Formally Verified AI Co-Reasoning: A Case Study of LeanTutor

UC Berkeley School of Education
UC Berkeley EECS

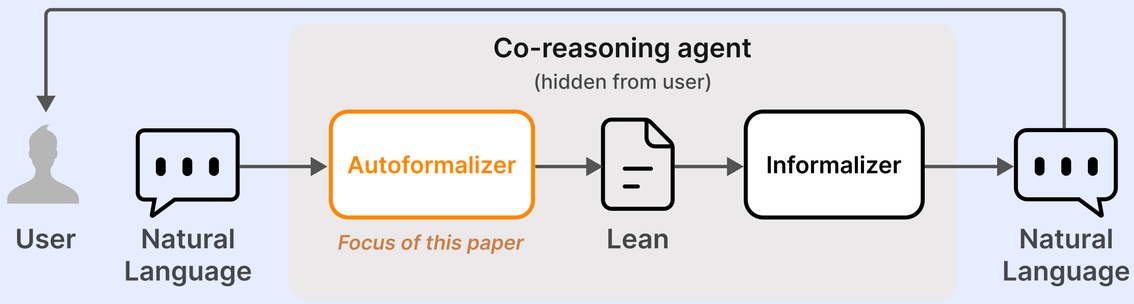
Arnav Mehta (arnavmehta@berkeley.edu), Thomas Lu, Chenjun Guo, Niels Voss, Manooshree Patel, Rayna Bhattacharyya, Peter Donovan, Gireeja Ranade (ranade@eecs.berkeley.edu)

BLISS Berkeley Laboratory for Information and System Sciences

Motivation

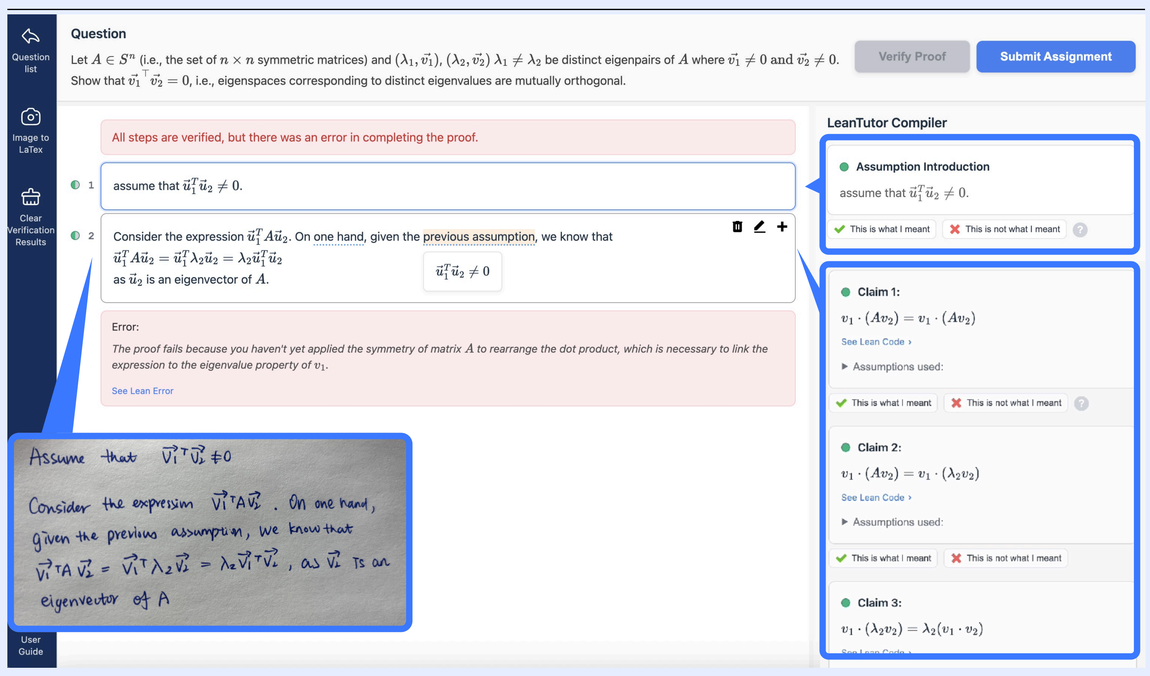
Proof assistants enable machine-checkable verification, but they have a steep learning curve.

