Projects

11 Nov. 2025, Reasoning in AI

Logistics

Deadlines:

Final report due on **January 23rd**

1-pager (Title, team, project goals, team organization, datasets/benchmarks)
December 8th

From now on, every Tuesday will be office hours in place of the lecture

Benchmark project

Team:

Felix	Bauer
Israa	Fakih
Amene	Gafsi
Rafaila	Galanopoulou
Juan	Garcia Giraldo
Siddharth	Gautam
Saibo	Geng
Julien David	Laurendeau
Chengkun	Li
Matteo	Santelmo
Szymon	Sobczak
Xiuying	Wei
Zhipeng	Xue

Goal: construct a benchmark using the proposed questions. This project involves cleaning the dataset, clustering the questions into different categories, doing comparisons with other (potentially opensource) LLMs, and trying to come up with ideas to address them (e.g., better prompts).

The dataset will be available this week.

Other projects

Teams:

Naser Kazemi, Ilia Mahrooghi, Matea Tashkovska, **Applied or Theory**

Vladyslav Shashkov, Amal Seddas, Luca Pana, Gabriel Alejandro Jiménez Calles, **Applied (own proposal)**

Kirill Brilliantov, **Applied (own proposal)**

Parsa Rahimi, **Applied (own proposal)**

Justin Deschenaux, Alejandro Hernandez-Cano, Andrea Ruglioni, Alexander Hägele, Philipp Schneider, Diba Hashemi, **Applied (own proposal)**

Vasiliki Rizou, Imane Araf, Amel Abdelraheem, Abdellah Rahmani, **Applied**

Luca Viano, Ru Zhang, Millen Kanabar, **Theory (own proposal?)**

Applied projects

1. **Learning abstractions for reasoning problems**

2. Soft guidance to outperform CoT prompting

3. Trade-offs of reasoning: visual domain vs. symbolic domain.

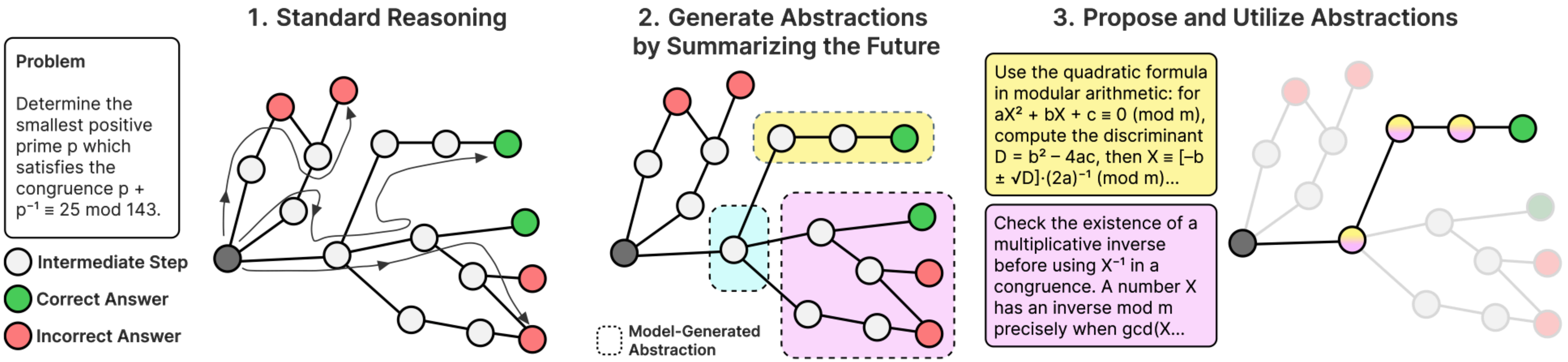
Learning abstractions for reasoning problems

Applied.

Learning abstractions

RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems

Yuxiao Qu*[1], Anikait Singh*[2], Yoonho Lee*[2], Amrith Setlur[1], Ruslan Salakhutdinov[1], Chelsea Finn[2], Aviral Kumar[1]
[1]Carnegie Mellon University, [2]Stanford University, *Equal contribution



Abstractions can serve as “hints” or general direction to solve a problem

Learning abstractions

AbstRaL: Augmenting LLMs' Reasoning by Reinforcing Abstract Thinking

Silin Gao[1,2], Antoine Bosselut[2], Samy Bengio[1], Emmanuel Abbe[1,2]
[1]Apple [2]EPFL

RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems

Yuxiao Qu[*1], Anikait Singh[*2], Yoonho Lee[*2], Amrith Setlur[1], Ruslan Salakhutdinov[1], Chelsea Finn[2], Aviral Kumar[1]
[1]Carnegie Mellon University, [2]Stanford University, [*]Equal contribution

and several other papers

Learning abstractions

Main question:
compare (and improve) the different “abstraction” training procedures

Question 1: which make the model more robust to distribution shifts?

Question 2: how to improve the notions of abstract concepts

Soft guidance to outperform CoT prompting

Effectiveness of Chain-of-Thoughts (CoT)

- Early LLMs had a tendency to generate the answer before the rationale, which degraded their performance.
- CoT was introduced to elicit sequential reasoning
- However, modern LLMs generally produce a high-quality step-by-step reasoning *natively* (i.e. without any supervision)
- Does Chain-of-Thought prompting (with randomly selected examples) remain effective for modern LLMs?

