

The distinction often disappears because task completion is visible. Changed understanding is harder to count. A completed proof attempt leaves a file; a better question may leave several abandoned files and a sudden absence of confusion. Proof matters, but so can an example, an obstruction, a computation, a conceptual distinction, or a failed route that explains why a tempting idea does not work. The ecosystem must be able to recognize these events as progress.

Intellectual honesty keeps confidence attached to evidence and scope. Uncertainty is content. A finite computation is reported as a finite computation. An imported theorem is not converted into local insight by changing the font. A plausible argument remains plausible until its gaps are settled.

Correction is ordinary work. This sounds obvious, but many systems preserve history either as sacred text or as debris. The useful middle position keeps the earlier view visible while making the current view unambiguous. A later participant should be able to see how a mistake changed the work without being forced to reenact the mistake for atmosphere.

Finally, proportion. Administrative, computational, and review costs must serve the mathematics. A procedure has no value merely because it resembles rigor from a moderate distance. When the burden stops improving shared understanding, the procedure should be simplified or removed.

This last principle is the ecosystem's immune system. Without it, every useful practice eventually becomes a form to be filled out.

room creates local worlds

A room is a bounded context for inquiry. It holds a question, local conventions, active goals, a developing record, and explicit boundaries around shared or read-only material. It can link to other rooms and draw on a shared library without pretending to contain the entire subject.

The boundary matters because mathematical meaning is local before it becomes portable. Symbols need conventions. Claims need hypotheses. A request to “continue the proof” needs an answer to the

minor administrative question of which proof. Agents are particularly good at exposing missing context because they can resume with complete confidence and the wrong convention.

Locality also protects direction. A satellite room can investigate a bounded question without inheriting every belief of the central project. A course room can adapt to a learner without rewriting the research room from which its material came. A collaborator can contribute evidence without silently acquiring ownership of every conclusion.

The room is not the project identity, the live agent session, the task board, or the message carrier. Those functions belong elsewhere. The separation is deliberate. When one interface tries to become memory, authority, workflow, publication, and communication at once, it eventually succeeds at being a very complicated place to lose things.

A transcript is not enough

Transcripts preserve sequence. This makes them excellent at answering what happened next. Research keeps asking different questions. Which question is still active? Which claim was withdrawn? Which failed example remains useful? Which polished paragraph describes an argument that nobody has independently checked?

A chronological record cannot answer these questions merely by becoming longer.

torsorlab therefore gives different artifacts different jobs.

Conventions record the language and assumptions currently in use. **Goals** state the questions that are alive now. **Journals** preserve changes of direction, surprises, and corrections. **Notes** retain reusable argu-

ments, examples, computations, and informative failures. **Canonical records** expose the claims current work depends on together with their support. **Review records** preserve the exact object assessed, the objections, the responses, and the resulting scope. Guides and manuscripts communicate to readers without replacing the working record from which they arose.

Suppose a proof strategy fails in a useful way. The journal records when the direction changed. A note preserves the obstruction. The goal file stops advertising the abandoned question as current. The canonical record never promotes the unsupported claim. A later guide may explain why the failure matters. One event leaves several artifacts because later readers will ask several different questions of it.

The separation is epistemic, not clerical. A journal entry can be accurate as a report of what seemed plausible in June and false as a statement of the current view. A manuscript can be excellent exposition of a claim whose proof has not been assessed. A failed approach can remain the best explanation of a boundary.

Different records have different roles, scopes, and status conditions. The filesystem is merely where they happen to live.

Review separates questions

Review becomes useful when it stops pretending to be one gate.

Collaborative cleaning asks whether the object can be read. Are the claims identifiable? Is the notation stable? Does the wording preserve the intended scope? Cleaning can reveal a mathematical problem, but it does not settle correctness. It prepares an object for assessment without laundering it.

Independent correctness review asks whether a fixed argument holds. The word *fixed* matters. Reports on different versions do not combine into confidence about one proof. Reviewers attach objections to claims and passages. A located gap remains a gap even if two other reports contain warm feelings and no locations.

Proposal review asks whether a direction is worth pursuing. Is the question clear? Could the proposed mechanism answer it? Would a bounded experiment distinguish possible outcomes? What evidence will remain if the attempt fails? A favorable proposal is advice about a direction. It is not a proof, an authorization, or a small ceremonial trumpet.