Natural-language co-reasoning system: a proof assistant that users interact with entirely in natural language, submitting incomplete or incorrect proofs and receiving formally verified feedback.



Students naturally write exploratory, incomplete, and incorrect proofs, making education an ideal testbed.

Interface



Left panel: student-written proof (transcribed from handwriting via Image-to-LaTeX) with feedback on its correctness & completeness. Right panel: the LeanTutor Compiler's informalization of each formalized step.

Desiderata

We identify five desiderata for an natural-language co-reasoning system:

1. Faithfulness: the formal proof accurately reflects the structure and semantics of the natural-language proof provided as input.
2. Accuracy: The system provides mathematically correct feedback.
3. Appropriate granularity: Lean requires finer-grained proofs than humans write. The system should match the user's granularity level.
4. Transparency: Users can understand the system's state without reading formal code.
5. Incremental editability: Local changes to a proof induce only local changes to its formalization.

Faithfulness Metric

Information Preservation

Did the autoformalizer retain everything that the user included?

$$\text{IP} = \frac{\#\text{ matched User steps}}{\#\text{ User steps}}$$

Higher is better (1.0 = nothing omitted).

Non-Dilution

Did the autoformalizer refrain from adding anything the user didn't include?

$$\text{ND} = \frac{\#\text{ matched Lean steps}}{\#\text{ Lean steps}}$$

Higher is better (1.0 = nothing added).

1. User steps: declarations and assertions.
2. Lean steps: (1) theorems/lemmas (2) sub-claims (`have`) (3) each tactic state
3. Matching: informalize the Lean step; an LLM judges whether it expresses the same mathematical claim as the user step.

Appropriate Granularity

A system can have misaligned granularity in two ways:

- Too strict: requires users to provide unnecessary intermediate steps.
- Too permissive (our focus): accepts large unsupported reasoning jumps.

Granularity detector

If the Lean proof corresponding to a claim "covers" 2 or more rubric items then our system considers that step as having a reasoning jump.

Method for granularity evaluation: grading

To measure how well this works in detecting large reasoning jumps. we grade a students proof in two ways:

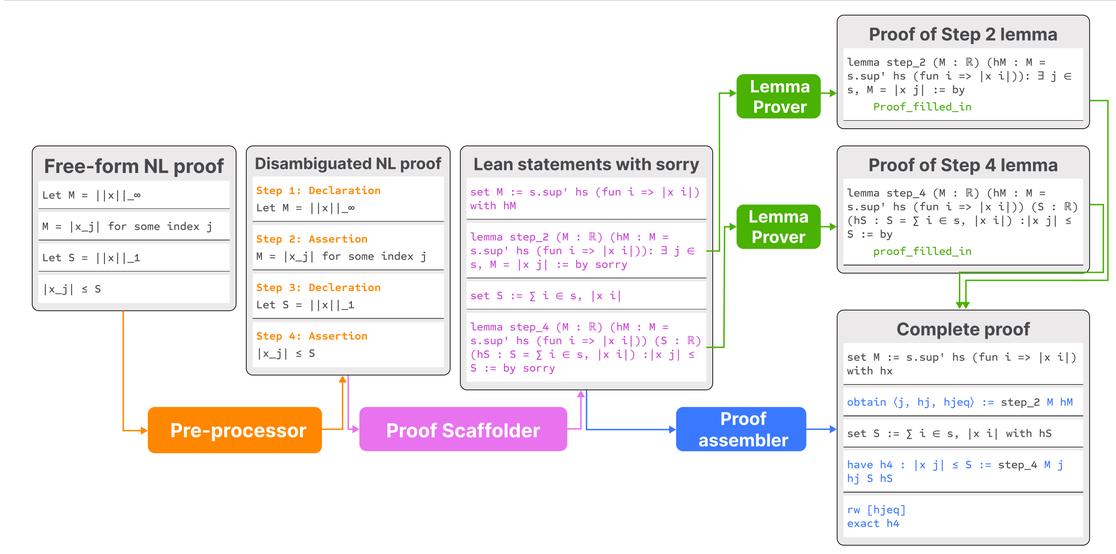
1. Without granularity detector: credit for all the claims that the user made
2. With granularity detector: withhold credit for unsupported reasoning jumps.