Model	CoT-prompt (best)	free-from
Llama-3.1-8B-Instruct	80%	67%
Qwen2.5-7B-Instruct	88%	89%
Mathstral-7B	74%	84%

Table 1: Accuracy on GSM8K

- Few-shot Chain-of-Thought prompting became the standard baseline for model evaluation
- However, it often performs worse than even a simple free-form model generation. A weak baseline can result in an overly optimistic assessment of a new method, which poses challenges for future research.

Research questions

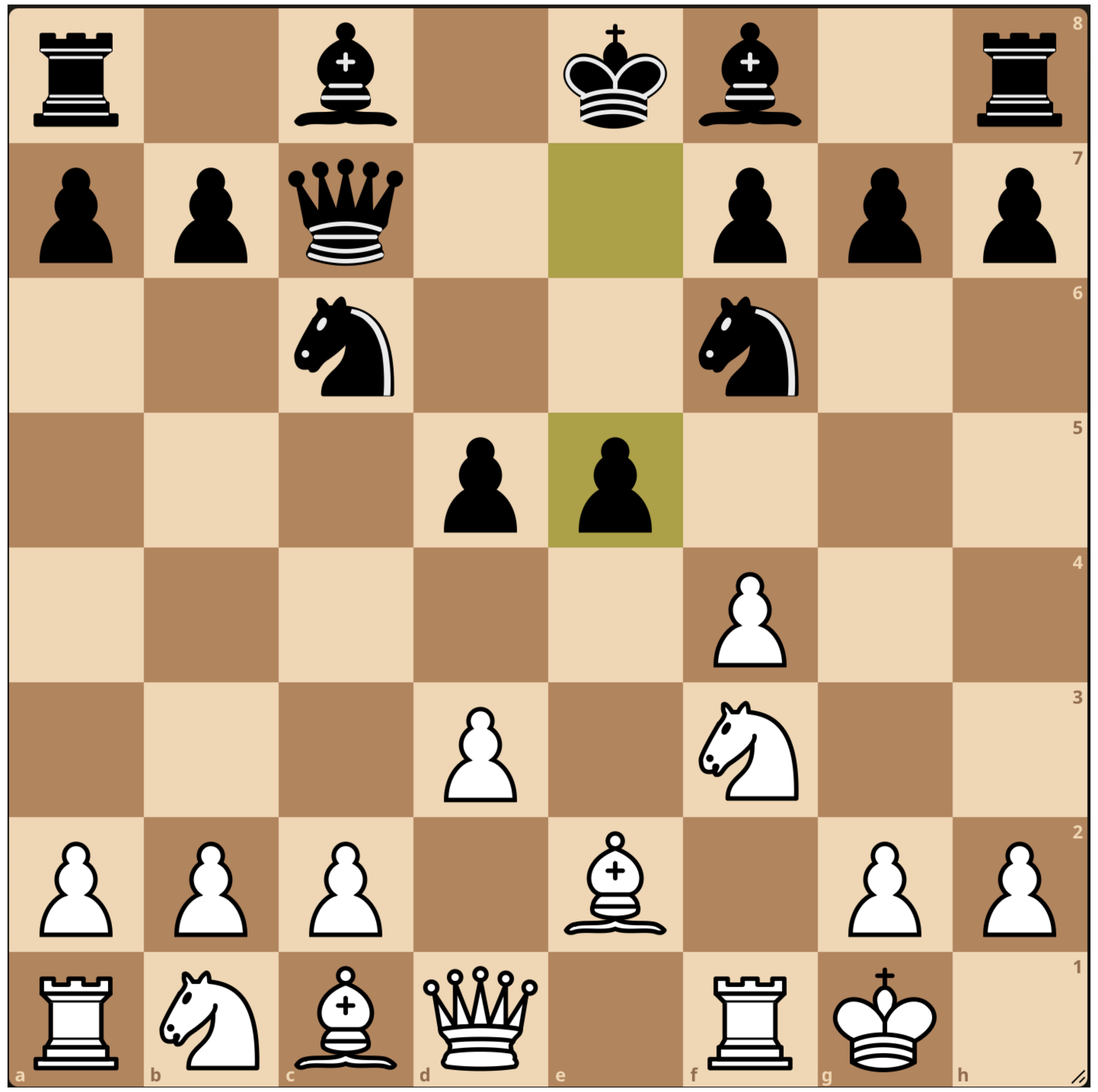
- Does CoT prompting remain effective for modern LLMs? How does it depend on model type and the task?
- Does free-form generation provide a stronger baseline for evaluating modern LLMs?

Trade-offs of reasoning: visual domain vs. symbolic domain

Applied (but theory could be possible).

Reasoning on visual tasks with a symbolic latent

- Many visual reasoning tasks have an underlying symbolic latent structure.



