Can Geometry Reduce LLM Hallucination in Math Formalization?

Concept Note

Ashani Dasgupta
Cheenta Academy
ashani.dasgupta@cheenta.com

March 2026

Vision: Geometry as Infrastructure

Right now, using an LLM to write formal mathematics is like dropping a brilliant bricklayer into a pitch-black forest and saying, "Build a bridge to the other side." The bricklayer knows exactly how to lay bricks. It knows Lean 4 syntax perfectly. But because it cannot see the forest, it starts building bridges in random directions, attaching them to imaginary trees. In the language of AI, it hallucinates fake theorems. The measurements in our related paper suggest a different architecture. We can turn the lights on.

The Master Map

Imagine taking all of human mathematical knowledge—both the messy, informal human text and the strict, verified computer code—and projecting them into a single geometric universe. The verified Mathlib declarations form a solid, glowing core near the center of the Poincaré ball. The unverified human mathematics—textbook theorems, research papers, lecture notes—sits out on the dark edges, organized by logical proximity but not yet checked by any machine.

This is not a metaphor. The embedding we trained places 439,250 verified declarations into a 16-dimensional ball. A natural next step is to embed informal mathematical corpora (NaturalProofs, ProofWiki, the arXiv) into the same space, anchored by the thousands of theorems that exist in both informal and formal versions. The result would be a single map in which every mathematical concept is a star in a galaxy. Things that are logically related are physically close. The boundary between verified and unverified mathematics becomes a geometric surface—a frontier that we can see, measure, and systematically advance.

Finding the Frontier

Suppose we want to formalize a major unverified theorem. We break it down into informal steps—a proof sketch, the way a mathematician would explain it on a blackboard. Because we have a map, we know exactly where each informal step lives in the geometric space. Some steps hover right on the edge of the green zone, one lemma away from verified territory. Others are deep in the dark, far from anything Mathlib has ever seen.

We target the step that is closest to the frontier. This is not a guess. It is a measurement: the hyperbolic distance from the informal step to the nearest verified theorem.

The Hermetic Seal

Here is the key architectural idea. Instead of asking the LLM to search through 600,000 Mathlib declarations—a haystack so large that hallucination is inevitable—we look at our map and say: "What are the 5 verified theorems closest to this exact coordinate?"

We then wake up the LLM and give it the ultimate constrained prompt:

> *Formalize this informal statement. You are locked in a room. You are only allowed to use these* 5 *specific, verified Mathlib theorems. Do not invent anything else.*

Because the LLM is restricted to the exact logical neighborhood, it cannot hallucinate. It cannot reference a theorem that does not exist. It cannot invoke a lemma from a distant corner of the library that happens to share a name with something relevant. It must connect the 5 verified dots, and nothing else.

If the Lean 4 compiler accepts the result, that node turns green. The frontier advances. We pick the next closest informal step and repeat.

What This Changes

The AI community is currently trying to solve the hallucination problem by making LLMs read more text; more context, longer windows, better retrieval. This approach solves it differently: by forcing the LLM to obey the intrinsic geometry of logical dependencies.

Retrieval-augmented generation asks: "What is textually similar?" Geometric constraint asks: "What is logically adjacent?" These are different questions, and for mathematics, the second one is the right one.

The measurements in our related paper provide the empirical foundation. The tree of cliques is real. The geometry is specific to mathematical content. The embedding faithfully represents the logical structure. What remains is to build the infrastructure that exploits it: the joint embedding, the frontier detection, the hermetic seal.

The geometry does the thinking. The LLM does the typing.

What Remains to Be Built

This vision requires four concrete components, each of which is a research project in its own right:

1. **Joint embedding.** Align informal mathematics (NaturalProofs, ProofWiki) and formal mathematics (Mathlib) in the same Poincaré ball, using paired informal-formal theorems as anchors. The key question: are a few thousand anchor pairs sufficient to align a 64-dimensional space?

2. **Frontier detection.** Given a proof sketch decomposed into informal steps, locate each step in the joint embedding and measure its distance to the verified frontier. Rank the steps by proximity and target the closest one first.

3. **Hermetic seal prompting.** Constrain the LLM to the K-nearest verified theorems at each step. Add a type-compatibility filter: among the K nearest, keep only those whose types are unifiable with the goal. This is geometric retrieval augmented with type checking.

4. **Iterative frontier expansion.** When a step is successfully formalized, add it to the verified embedding, recompute the frontier, and advance to the next closest informal step. This creates a feedback loop: each success makes the next step easier.

None of these components requires a breakthrough. Joint embeddings of heterogeneous data are well-studied. Frontier detection is a nearest-neighbor query. Constrained prompting is standard. Iterative expansion is curriculum learning. What is new is the combination, and the claim that underlies it: *the geometry of mathematical knowledge is a reliable guide to what can be formalized next.*

The results of our related paper, the tree of cliques, the null-model controls, the two verified case studies, are evidence for that claim. The next paper will test it.