

Flexiformality and LLM-Assisted Mathematics

Dennis Müller¹ 

Computer Science, FAU Erlangen-Nürnberg, Germany

The emergence, proliferation, and continuing improvement of LLMs has had a strong impact on the mathematics community. LLMs have been used to (partially or fully) automatically generate proofs and counterexamples to open mathematical conjectures, exacerbating a trend that was emerging even *before* their advent: New results are published significantly faster than they can be reviewed, evaluated, scrutinized, digested, and ultimately *canonicalized* by the community. This problem was also a central topic of Terence Tao’s 2026 presentation “*Mathematics in the Age of AI*” at the *International Congress of Mathematicians (ICM)*¹, calling it a “crisis of mathematical practice and values” comparable to the foundational crisis at the beginning of the 20th century.

Moreover, LLM generated proofs being both less naturally written, and having “inhuman” kinds of errors that are less easy to spot for human reviewers, exacerbate the problem qualitatively *in addition* to quantitatively.

A mitigation of this problem in mathematics is to use interactive theorem provers, e.g. by prompting an LLM to generate a *formal proof*. This moves the to-be-verified aspect of the artifact from “*is this proof correct?*” to “*is this formal statement/proof a faithful translation of the informal one?*” (the *faithfulness gap*). This gap on its own poses a (smaller) problem, but a formal proof is also even more difficult to comprehend to an even smaller number of human experts.

Thus, *dispensing* comprehensible prosaic proofs for autoformalized ones gives us in essence a partial *oracle* producing either a formally verified proof, or failing for a plurality of reasons.

What remains in all cases is the *sociological* problem for the community to digest and transform the new results into a shared, canonicalized understanding of the subject matter. Sadly, most discussions of LLMs in Mathematics, in particular in the *public eye*, tend to focus on proofs alone, and as “correct or not” as the metric of value (“*LLM has proven another open problem! Will ChatGPT put Mathematicians out of work soon?*”).

I argue that *flexiformalization* [2] can mitigate these issues: That is, human readable documents, enriched via formal annotations that put a natural language / notational fragment into direct correspondence with its formal equivalent: 1. It allows for active documents that make its contents more accessible to readers. 2. By assigning globally unambiguous identifiers to concepts *and* document paragraphs (definitions, examples, theorems, etc.), and making the relations between them queryable, it makes mathematical results FAIR (Findable, Accessible, Interoperable, Reusable).² 3. By having LLMs generate flexiformal documents,

¹ <https://youtu.be/M0--ZH1l0zg>

² <https://go-fair.org>

they are constrained by two goals simultaneously: The formal annotations can be checked for correctness, and the document itself can be evaluated by an adversarial critic with respect to its “naturalness” as human-oriented prose. 4. By being *foundation independent* and following the *little theories approach*, it is more suitable to building a global digital mathematical library that is both accessible to humans and algorithms (symbolic or statistical).

We have recently developed the **FTML** (Flexiformal Text Markup Language) dialect of **HTML** and an associated mature software toolchain (**F&Mf** - Flexiformal Annotation Management System), which allows us to generate active flexiformal documents from **L^AT_EX** (using the **sT_EX**-package) [4,1,3], ≈ 15.000 of which are available on mathhub.info³.

While *formal verification* in **FTML/F&Mf** is still in an experimental state, we have experimented with LLM use in generating flexiformal verifiable structured proofs. This resulted in human-readable *and* verified proofs for the equivalence of the order-theoretic and algebraic definitions of lattices via theory morphisms, and a proof of the irrationality of $\sqrt{2}$ (in a dependently-typed higher-order logic with natural deduction rules declared in a rudimentary logical framework); generated using Claude Code with two interleaved goals: Proof/definition/lemma generation using **sT_EX** syntax, and “naturalness” as judged by an agent critic on the pdf_latex output.⁴ Some early evidence shows that the correspondence between the natural language text and the annotations can effectively rule out “cheating” such as e.g. exploiting kernel bugs for proofs – in this case, trying to exploit hardcoded arithmetic simplification rules to prove $n+0 = n$ via *reflexivity* rather than fully proving it via induction.

References

1. The sT_EX documentation, <https://mathhub.info/?a=sTeX/Documentation&rp=documentation.tex>
2. Kohlhasse, M.: The flexiformalist manifesto. In: et al., A.V. (ed.) 14th International Workshop on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2012). pp. 30–36. IEEE Press, Timisoara, Romania (2013), <https://kwarc.info/kohlhasse/papers/synasc13.pdf>
3. Müller, D., Kohlhasse, M.: The next step in mathematical publication: Embedded semantics and in-situ services. Mathematische Semesterberichte (2026), <https://mathhub.info/?a=Papers/26-Scientific-Publication&rp=paper.tex>, submitted (in review)
4. Müller, D.: The flexiformal manifesto made manifest. In: Daimi, K., Alsadoon, A. (eds.) The 2026 International Conference on Breakthroughs in Artificial Intelligence (BAI 26). LNNS, Springer (2026), <https://mathhub.info/?a=Papers/26-BAI-FLAMS&rp=paper.tex>

³ See <https://mathhub.info/?a=FTML/demos&d=demo&l=en> for a demo.

⁴ See <https://mathhub.info/?a=FTML/partial-orders&d=all&l=en> and <https://mathhub.info/?a=FTML/demos&p=sqrt2&d=all&l=en>.