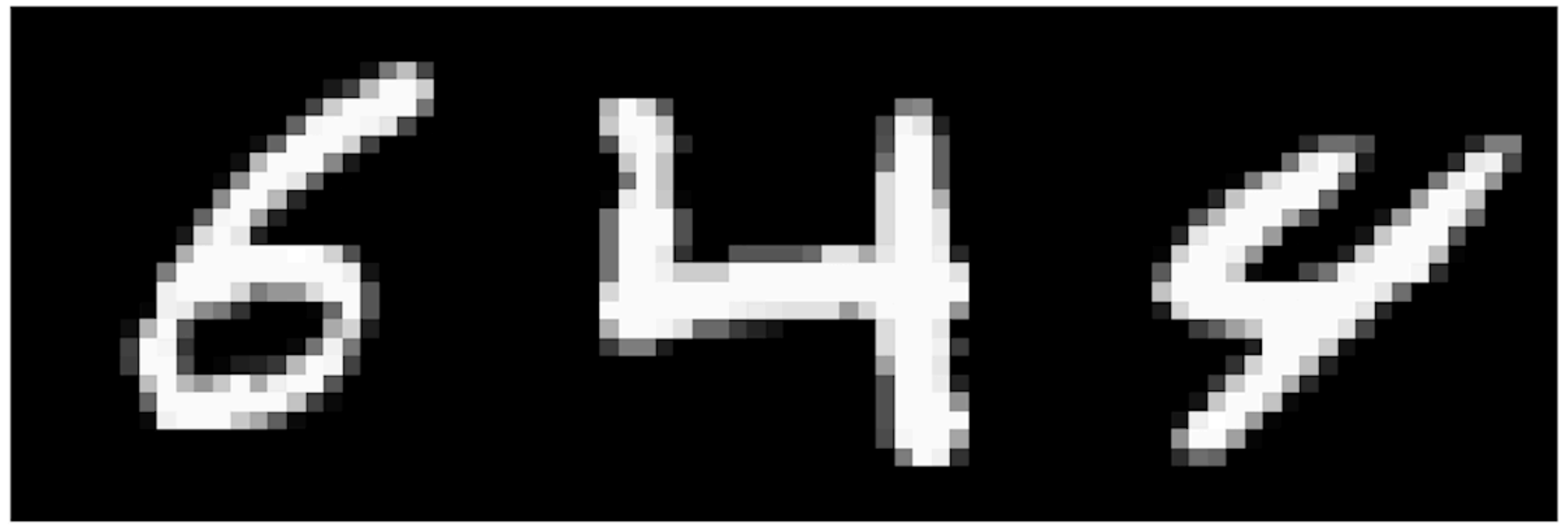
Taken from Lichess

“Can white checkmate black in 3 steps?”

Reasoning on visual tasks with a symbolic latent

- Many visual reasoning tasks have an underlying symbolic latent structure

X



y

4

"The function that the model has to learn is sum of digits mod 10"

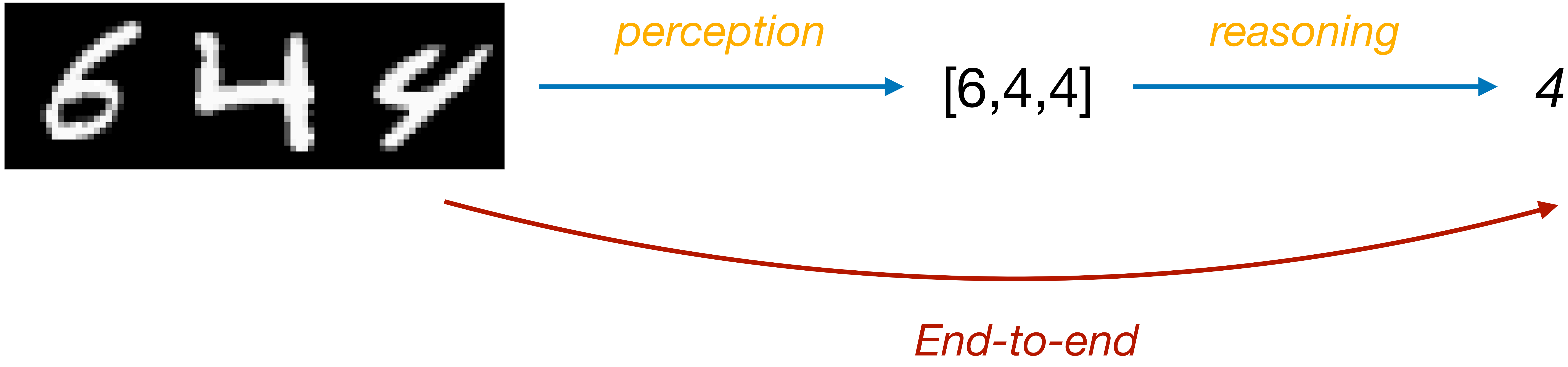
Reasoning on visual tasks with a symbolic latent