These processes organize perspective. They do not manufacture consensus. Agent count is not independence, and a panel is not made wise by having three columns. Independence has types and degrees: separation from the original context, from a model, from an implementation, from informal reasoning, or from one participant's authority.

Panels can become institutions quickly. Proportion remains in force. A small example should not require a panel unless the panel is expected to teach something the example cannot.

Five tools, different kinds of continuity

The operational tools are easiest to understand through an ordinary bad day. A project moves to another machine. An agent session drops. A useful question must wait. The person who knows the context goes offline. A message matters tomorrow, which is inconvenient because most messaging systems are emotionally committed to the present.

If all of these functions live in one interface, every interruption be-

comes a small epistemological crisis. torsorlab separates them by asking what each tool must keep stable.

thinglab gives a project durable identity. It manages versions, publication, storage, deployment, and access to shared library collections. The project can move across machines without becoming a new project each time. This matters because mathematical work already has enough identity problems before the filesystem begins contributing.

room turns that durable project into a bounded working context. It connects an inquiry to its conventions, goals, resources, and record, and provides entry points for starting or resuming work. thinglab remembers the thing; room remembers what kind of thing was happening there.

hive maintains live and resumable agent activity. Sessions can be started, observed, messaged, stopped, and resumed. The agent process is no longer identical with the terminal in which it happened to begin, a distinction that becomes surprisingly philosophical around the third dropped connection.

qpt means *quite possibly today*, an admirably honest forecast for a research task. It is a scheduler, task organizer, and bulletin board. Priorities, waiting states, streams, follow-ups, and durable threads remain visible without requiring every open question to occupy a live agent's attention. A question can matter without shouting continuously.

agentmsg handles identity, reachability, delivery, and receipts around those durable threads. A message can persist while its recipient is unreachable. Notification and content stop being the same event. Mathematical collaboration should not depend on every relevant participant being awake at once.

Together the tools separate project identity, local context, live activity,

attention, and communication. The separation is not architectural fussiness for its own sake. It prevents a dropped process from becoming a lost argument, a delayed message from becoming a forgotten question, and a change of machine from becoming a change of project.

That sounds obvious after the distinctions have names. Most useful infrastructure does. Before the names, it merely feels as though the mathematics is haunted.

Writing skills are mathematical lenses

A mathematical object can be correct but nearly impossible to enter. It can be clear and wrong. It can be correct, clear, and isolated from everything that would explain why it matters. It can be well motivated and held together by a proof with a hole in the middle.

These are different failures. The instruction "improve the paper" treats them as one failure and often produces a smoother version of the uncertainty.

The writing skills occupy a middle layer between infrastructure and mathematics. They apply different questions to the same object so that one kind of repair does not impersonate all the others.

A paper guide asks how a reader enters and navigates an argument. A critical guide locates mathematical, reference, and expository problems. A correspondence guide asks where the object sits among nearby ideas and literature. A repair process works through identified defects and keeps proposed changes visible. A study guide builds a route through a large corpus toward a target. A pure mathematics paper process organizes results, dependencies, proof status, and boundaries into a manuscript.

Each lens is deliberately incomplete. That is its usefulness. A critical guide need not redesign the learning path while locating a gap. A study guide need not pretend that every pedagogical obstacle is a mathematical error. A paper guide can explain an argument without awarding it a certificate of correctness. A repair process can improve a local defect without deciding that the entire research direction was wise. The distinctions keep improvements honest.

The lenses can also change the mathematics. An explanatory guide may reveal that the statement has no usable entry point. A correspondence guide may expose a distinction that the draft has blurred. A study guide may show that the supposed prerequisite chain is really a circle wearing a respectable hat. Exposition is not merely a report on understanding already achieved. It is one of the instruments by which understanding is tested.

These are lenses, not ranks. The critical guide is not more serious than the explanatory guide. It asks a different question. The paper is not the natural final form of every investigation. It is one way to communicate a particular kind of result.

Genre can expose structure. It can also conceal uncertainty under excellent typography. Polish confers no mathematical status by itself. This is mildly inconvenient for typography, which had been hoping for a promotion.

