Towards Large Language Models as Copilots for Theorem Proving in Lean

Yeajin Lee

Jan 26, 2024

AIML@K
KOREA UNIVERSITY

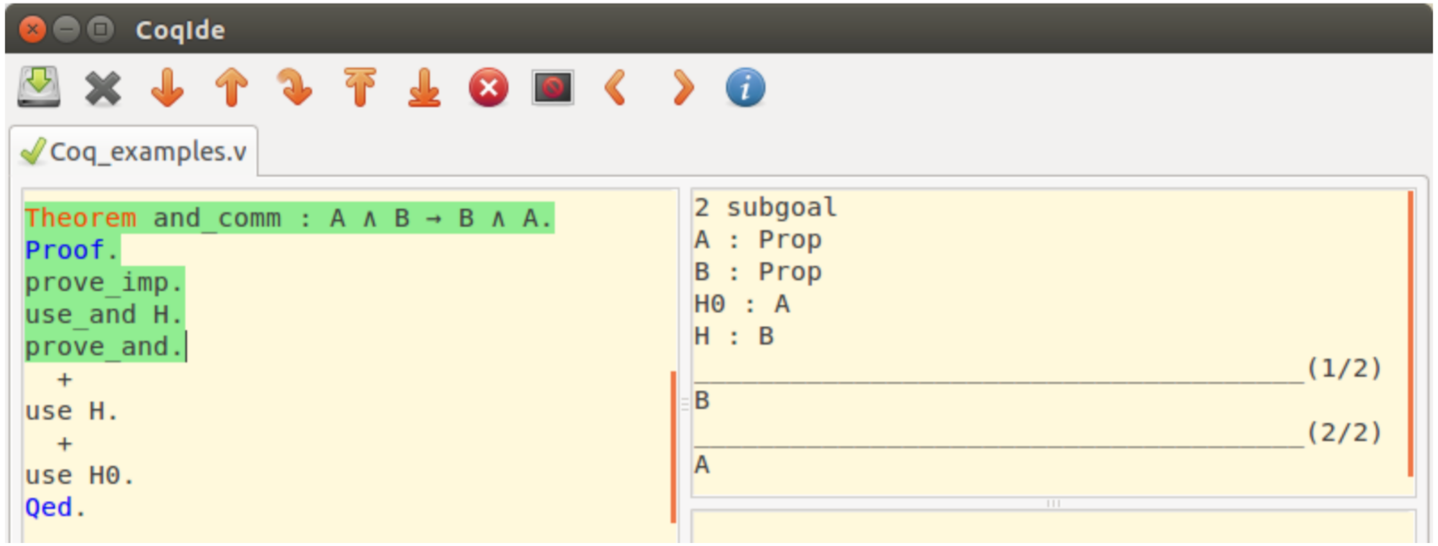
Contents

Background

- Proof assistants (interactive theorem provers)

  : software for mathematicians to write formal proofs

➡️ Use them with machine learning (LLMs), to prove theorems automatically.



Learning how to Prove: From the Coq Proof Assistant to Textbook Style – Figure 3 : The complete proof of of A∧B→ B∧A.

Background

- **Previous aim** : to prove theorems fully autonomously without human intervention.

  ➡️ Often fail to prove theorems

- **Present aim** :

  Instead of proving theorems by itself, AI can also assist human mathematicians in theorem proving.

Lean

: A functional programming language that makes it easy to write correct and maintainable code.