Different approaches

1) End-to-end

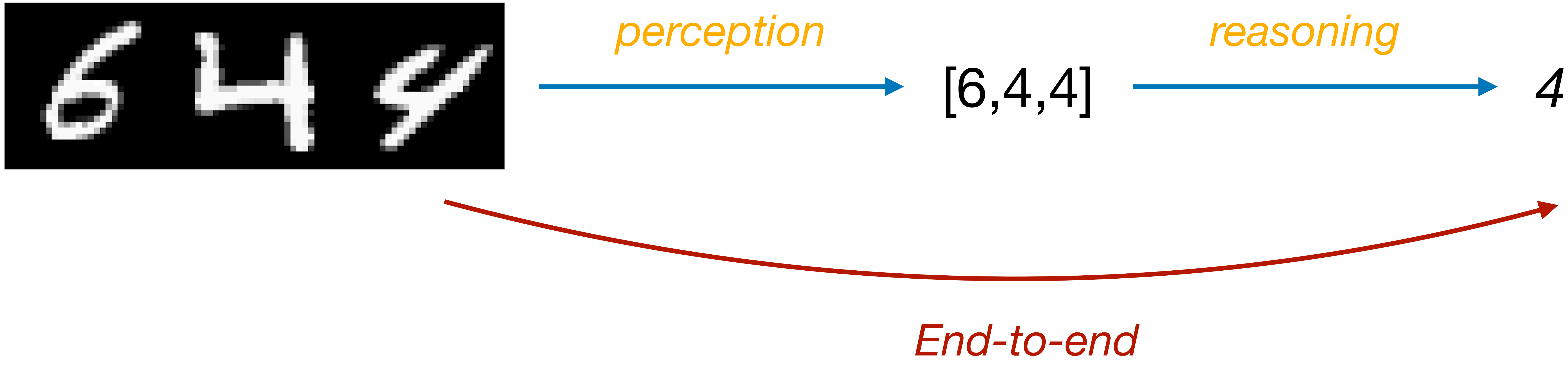
2) Separating perception and reasoning

Models can be trained from scratch or be pre-trained working with prompts.



Research questions

- What are the trade-offs?
  - In distribution: learning complexity (e.g., sample complexity)
  - OOD generalization



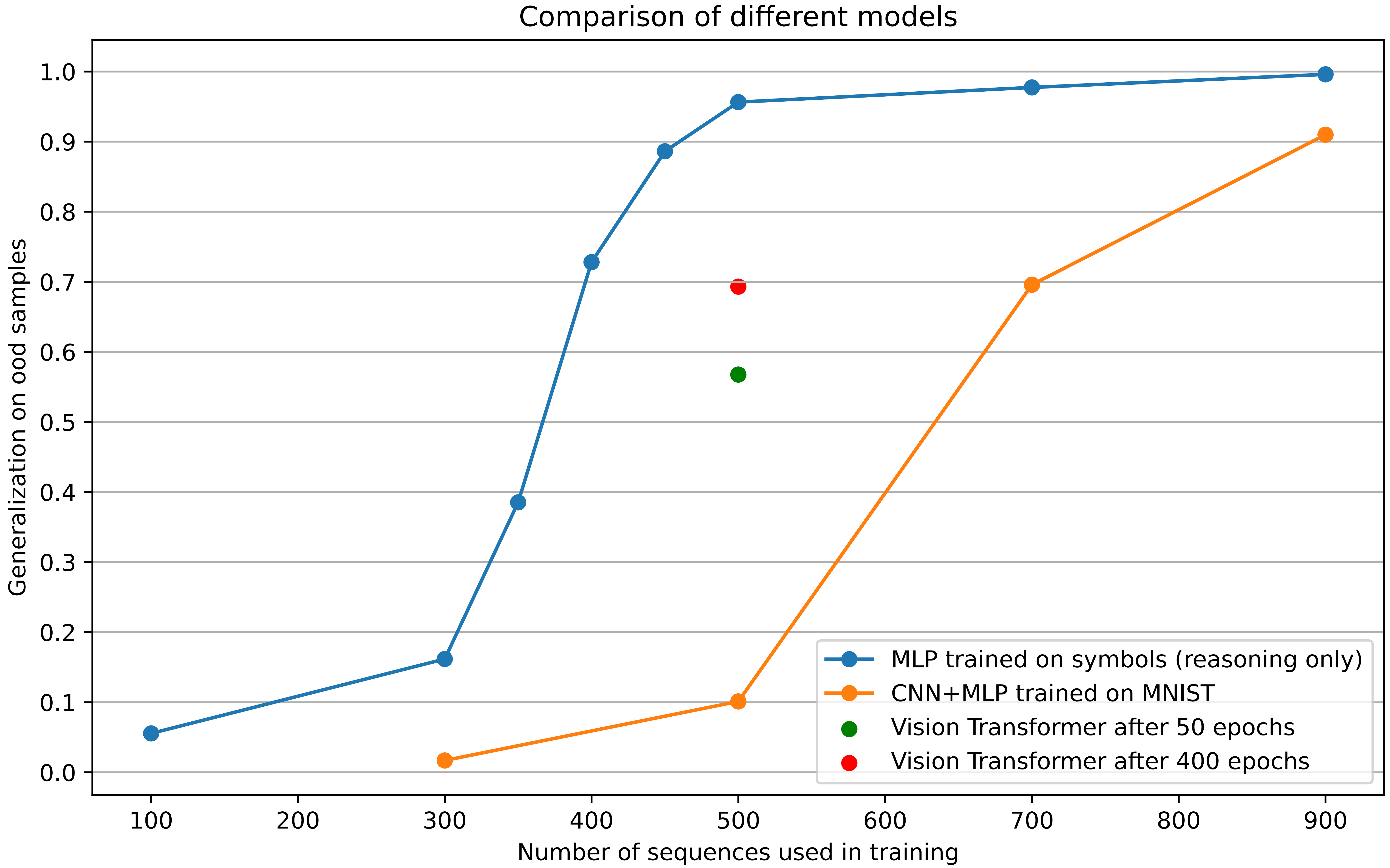
Models can be trained from scratch or be pre-trained that work with prompts.

Research questions

- Consider this *fair* scenario:
  - We choose m symbolic (latent) sequences.
  - For each, we generate many images (e.g., 10k), that becomes our dataset.
    - => Perception is not a limitation.
  - Complexity comes from the limited symbolic latent instances in the dataset.
  - We can assess learning complexity with m.

