# A Study of Knowledge Distillation for Theorem Proving in Small Language Models

Shubhra Mishra

Stanford University, Stanford, California
shubhra@stanford.edu

**Abstract**

In this work, we will be comparing the performance in autoformalization of a small language model, and comparing it its own performance when it distills knowledge from a larger teacher model. We use Microsoft's `Phi-2` as the small student model, and OpenAI's `GPT4` as the teacher model. We propose a talk where we will be discussing the ability of `Phi-2` to autoformalize, given feedback from the teacher model `GPT-4`.

## 1 Introduction

AI models have shown significant progress across a variety of tasks, often nearing or exceeding human performance on tasks as difficult as visual math problem solving [5]. Because of this, AI models have often been used as human preference/knowledge substitutes. THis has let distillation from a large teacher model to a small student model show immense potential across a variety of tasks [2, 4, 7]. This technique has been especially important, since it lets small language models (SLMs) improve at difficult tasks without the need of scraping or creating data for finetuning. This is especially important in the setting of autoformalization, where collecting or creating data can be difficult, given the difficulty and limited scope of data available for formal mathematical languages.

## 2 Related Work

Autoformalization is the process of taking a natural language statement, and converting it into a mathematical formal statement. An example of a natural language statement, and its formalized counterpart in Lean4 is provided below [6].

```
Natural language statement: Hamming distance is commutative.

Formalized statement in Lean4: theorem hammingDist_comm
(x y : $\forall$ i, $\beta$ i) : hammingDist x y =
hammingDist y x :=
```

Currently, `GPT-4` has proved to be the model with the leading performance across a large variety of tasks. Specifically, a human evaluation of autoformalization capabilities showed that `GPT-4` and `GPT-3.5` outperform Google's `Gemini Pro` in autoformalizaiton abilities. [1]. Jiang et. al., even use `GPT-4` as an informalization tool, showing a 76% accuracy on the task, and create the largest dataset of formal-informal pairs for `Lean4`. [3]

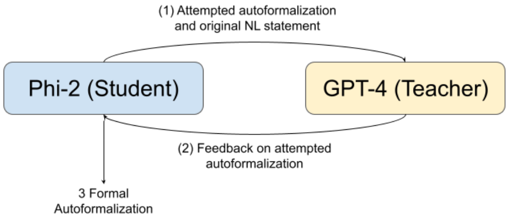
(1) Attempted autoformalization and original NL statement
Phi-2 (Student)
GPT-4 (Teacher)
(2) Feedback on attempted autoformalization
3 Formal Autoformalization

Figure 1: Teacher-student feedback loop

# 3 Methods

## 3.1 Dataset

We will be using the dataset of 101 formal-informal pairs for theorem statements in `Lean4` provided in [1].

## 3.2 Feedback Loop

We will provide as input a natural language statement to our student model, and prompt it to autoformalize the statement in `Lean4`. We will provide the output of the student model, alongside the original natural language statement to the teacher model, and prompt it to provide feedback. We will then provide this feedback to the student model, and ask it to fix its autoformalization as necessary based on the feedback, as shown in Figure 1.

# 4 Future Work

For our talk, we will be presenting our results for this teacher-student setup and its performance on the autoformalization task. While gaps exist even in large language models autoformalization, we hope for this to serve as an exploration of the effectiveness of knowledge distillation in autoformalization, a task that has yet to be explored.

# References

[1] Leonardo de Moura and Sebastian Ullrich. An evaluation benchmark for autoformalization in lean4, 2021.

[2] Chengming Hu, Xuan Li, Dan Liu, Haolun Wu, Xi Chen, Ju Wang, and Xue Liu. Teacher-student architecture for knowledge distillation: A survey, 2023.

[3] Albert Q. Jiang, Wenda Li, and Mateja Jamnik. Multilingual mathematical autoformalization, 2023.

[4] Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kaylan. Let gpt be a math tutor: Teaching math word problem solvers with customized exercise generation, 2023.

[5] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts, 2024.

[6] Shubhra Mishra, Jasdeep Sidhu, Aryan Gulati, Devanshu Ladsaria, and Brando Miranda. The lean 4 theorem prover and programming language (system description), 2023.

[7] Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models, 2024.