- Starting from the theorem as the initial goal, tactics repeatedly transform the current goal into simpler sub-goals, until all goals are solved.
- **Tactics** : commands, or instructions, that describe how to build such a proof.
  - ex) rfl : X = X, x+37 = 37+x

$$a + (b + 0) + (c + 0) = a + b + c.$$ ➡️ Goal

```
example (a b c : ℕ) : a + (b + 0) + (c + 0) = a + b + c := by

    1 rw [add_zero]
    2 rw [add_zero]
    3 rfl
```

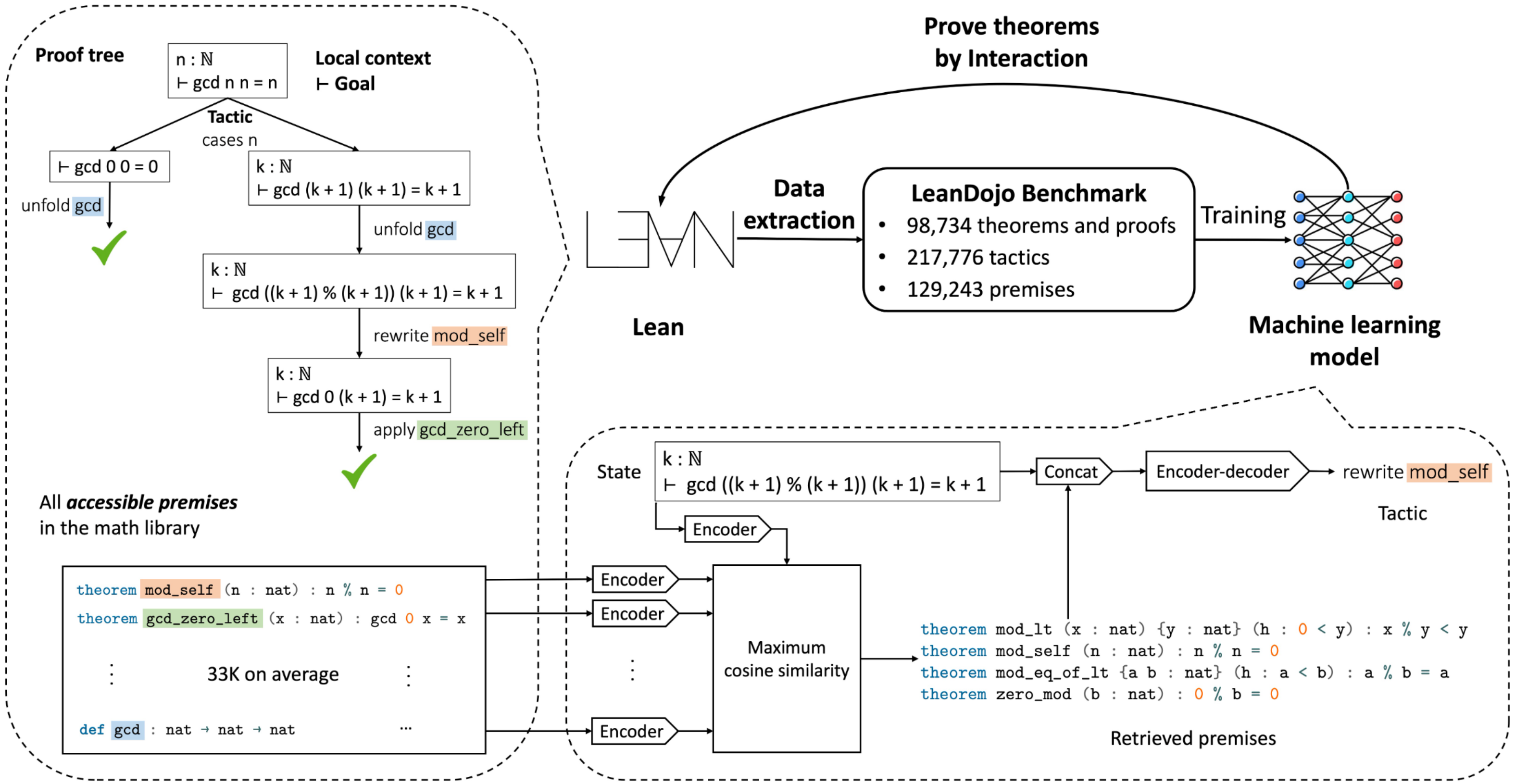
Lean Theorem Probing Tutorial website : https://adam.math.hhu.de/ - /g/leanprover-community/NNG4