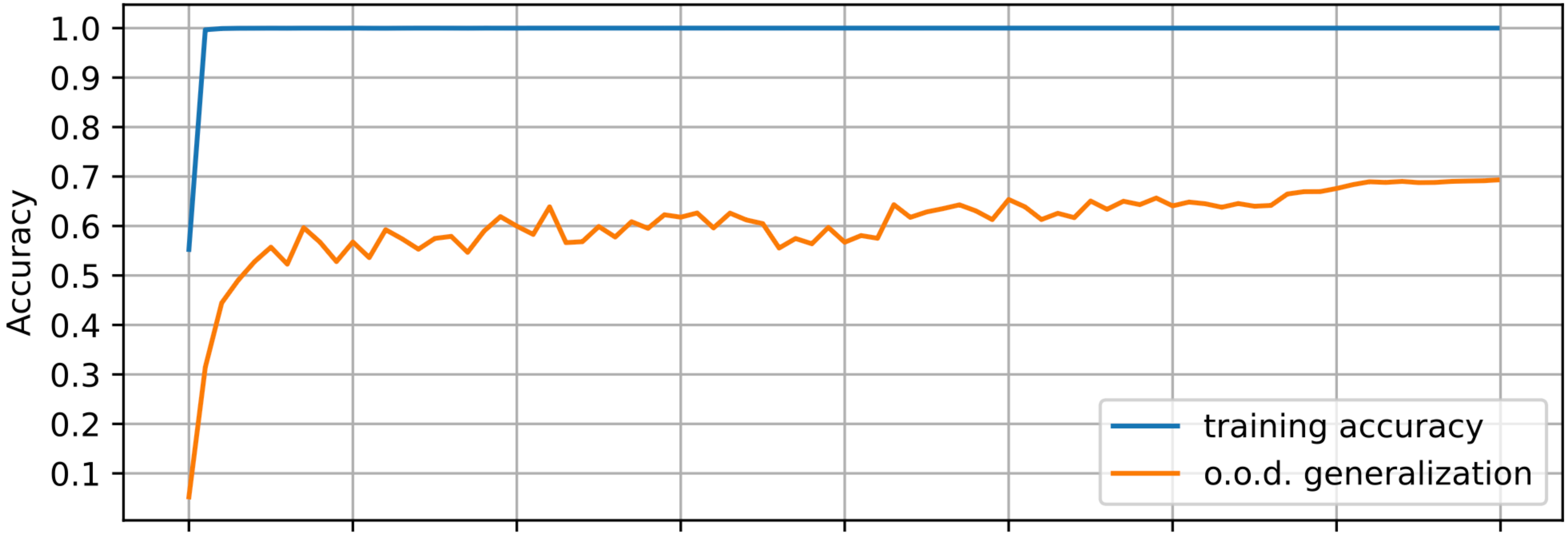
Initial observations

- Solving the task seems harder with the end-to-end approach.
- **How can we move the orange curve to the left?**



Research questions: understanding

- With a ViT, the model learns the training set rather fast, nevertheless, continuing training still improves the generalization. *(related to grokking?)*



Theory projects

1. **GOTU and different grokking regime**

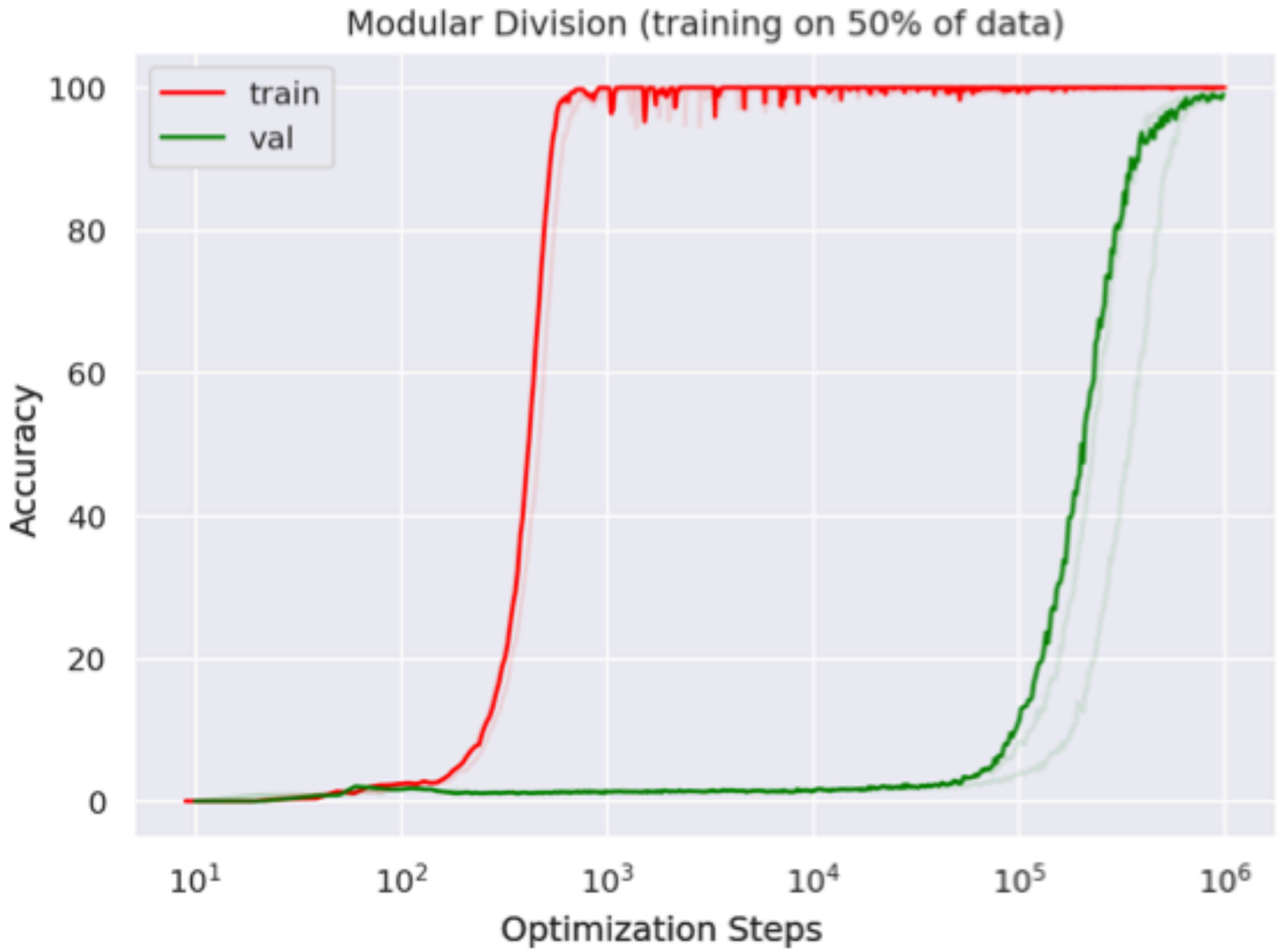
2. Interplay of initialization, speed of learning, and generalization