Since humans naturally detect large jumps and deduct points for it in the process of grading, we measure granularity by the improvement in agreement (A) with human grades as compared to a baseline (without granularity detection):

$$\Delta = A_{\text{granularity}} - A_{\text{baseline}} \quad \text{(higher is better)}$$

We recognize that this is a proxy measure

Autoformalization Architecture



- Pre-processor: Decomposes the proof into declaration and assertion steps
- Proof Scaffolder: Converts each assertion into a Lean scaffold lemma
- Lemma Prover: Proves each scaffold lemma
- Assembler: Combines the proved scaffold lemmas into a complete proof

Evaluation

100 proofs (10 problems with 10 student solutions each) drawn from homework in EECS 127, an upper-division linear algebra course at UC Berkeley

Faithfulness

	Info. Preservation	Non-Dilution
Lemmas	0.719	0.793
Sub-claims	0.763	0.405
Tactic states	0.410	0.104

Table 1. Info. preservation and non-dilution across Lean step types (lemmas, sub-claims, tactic states)

Finer units capture more content (high info. preservation) but introduce more scaffolding not present in the user's proof (low non-dilution).

Granularity

- 50 student submissions for 1 problem, human graded on a 7-item rubric

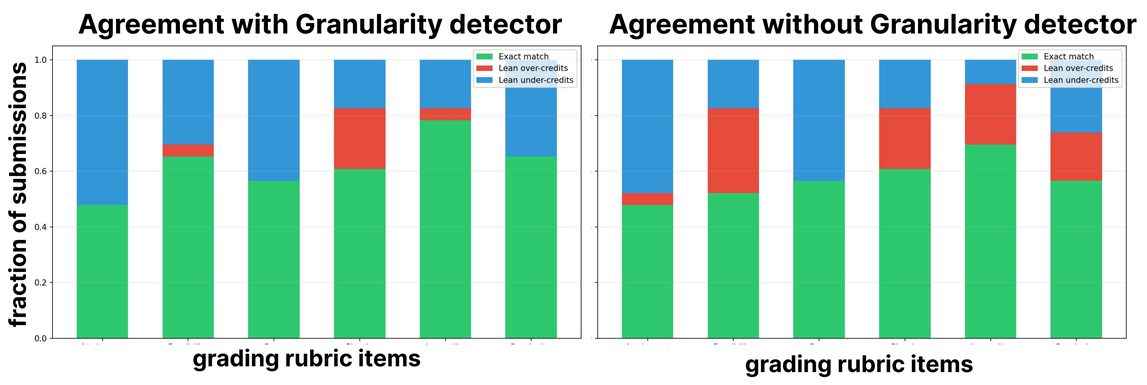


Figure 1. Per-rubric-item agreement between our system and human graders. Each bar shows the fraction of submissions where the system exactly matched (green), over-credited (red), or under-credited (blue) the human grade

Granularity detection improves human agreement by reducing over-crediting

- with granularity detector: 62.1% total agreement with human ($A_{\text{granularity}}$)
- without granularity detector: 57.1% (A_{baseline})
- Δ = +5.0%