Direct Induction Proof Challenge:
Evaluating Large Language Models on Deeply Nested Mathematical Induction

Risako Ando, Koji Mineshima, Mitsuhiro Okada (Keio University, Japan)

Background & Research Question

- Automating mathematical induction has been studied since the 1970s [Boyer & Moore, 1979], but full automation still presents challenges.
- Modern proof assistants require user guidance or lemmas for anything beyond simple proofs.
- While LLMs have shown promise in proof generation [Lightman et al., 2024], they often rely on learned lemmas or library-based tactics.

- **Question**: Can LLMs generate proofs from scratch, i.e., entirely from definitions and mathematical induction without helper lemmas or libraries?
- **Goal**: We investigate the capacity of LLMs to construct deeply nested induction proofs without relying on predefined lemmas.

⚙️ Experimental Setting

Models: GPT-4o, GPT-3.5, Llama-3-70B
Problems: 20 arithmetic statements involving primitive-recursively defined **addition** and **multiplication**
Two settings: Each model is prompted to generate both **informal** English proofs and **formal** Lean 4 proofs.

ID	Example Problem	#variable	#depth
1	$a + 1 = 1 + a$	1	1
6	$a \times b = b \times a$	2	4
14	$(a + b) \times c = (a \times c) + (b \times c)$	3	2
16	$(a + b) \times (c + d) = ((a \times c) + (a \times d)) + ((b \times c) + (b \times d))$	4	4

Informal Proof Task in English

- **Direct task**: No external lemmas/tactics allowed
- **Lemma task**: All used lemmas must be proven.
- Only provided **two-shot** examples on addition
  - Proofs of $a + succ(0) = succ(a)$ and $a + b = b + a$

✔ **Human** evaluation

Formal Proof Task in Lean 4

- **Direct** and **Lemma** tasks
- **Library task**: Use of libraries (**Mathlib**) allowed, but no automation.
- Iterative attempts using Lean error feedback