Lean Copilot

: A framework for developing LLM-based proof automation in Lean

- Works out of the box
- LeanDojo
- The model is small and efficient enough to run on most hardware

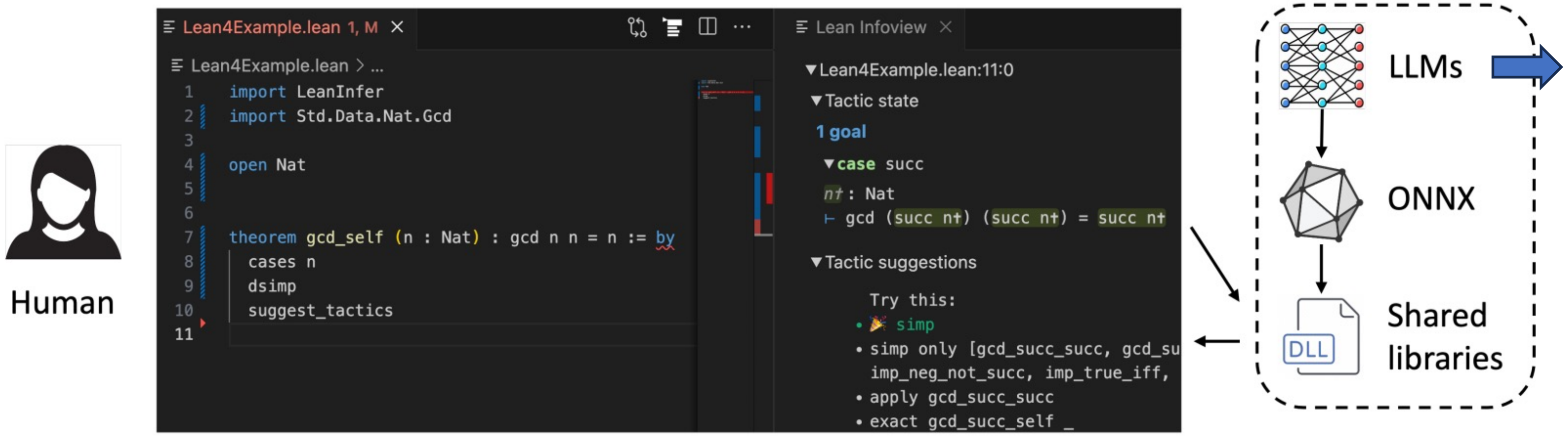
Overview of LeanDojo



https://leandojo.org/

AIML@K KOREA UNIVERSITY

Lean Copilot



The ReProver model from LeanDojo, is Based on **ByT5**

Figure 1: Large language models (LLMs) can assist humans in proving theorems. To prove the theorem `gcd_self` in Lean, the user enters two tactics manually (`cases n` and `dsimp`) and then calls `suggest_tactics`, which uses an LLM to generate four tactic suggestions, displayed in the infoview panel (*Right*). The LLM-generated tactic suggestion `simp` successfully proves the theorem.

- Convert the model into a platform-independent format, ONNX (Open Neural Network Exchange)
- Run it as a shared library through Lean's foreign function interface (FFI)

LLM-based proof automation

Use Lean Copilot to build two **tools** for **assisting** humans in theorem proving