GOTU and different grokking regimes

Theoretical directions.

GOTU and different grokking regime

Grokking = phenomenon where models suddenly begin generalizing to unseen samples during extended training. Original paper from **Power et al. [2022]**.



GOTU and different grokking regime

Grokking was originally observed on algorithmic tasks

If (i) the target function is not "algorithmic" (i.e. does not have a simple representation) or (ii) the unseen domain is structured in a particular way

then Transformer generalizes according to a different rule, which we call "majority on the nearest neighbors".

Intuition: Transformer cannot figure out the underlying rule and has to "memorize" the function.

Question: can we propose an (informal) explanation to this? Can we prove some generalization results?

Interplay of initialization, speed of learning, and generalization

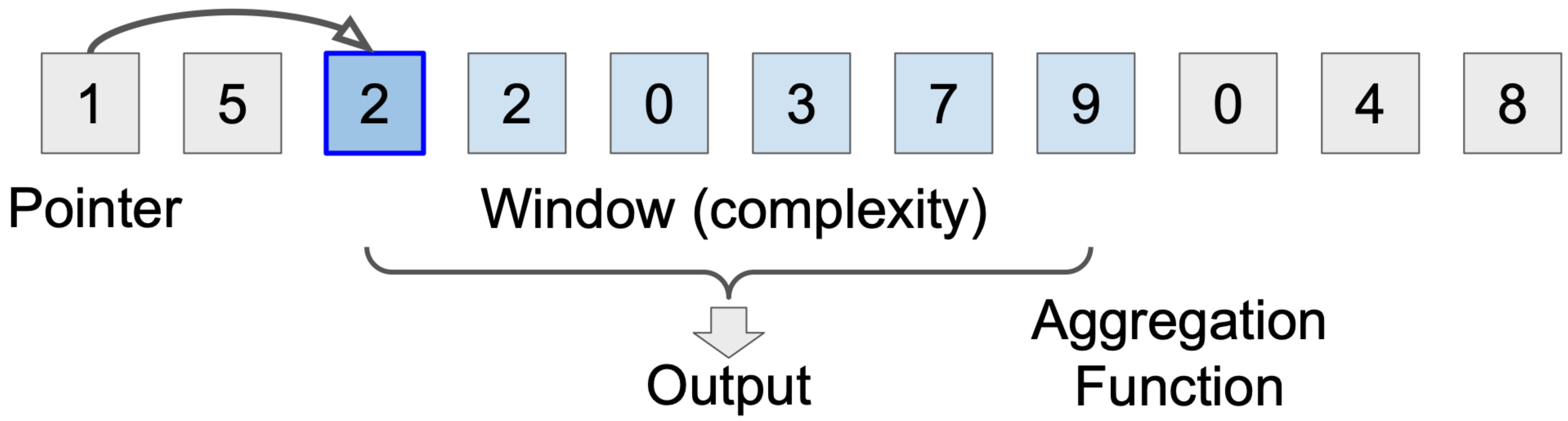
Mixture of applied and theoretical directions.

Starting with an observation

Pointer Value Retrieval: A new benchmark for understanding the limits of neural network generalization

Chiyuan Zhang*[1] Maithra Raghu*[1] Jon Kleinberg[2] Samy Bengio†[1]
[1]Google Research
[2]Cornell University

- Pointer Value Retrieval (PVR) task:



Observation of the interplay

For the same architecture and task, some initializations succeed and others don't. Interestingly, training is faster in cases that lead to generalization and slower in cases that don't generalize.