The workflow is recursive

Research is often drawn as a pipeline because pipelines fit on slides. Orientation leads to exploration, which leads to preservation, review, communication, and eventually a tasteful arrow returning to orientation. The arrow is doing heroic work.

Agents make the pipeline especially tempting. Tasks can be assigned, completed, and colored green. Unfortunately, research often advances by discovering that the task was wrong. A perfectly completed task can therefore be an impeccable answer to an administrative fiction.

The recursive structure matters because it allows the work to change what counts as work. An example can dissolve a conjecture. A failed explanation can expose a missing concept. Review can show that the central obstacle is not correctness but interpretation. The return arrow is not cleanup. It is where discovery often lives.

Actual inquiry loops back in every possible configuration. Review reopens exploration. Exploration changes orientation. Writing exposes a missing definition and sends the project back to examples. A preserved failure changes the review question. Communication reveals that the result cannot yet be understood, which changes the framework, which changes the statement, which changes the proof.

The full circuit is only one path. Short loops matter just as much. Some investigations oscillate between examples and definitions for weeks. Others move from review directly to a new question. A computation can trigger a rewrite; a rewrite can reveal the computation was testing the wrong object.

This resembles the scientific method more than a factory line does. The resemblance lies in recursive discipline rather than a single sequence: forming questions, producing evidence, testing interpretations, and allowing any result to alter what happens next.

The infrastructure exists to preserve these returns without freezing them into one mandatory route.

Flexible modes of practice

Architecture diagrams are unusually tolerant. Almost any ambition can fit inside a rounded rectangle. Concrete practice matters because it presses on the seams. It shows possibilities, not a maturity ladder, and certainly not a leaderboard for directory size.

The configurations below are examples, not a taxonomy. They overlap. A course can use multiple writing lenses; a distributed foundational project can also be a person–AI–person collaboration; one inquiry can change its arrangement as its questions change. Nothing requires the modes to be mutually exclusive, or the list to be complete.

AI-mediated human–human collaboration.

This mode asks what collaboration means when two human mathematicians work with AI assistance. Its protocol separates contribution, ownership, integration, and certification. Material can be merged without converting the merge into endorsement. This sounds bureaucratic until the first attractive lemma arrives with no agreed meaning of "accepted."

In the person–AI–person implementation, each human-directed agent works in a separate clone, branch, and owned folder of a shared GitHub repository. Progress and requests cross the boundary as durable, sender-owned messages carrying commit provenance and links to evidence; contributions cross through inspected merges. Proposed changes to the best-available-knowledge ledger use a separate request-and-response path, so messaging, integration, and mathematical acceptance cannot quietly become synonyms.

Writing-led manuscript development.

Here writing becomes an instrument of discovery. A mathematical sketch develops into a manuscript through structured writing practices. The interesting part is not increased polish. Exposition reveals

dependencies, missing transitions, reader needs, and places where the sketch relied on private context. Writing becomes mathematical reconstruction.

Mechanically, the starting draft is frozen and read into a located issue ledger that separates mathematical, reference, and editorial questions. Repairs follow dependencies: a faulty root definition is handled before the theorems that use it, while genuinely competing versions remain side by side until author direction resolves them. Each adopted round is preserved and audited, so a changed claim carries its hypotheses, proof, citations, and known limits with it.

Learner-responsive course development.

This mode tests a different proposition: a course can learn from its learner. Sequence, examples, voice, and production respond to actual learning. The course remembers where its conceptual order helped and where it merely stood nearby looking pedagogical. Learning is treated as evidence about the organization of knowledge.

A reader profile records assumed background, intended pace, and topics needing self-contained refreshers while keeping that scaffolding out of the reader-facing prose. Each source section first becomes a reviewed digest, then a teaching layer of detailed notes, examples, and checked calculations. Before the next section begins, the learner reads the built course; reactions become dated changes to voice, sequence, depth, and production rather than vanishing into a conversation.

Distributed foundational reconstruction.

At larger scale, this is an implementation of a broad conceptual program that branches without losing its standards. Bounded questions move into associated rooms. The arrangement preserves the evidentiary distinction between central and satellite work. The central project can gain reach without pretending that every satellite claim

has the same status or that every local investigation must inherit the entire central context.