*1. **suggest_tactic***

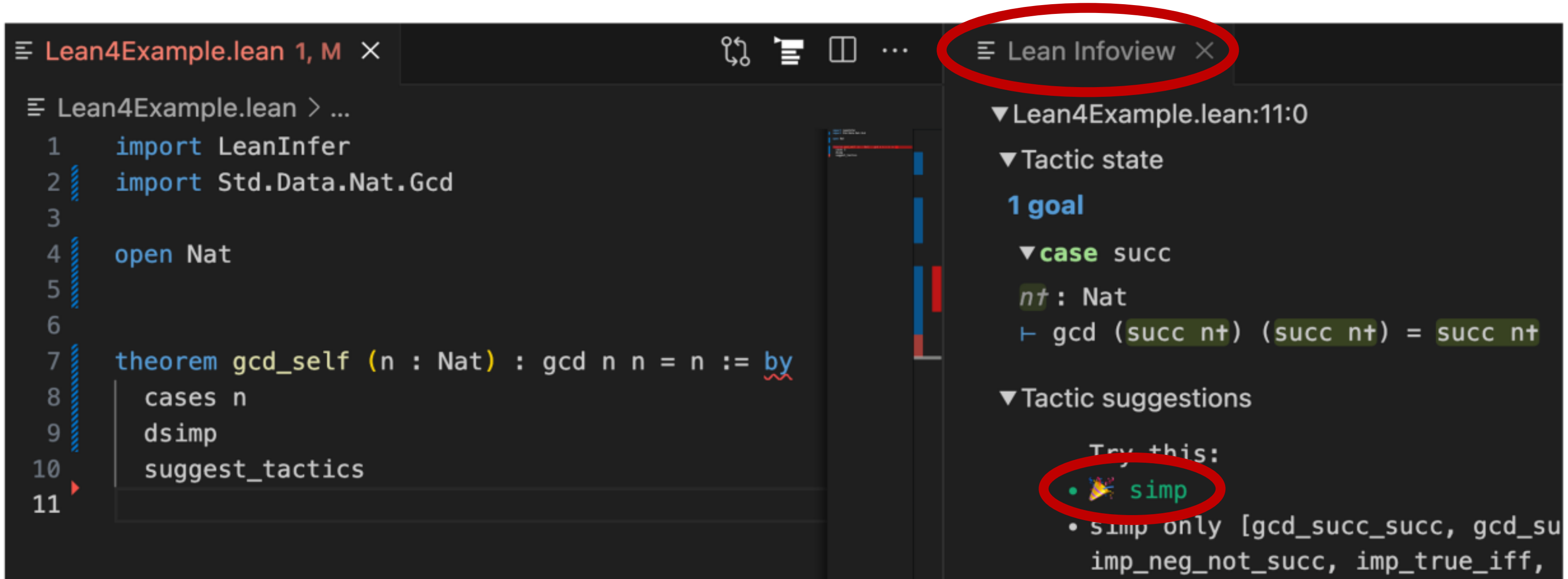
: a tactic that uses LLMs to suggest proof steps

2. **LLM-aesop**

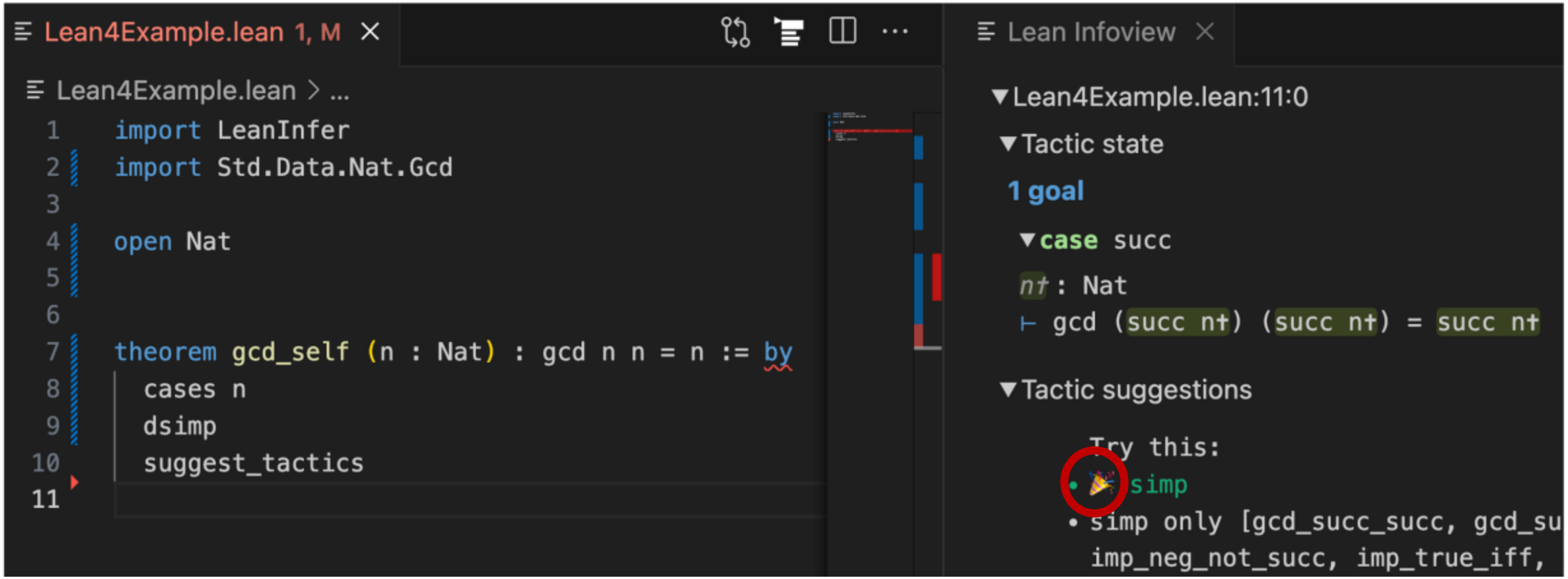
: a proof search tactic that combines LLM-generated proof steps with *aesop*