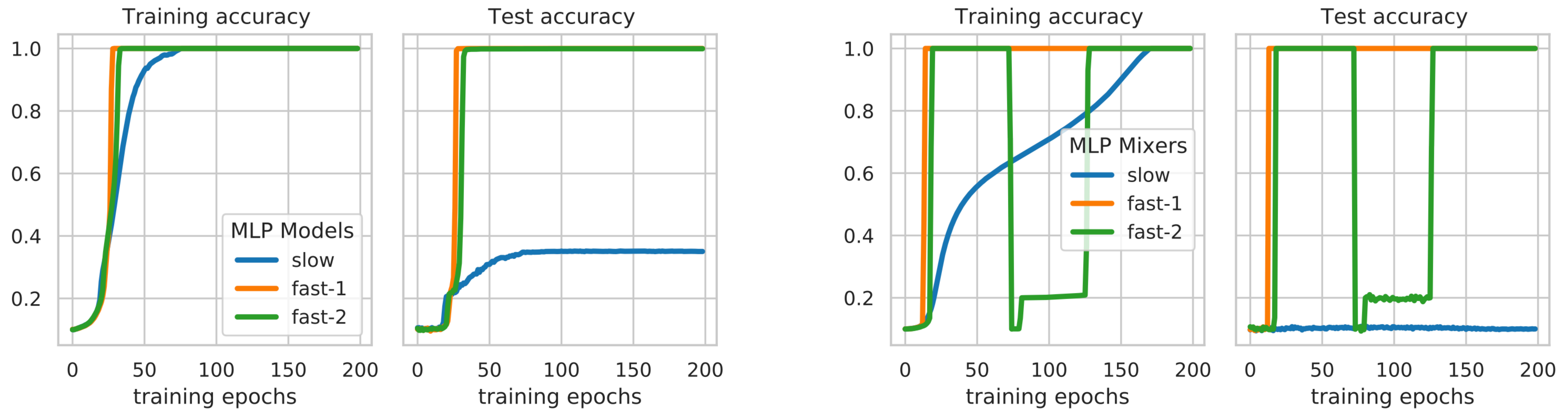
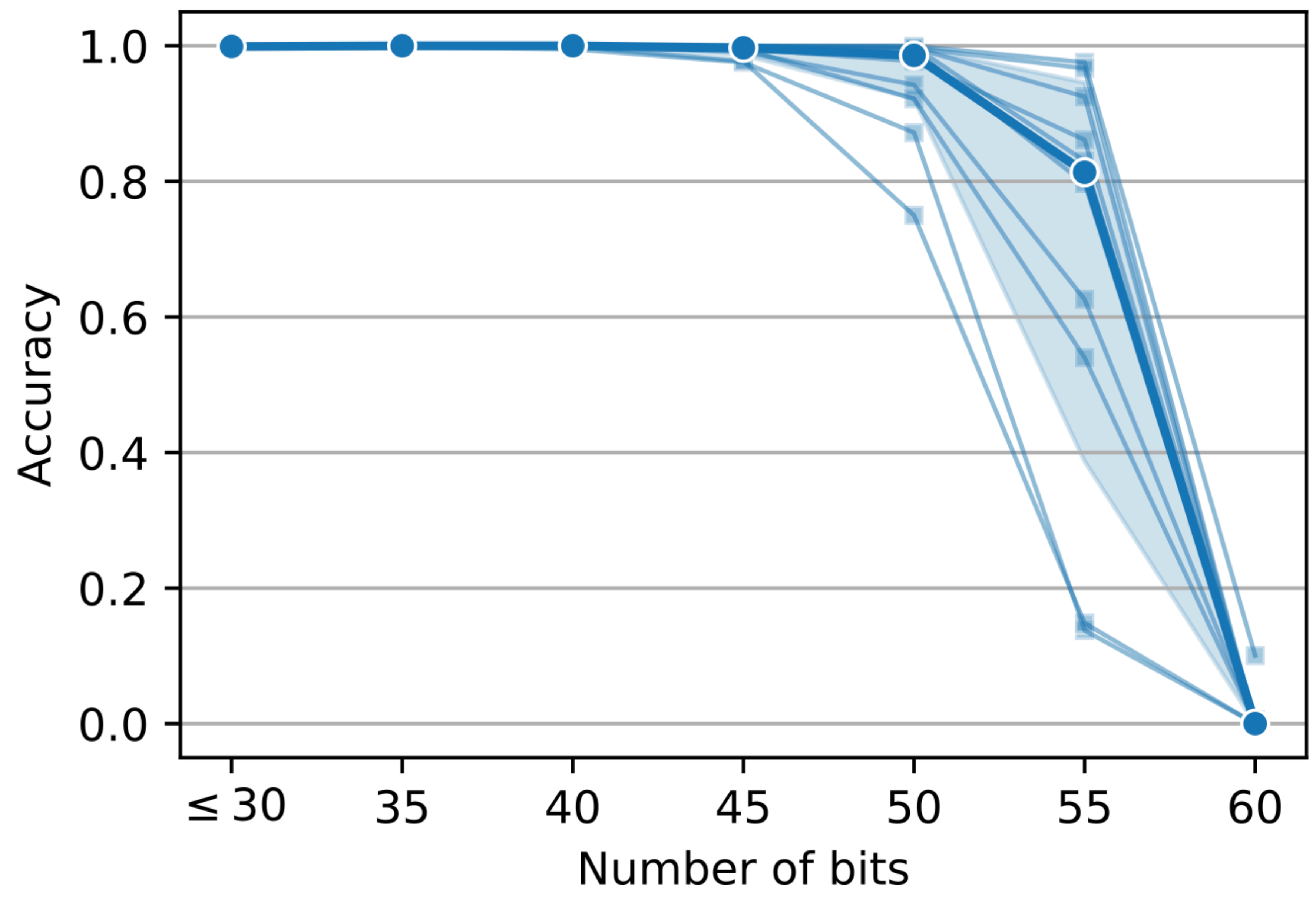


