ADAPTING MATH INSTRUCTION
TO GENERATIVE AI

Lisa Carbone, Rutgers University

New challenges

The availability of AI tools to the education landscape has brought many new challenges to instructors. Perhaphs the primary ones are guiding students on the appropriate use of generative AI and addressing the needed changed in curriculum and course content.

Mathematical training data for LLMs

Training data for LLMs includes web documents, code with mathematical content, textbooks, online faculty-authored lecture notes and course materials, solutions to problem sets, as well as academic and scientific journal papers and content from arXiv. For example, Google has an unparalleled collection of data from Google Books.

The training data for LLMs extensively covers the standard undergraduate mathematics curriculum from US and other universities, as well as the standard coursework curriculum for PhD programs in mathematics from US and other universities.

The training data for LLMs is not a curated, edited or verified account of these materials. LLMs inherit errors and misconceptions from their training data. Verification of AI output is a necessary step in using LLMs.

LLMs simulate computer algebra systems

LLMs simulate a scientific computing environment by delegating the computation to specialized Python libraries.

Gemini and the paid version of Chat GPT give automatic user access to the Python packages NumPy, SymPy, SciPy, Pandas, Matplotlib.

The Python code is written by the LLM using its probabilistic methods.

Prompts may give incorrect answers if the problems are inaccurately posed. The LLM could also misunderstand the prompt, have bugs in its code or fail to handle boundary cases.

Making LLMs more mathematically rigorous

In LLMs, there is a **system instruction** feature which can be used to influence the type of output. You can enter a ‘pre-prompt’ with your own personal profile that governs the model’s behavior for the entire chat session.

Table 1: Key Python libraries used by LLMs for mathematics

Library	Main purpose	Key capabilities & LLM use
NumPy	Numerical Computing	Handles arrays & matrices. Used by LLMs for linear algebra (matrix multiplication, row reduction, eigenvalues).
SymPy	Symbolic Mathematics	Performs precise algebra & calculus. Used for symbolic derivatives, integrals, and simplifying expressions.
SciPy	Scientific Computing	Provides advanced numerical routines. Used for optimization (finding minima/maxima) and solving differential equations.
Pandas	Data Analysis	Manages structured, table-like data. Used for reading and analyzing data from files such as Excel.
Matplotlib	Plotting & Visualization	Creates a wide variety of 2D graphs and charts. Used to plot functions and visualize data.

Changing the 'knobs'

System settings in an LLM can also be manually changed in order to make mathematical proofs more rigorous and less random.

For mathematical rigor, the most important setting is **temperature**. A high temperature (such as 0.9) encourages more diverse and novel responses. A low temperature makes the output more deterministic. The recommended setting for math is to lower the temperature to 0.2 or 0.1.

Table 2: Optimized settings for math instructors.

Setting	
Temperature	**Low** (0.1 - 0.3) *Reason:* A low temperature minimizes randomness and 'hallucinated' steps in a proof.
Top P	**Low** (0.8 - 0.9) *Reason:* Complements a low temperature. It restricts the model to the most probable tokens, ensuring responses are factual and standard.
Thinking Mode	**Advanced** *Reason:* Advanced mode allows for more complex chains of thought.
Set Thinking Budget	**High** *Reason:* A higher budget prevents the model from 'rushing' and making logical errors.
Code Execution	**On** *Reason:* Allows you to ask the model to verify its own answer, using a tool such as SymPy.
Grounding (Google Search)	**On** *Reason:* Important for fact-checking. Uses known references.
URL Context	**On** *Reason:* Allows you to provide a URL to a textbook chapter or problem set and ask the model to solve specific problems based on that context.
Structured Output	**On** *Reason:* You can ask the model to give a clear, logical format.
Function Calling	**On** *Reason:* Allows you to build functions that connect a specific computational tool, such as like WolframAlpha.
Media Resolution	**High** *Reason:* If you upload an image of a handwritten problem or a diagram from a textbook, this allows the model to read every symbol correctly.
Output Length	**High** *Reason:* A full proof or a detailed problem solution may require substantial length.

Prompt an LLM to act as an expert

You can prompt an LLM to 'give a rigorous mathematical response as one mathematician would explain to another'.

Hybrid systems: LLMs with Computer Algebra Systems or formal proof assistants

LLMs can generate flawless looking mathematics but they lack genuine computational, symbolic and logical reasoning engines.

They can recognize common mathematical proof structures and when to use them. They are masters at *the rhetorical structure of a proof*, but proofs may contain logical and symbolic errors.

The primary way to offset the limitations of using LLMs as tools for mathematics is to use them in conjunction with Computer Algebra Systems or formal proof assistants such as Lean.

WolframAlpha can be enabled in the paid version of ChatGPT and may auto-invoke it when the tool is enabled. It automatically decides whether or not to use the Wolfram tool when prompted with a question.

Instructors can ask an LLM to (first drafts only):

- Create a draft of course content, homework problems or exam questions.
- Generate an alternate version of a given exam for a different section or make-up.
- Fill in details from statements in the text, when the 'proof of the converse is left as an exercise'.
- Generate examples and non-examples.
- Generate several possible solutions to a problem.
- Translate mathematical problems into code or visualizations.
- Generate latex code from handwritten notes or a typed pdf file.
- Generate a Tikz diagram or a table.
- Create a grading rubric.

Output from LLMs can be used as a first draft for teaching materials, but must be rigorously checked for symbolic and logical correctness

Drawbacks for instructors using AI:

- LLMs often omit boundary cases and forget quantifiers.
- They make algebraic and analytic errors (incorrect substitutions for variables, changing the direction of an inequality).
- Computations need to be checked.
- LLMs often make logical gaps and give 'proof by plausibility' arguments.
- LLMs may ignore assumptions and context.
- They have authoritative confidence in falsehoods.

Drawbacks for students using AI

- Students may become passive consumers, not active learners.
- Students may lose fundamental skills and learn 'wrong' statements.
- Learning is short-circuited by fast answers.

As a learning guide for students, AI can:

- Provide step-by-step hints for homework problems.
- Generate a ‘template’ or ‘worked example’.
- Find valid alternative solutions.
- Generate new examples or practice tests and quizzes.
- Explain where their solutions went wrong.
- Translate math into executable code.
- Format answers using LaTeX.
- Ask questions without fear of repercussions.

To offset the effects of AI, instructors can:

- Establish a clear and detailed AI usage policy.
- Deliberately steer students towards critical thinking, conceptual understanding and creative problem-solving.
- Introduce a verbal component to the assessment if possible.
- Emphasize in-person, closed book , ‘AI-resistant’ assessments.
- Thoughtfully introduce AI-assisted assignments where AI is a required tool.
- Design assignments that require students to debug and verify AI output.