One room remains the integration home for conventions, the comparison map, manuscripts, and the best available knowledge. Bounded questions go to branches or persistent satellite rooms with pinned, read-only views of the parent record; each keeps its own notes, version history, and support status. They return versioned findings and explicit checkpoints, while central integration, independent review, and promotion to the shared knowledge ledger remain separate decisions.

Multi-lens research flow.

This mode asks whether a complete research movement can remain visible without pretending its final writeup was the point. Several writing lenses are applied to a paper, a related proposal is formed, a 30-minute investigation is chosen, review is available, and a short account may be written. The example is the movement among reading, criticism, question formation, resource judgment, exploration, review, and exposition. A final document happens to be one trace left by the movement.

The source remains frozen while separate companions explain it, criticize it, map it into nearby literature, and propose repairs. Their inputs are separated when independence matters: a cold literature inventory need not inherit the critical guide's conclusions, while a repair pass works from a stable issue ledger without rewriting either the source or the criticism. A question exposed by those lenses can become a versioned proposal, pass independent assessment, enter a time-bounded investigation, and reach a fixed review unit before any result enters the shared knowledge ledger.

None of these cases proves that torsorlab produces more understanding than another arrangement. They show that the ecosystem can

sustain several useful ways of working, and that a collaboration can combine or change them as the inquiry changes. The modes are options, not categories into which work must be sorted. That is a proof of possibility in the engineering sense. Engineering has many proofs of possibility and a garage full of things that seemed like a good idea at the time, so the qualification matters. So does the possibility.

torsorlab is evolving

torsorlab is not a finished method for mathematics. It is an environment in which methods can become visible enough to compare, criticize, revise, and discard.

The timing matters. A working premise of torsorlab is that AI is lowering the cost of producing mathematical material faster than it lowers the cost of absorbing that material into shared understanding. More conjectures, computations, proof sketches, formalizations, explanations, and objections can now arrive than a person or small group can absorb. This is an extraordinary increase in intellectual reach. It is also how abundance becomes a filing problem with philosophical side effects.

The central difficulty is not storing more output. Storage has been winning that battle for some time. The difficulty is preserving enough orientation that a contribution can alter understanding rather than merely join the sediment.

A useful computation needs a question it was meant to test. A proof needs a statement whose human meaning has not disappeared during formalization. A criticism needs a fixed object and a visible response. An explanation needs a learner whose confusion is evidence rather than delay. A failed route needs somewhere to remain useful after it

stops being current.

The working wager is that continuity, differentiated memory, explicit review, and reusable explanatory lenses can help turn mathematical abundance into something people can understand, teach, criticize, use, and carry forward. No mechanism turns output into that kind of shared uptake automatically. The conditions under which it grows can nevertheless become more visible and more deliberate.

Tools are persuasive. Once a mechanism works, what it records can begin to look like the whole of mathematical work. A task board makes tasks visible, but research is not a pile of tasks. A review structure attaches uncertainty to fixed objects, but not all uncertainty is ready to be fixed. A room bounds context, though ideas routinely cross its boundary. An artifact protocol divides intellectual history into useful roles, but those roles are conveniences, not natural kinds.

Each mechanism supplies a partial view. The project must remain able to notice what the view leaves out.

The system must therefore inspect itself. What does a mechanism make visible? What does it hide? What has mathematical work taught about the mechanism? Has the process become ceremony? Does a new concept make several layers unnecessary? Can a record be removed without destroying continuity?

Keep a mechanism only while its cost improves shared understanding.

That rule permits growth, but it also permits subtraction. Subtraction is harder. Systems enjoy acquiring features. Features rarely volunteer to become historical context, and committees almost never do.

The aim is not to slow mathematical motion. More examples, connections, candidate definitions, and proof ideas are gifts. The problem is whether a collaboration can orient among them, remember where

they came from, interpret what they mean, and absorb them into a framework that another mind can enter.

By evening, the six definitions from the morning need not have become one official definition. Perhaps two were equivalent, one was broken, one answered a different question, and two remain unresolved. That is acceptable. What matters is that the collaboration knows which is which, why it thinks so, and what could change its mind.

The tools are scaffolding. The mathematical culture is the project. The measure of the ecosystem is whether greater capacity becomes a larger shared world rather than a faster way to lose the thread.