Suggest_tactic

- It feeds the current goal into an LLM and displays the generated tactics in the 'infoview panel'
- The user can choose whether to accept one of the suggestions by clicking on it
- Our frontend for displaying tactics is based on an existing tactic suggestion tool, *llmstep*
  - *llmstep* : A Lean 4 tactic for suggesting proof steps using a language model



Suggest_tactic



- If a suggestion <u>can directly solve the current goal</u>, it is marked by a party popper emoji (🎉)

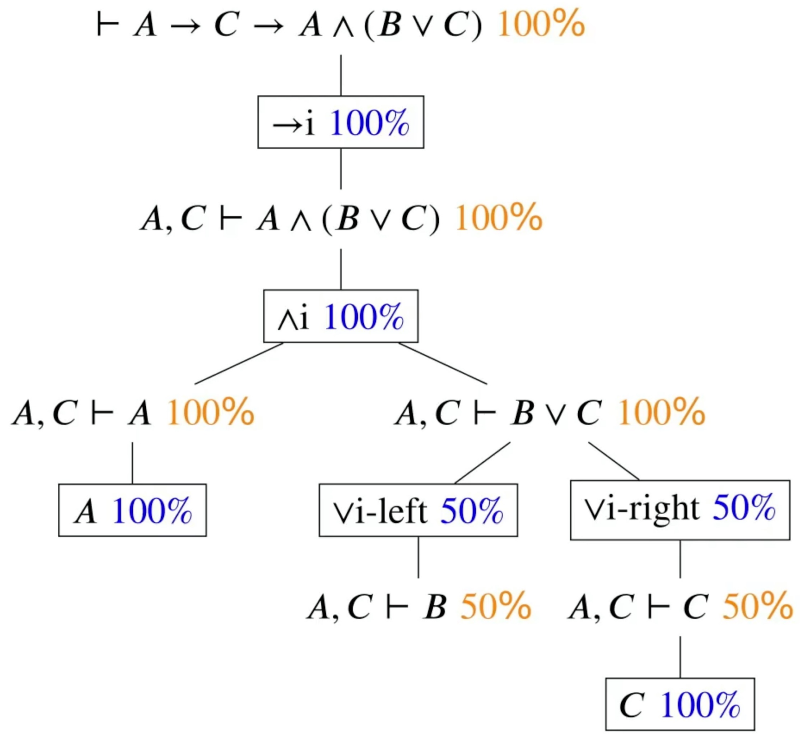
LLM-aesop

- ***Suggest_tactics*** only generates tactics for the current step, without the capability to search for multi-tactic proofs.

