We aim to build LeanTutor, an AI tutor for undergraduate-level mathematical proofs, which can verify students' proof correctness, identify incorrect proofsteps, and provide pedagogically motivated feedback to guide students towards the correct proof on homework assignments. LeanTutor:
1) auto-formalizes student proofs into Lean4 code
2) verifies proof accuracy via successfully compiled Lean code or generates a feasible next step in the proof
3) in-formalizes feedback from the Lean system and combined with the class curriculum generates a guiding hint for the student.
A thoughtful UI and UX, drawing upon HCI design principles, are core to the development and success of our system.

While at a surface level the focus of our project is building a provably-correct tutoring system, the system provides a new lens through which we can tackle fundamental open problems in the AI for Math space. Through this project we expect to develop:

- New approaches for auto-formalization
- New datasets that match NL with FL tactics
- A novel use case for auto-informalization

Auto-Formalization
We tackle a fundamentally new challenge in deep-learning for theorem-proving, which is to auto-formalize not only correct, but also incorrect proofs faithfully. Further, we want to identify exactly where the incorrect proof went wrong. However, we have an advantage in auto-formalization — we already have a correct Lean formalization as side-information.
We propose a novel auto-formalization technique that assumes a 1:1 mapping between student proof steps and Lean tactics (or groups of Lean tactics) to sequentially construct a corresponding Lean proof tactic by tactic. We will measure correct auto-formalizations with both the exact match metric and successful compilation, as one NL proof will only have one correct Lean translation with our assumption of 1:1 correspondence. We expect NL proofs which are incorrect to be translated into Lean proofs that do not compile. Our auto-formalization model is currently under development and using a semantic similarity-based approach to match NL steps with FL tactics.

Tactic Generation and Auto-Informalization
Once a student inputs their work, LeanTutor must generate a plausible next step in the proof and a corresponding hint. Our naive first approach for a next step generator is to both search through a dictionary of all successfully compiling tactics and to use our ground truth knowledge of homework proofs. Successful next step generation will be measured by successful compilation in Lean and measuring whether the goal state was

appropriately simplified. This next possible tactic will be auto-informalized into NL, and used to generate a hint to the student within the context of their existing proof. Our project provides a novel and concrete use case for auto-informalizations and helps to make theorem provers more accessible.

Datasets

This project will create two novel open-source datasets.

- The first dataset will contain pairs of NL proofsteps and their corresponding Lean tactics, built specifically for undergraduate mathematics, an area considered data scarce. We have begun to build this dataset by manually translating proofs from two linear algebra classes and building on an existing repo for proofs using the Peano Axioms. While the popular open-source Mathlib4 repo contains many undergraduate math proofs, these proofs use concepts too advanced for some courses (i.e. matrices are defined as modules over rings) and they don't attempt to avoid circularity.
- The second dataset will come from our real world deployment. By testing LeanTutor in Berkeley undergraduate classes, we will collect high quality human-written correct and incorrect proofs and corresponding formalized versions, which can be used to improve auto-formalization. Students' feedback on interactions with LeanTutor will create a dataset of various informalizations and human ratings of them, which can be used to improve auto-informalization. Additionally, the rich dialogues collected between students and LeanTutor provide data for tuning the HCI component of LLM based ITPs (such as proof copilots) or AI Tutors.

Project Relevance and Educational Impact

- Currently, LLM-based math tutors are prone to hallucinations, usually target K-12 levels, and struggle with identifying student errors. Due to these problems, most AI-based learning tools are constrained to human-in-the-loop scenarios. By formalizing proofs in Lean, we aim to greatly reduce hallucinations and interface directly with students, thereby making AI tutors for mathematics more scalable.
- Deploying LeanTutor in a classroom will provide a rigorous evaluation of a DL based ITP tool in a real world setting; this is particularly valuable as most such tools are usually evaluated by only their developers.
- I am a graduate student in both the EECS and Education departments; I plan to bring an interdisciplinary lens to this work in which LeanTutor is both technically novel and pedagogically motivated.