Figure 12: **Fast vs Slow learning in `MLP` and `MLP-Mixer`.** Learning curves for (left) MLPs and (right) MLP-Mixers that depict fast learners that generalize well and slow learners that generalize poorly. The MLPs are trained on 300k training examples with 2 neighbors. The MLP-Mixers are trained on 5M training examples with 5 neighbors.

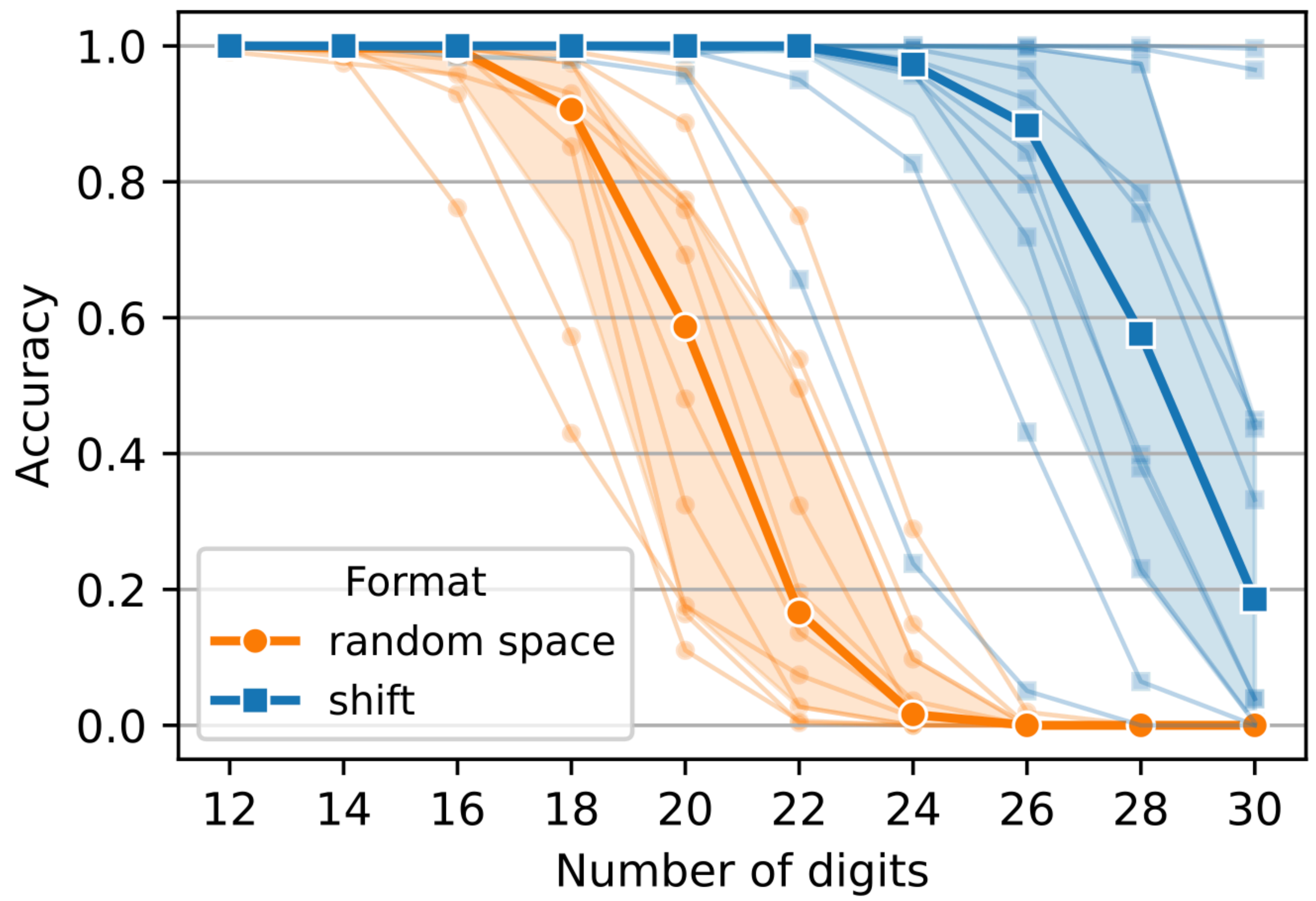
Question: can we understand/leverage this?

Seed dependence also appears in OOD generalization

- Length generalization depends heavily on the seed (task and method are fixed).



(a) Length generalization for parity when training is done up to 30 bits.



(b) Length generalization for addition where the random space method is trained up to 10 digits and the shift method is trained up to 4 digits.

Taken from [How Far Can Transformers Reason? The Globality Barrier and Inductive Scratchpad, Abbe et al., 2024]

Questions?

Please send an email to etienne.bamas@epfl.ch by **tomorrow** with your preferred project and a backup project.