  ➡️ Combine it with ***aesop*** to build an LLM-based proof search tool named **LLM-aesop**.

- ***Aesop*** : Implements best-first search and allows users to configure how the search tree gets expanded.

Best-First Search



https://url.kr/rduyo1

LLM-aesop

- Aesop's performance depends critically on problem ➡️ **LLM-aesop**
- Augments aesop's tactic set with goal-dependent tactics generated by ***suggest_tactics***.
- Allow tactics to be customized for each goal, which makes aesop substantially more flexible.
- A drop-in replacement of *aesop*

  : Users can easily switch between LLM-aesop and the original *aesop* by activating/deactivating the LLM-generated tactics.

Experiments

To validate the effectiveness of **LLM-aesop** compared to ***aesop*** and ***suggest_tactics*** in two settings:

- (1) proving theorems autonomously
- (2) assisting humans in theorem proving.
- **Dataset** : Randomly selected 50 theorems in "Mathematics in Lean"

  → their proofs have 5.52 tactics on average

  - Data example

```
example : (a + b) * (a + b) = a * a + 2 * (a * b) + b * b := by
  rw [mul_add, add_mul, add_mul]
  rw [← add_assoc, add_assoc (a * a)]
  rw [mul_comm b a, ← two_mul]
```


AIML@K KOREA UNIVERSITY

Experiments

- **Setup**

  1) To mimic a human user, we enter the ground truth tactics one by one.

  2) After each tactic, we try to prove the remaining goals using an automated tool

  : LLM-aesop, aesop, or suggest_tactics.

  3) Record the number of tactics entered manually before the tool succeeds,

  and the number is zero if it can prove the original theorem fully autonomously

  without requiring human-entered tactics.

Experiments

- Results

Table 1: Performance of `suggest_tactics`, `aesop` and LLM-aesop on proving 50 theorems selected from "Mathematics in Lean" [20]. LLM-aesop outperforms both baselines in proving theorems autonomously and in assisting human users, requiring fewer tactics entered by humans. More detailed results can be found in Appendix B.

Method	Avg. # human-entered tactics (↓)	% Theorems proved autonomously (↑)
`aesop`	3.62	12%
`suggest_tactics`	2.72	34%
LLM-aesop	**1.02**	**64%**

➡️ LLM-aesop can prove **64%** (32 out of 50) theorems autonomously, which is significantly higher than *aesop* and suggest_tactics.

➡️ When used to assist humans, LLM-aesop only requires an average of **1.02** manually-entered tactics, which also compares favorably to *aesop* (3.62) and *suggest_tactics* (2.72)

Conclusion

- Introduced Lean Copilot: a framework for running neural network inference in Lean through FFI.
- Using Lean Copilot, have built LLM-based proof automation for generating tactic suggestions (suggest_tactics) and searching for proofs (LLM-aesop).
- Lean Copilot provides an extendable interface between LLMs and Lean.

  → This work has explored how it enables LLMs to assist Lean users.

⇒ In the future, we hope to see LLM-based proof automation help us formalize mathematics and ultimately enhance LLMs' capability in mathematical reasoning.

Reference:

Lean - https://leanprover.github.io/lean4/doc/whatIsLean.html

Tactic - https://leanprover-community.github.io/mathlib_docs/tactics.html#dsimp

Lean Theorem Probing Tutorial website : https://adam.math.hhu.de/ - /g/leanprover-community/NNG4

Leandojo - https://leandojo.org/

Leancopilot - https://github.com/lean-dojo/LeanCopilot

ByT5 - https://arxiv.org/abs/2105.13626

Llmstep - https://github.com/wellecks/llmstep?tab=readme-ov-file

Pythia - https://github.com/EleutherAI/pythia

Aesop best-first search - https://url.kr/rduyo1

Mathematics in Lean - https://leanprovercommunity.github.io/mathematics_in_lean/