✔ **Lean** verification

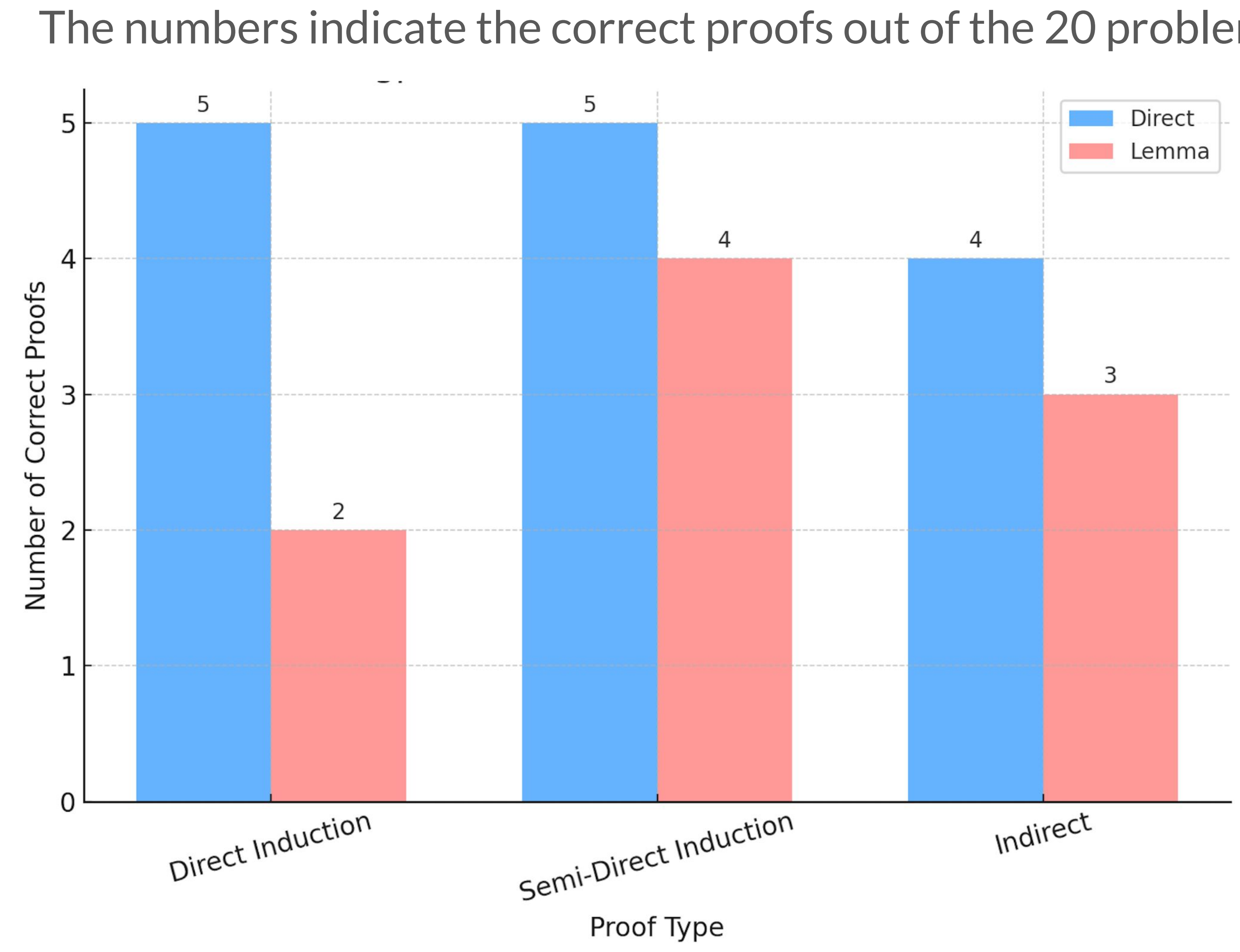
📊 Results

Informal Proof Results (GPT-4o)

- 5/20 correct under the **direct** proof criterion
- 14/20 correct under relaxed criteria (including **semi-direct/indirect** proofs)
- Common issue: incorrect use of definition
  - E.g., the model used $a + succ(b) = succ(a + b)$ (Right Rule), while the definition is $succ(a) + b = succ(a + b)$ (Left Rule)
- Challenge: proving and structuring auxiliary lemmas

Evaluation criteria:
- **Direct**: Only definitions and induction
- **Semi-direct**: Allows left/right addition & multiplication
- **Indirect**: Correct, but not direct or semi-direct

The numbers indicate the correct proofs out of the 20 problems.

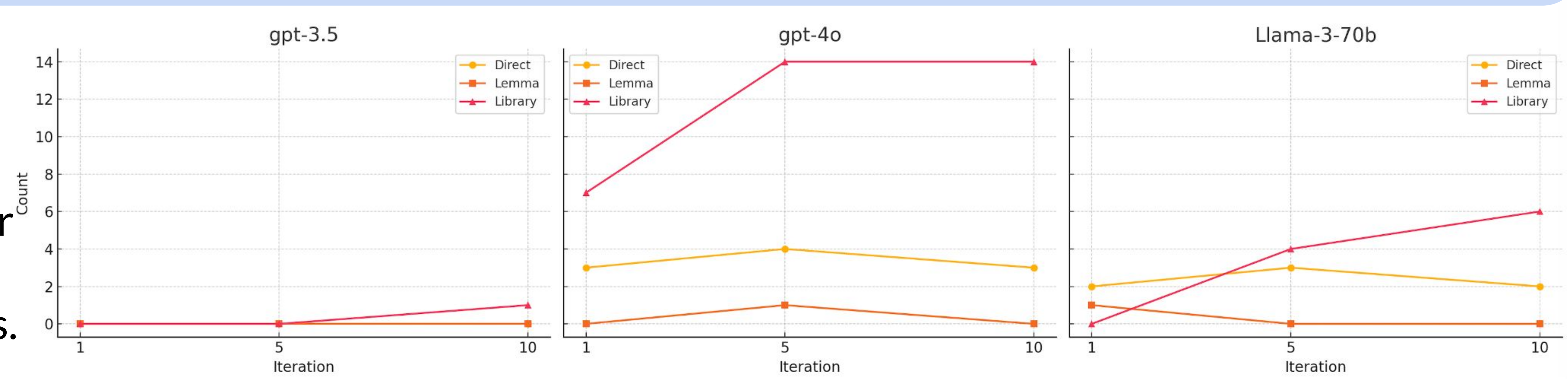


Observations: Generalization abilities of LLMs

- Although the provided samples involved only addition, the model proved a **multiplication** theorem (ID-8) by direct induction, showing generalization beyond **addition**.
- Given only **double induction** samples, it successfully proved a **triple induction** case (ID-12) under relaxed criteria.

Formal Proof Results

- **Direct/Lemma** tasks remain difficult across all models.
- No improvement from Lean error feedback after 1, 5, or 10 iterations in **Direct/Lemma** tasks.



Summary: LLMs show promising generalization in informal settings but struggle with strict direct induction proof construction. Deeper induction remains a significant challenge for automated theorem proving.

2nd AI for Math Workshop @ ICML 2025

This work is partially supported by JST, CREST Grant Number JPMJCR2114.