Karlsruhe Institute of Technology

Neural Networks in Formal Verification

Kickoff Event

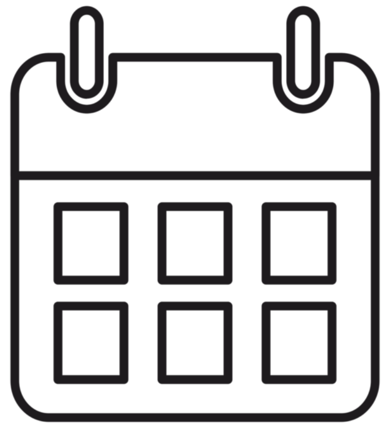
Lecturers:

Prof. Bernhard Beckert
Michael Kirsten
Samuel Teuber
Philipp Kern
Debasmita Lohar

Chair: Application-oriented Formal Verification

October 23, 2024

Logistics

Reading Assignment

- Read the assigned papers
- Find and read **1-2 additional papers** in the area (more is welcome but ***not*** required!)
- Build a foundation on the given topic

Meetings: Once in every 2 weeks (schedule appointment with your advisor)

Instruction Language: **English**

Presentations / Discussion

Time — **15:45 — 17:15** (25 + 5 + 25 + 5 + 30 minutes), Room No. **301**

January 23:

presentation 1 RL for Theorem Proving

February 6:

presentation 2 Fairness

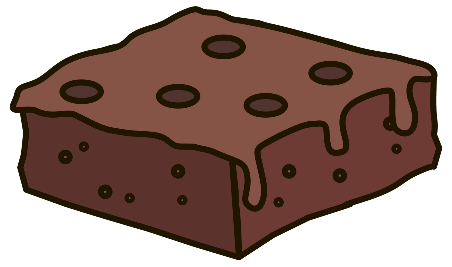
presentation 3 Robustness

February 13:

presentation 4 LLMs for Formal Specifications

presentation 5 LLMs for Program Synthesis

Discussion — Presenters are responsible for leading the discussion

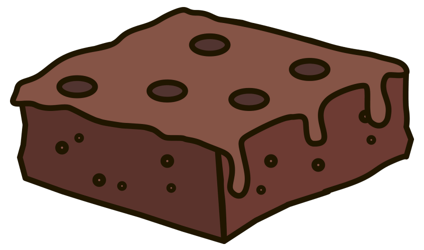
Everyone must attend,  points for contributing to the discussion!

Writing Assignment

Report: 7-8 pages, ACM Generic Journal Manuscript Format

Submission: **March 31, 2025**

- Topic: brief overview including
  - motivation,
  - different methods, their strengths and weaknesses,
  - discussion of results, and
  - conclusion
- In-class discussion: include relevant ones
- Future extensions: potential applications of your methods to other topics and/or vice versa

 points for ideas!

Guidelines for using Generative AI

For example,

✓ Polish the writing including spelling, grammar, style, translation

✗ Generate new ideas, e.g., the future extension in the report

> "In all cases, students remain responsible for their work. This also applies to the parts of their work that have been created using or influenced by AI."

https://www.informatik.kit.edu/downloads/studium/Guidelines_Generative_AI_Informatics.pdf

Distribution of Points

- Presentation: 60%
- Report: 30%
- Bonus (in-class discussion): 5%
- Bonus (future extensions in the report): 5%

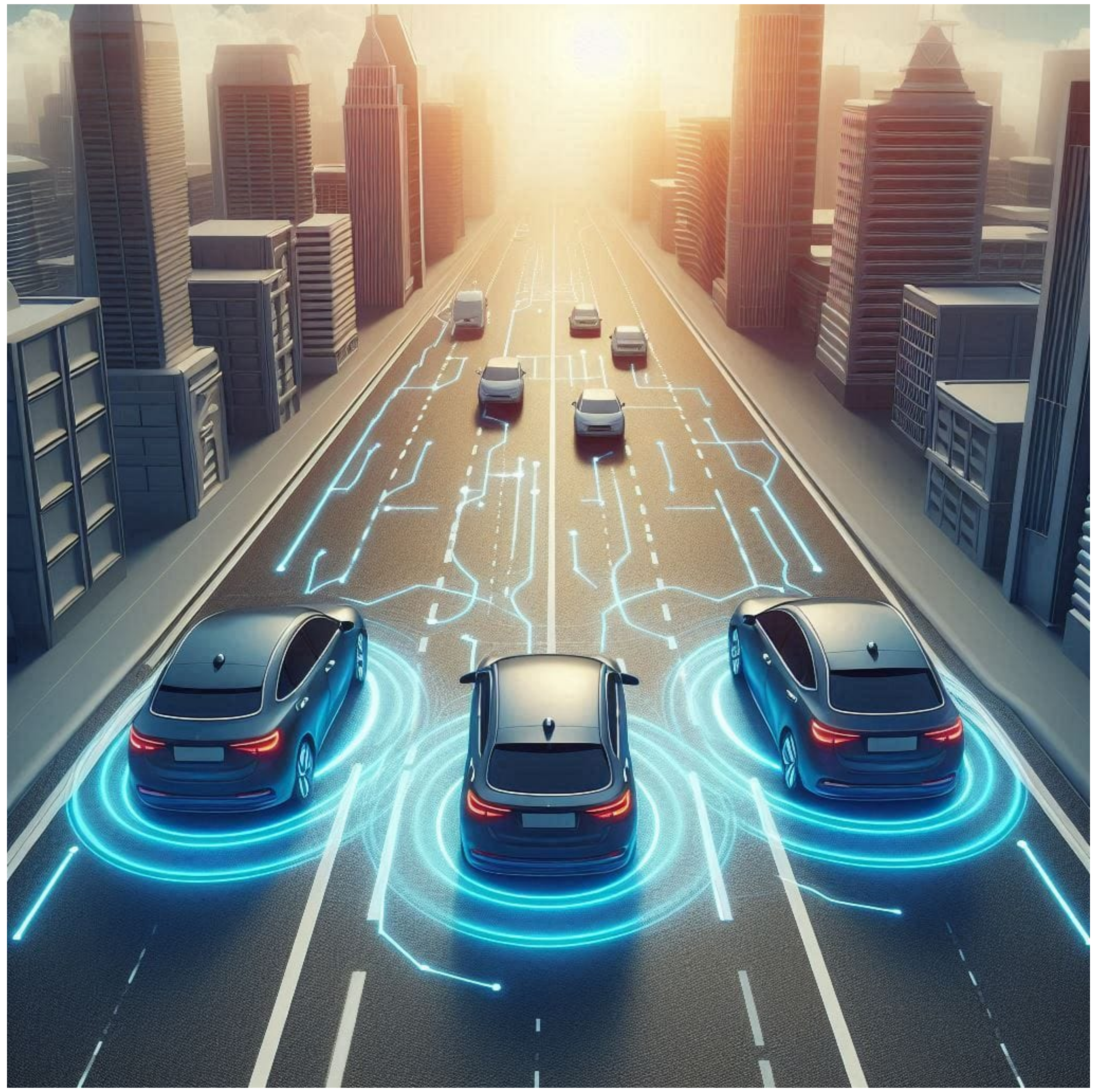
Note: Everybody needs to submit the report to pass!

About the Topics

- [x] Verification of Neural Networks
- [ ]

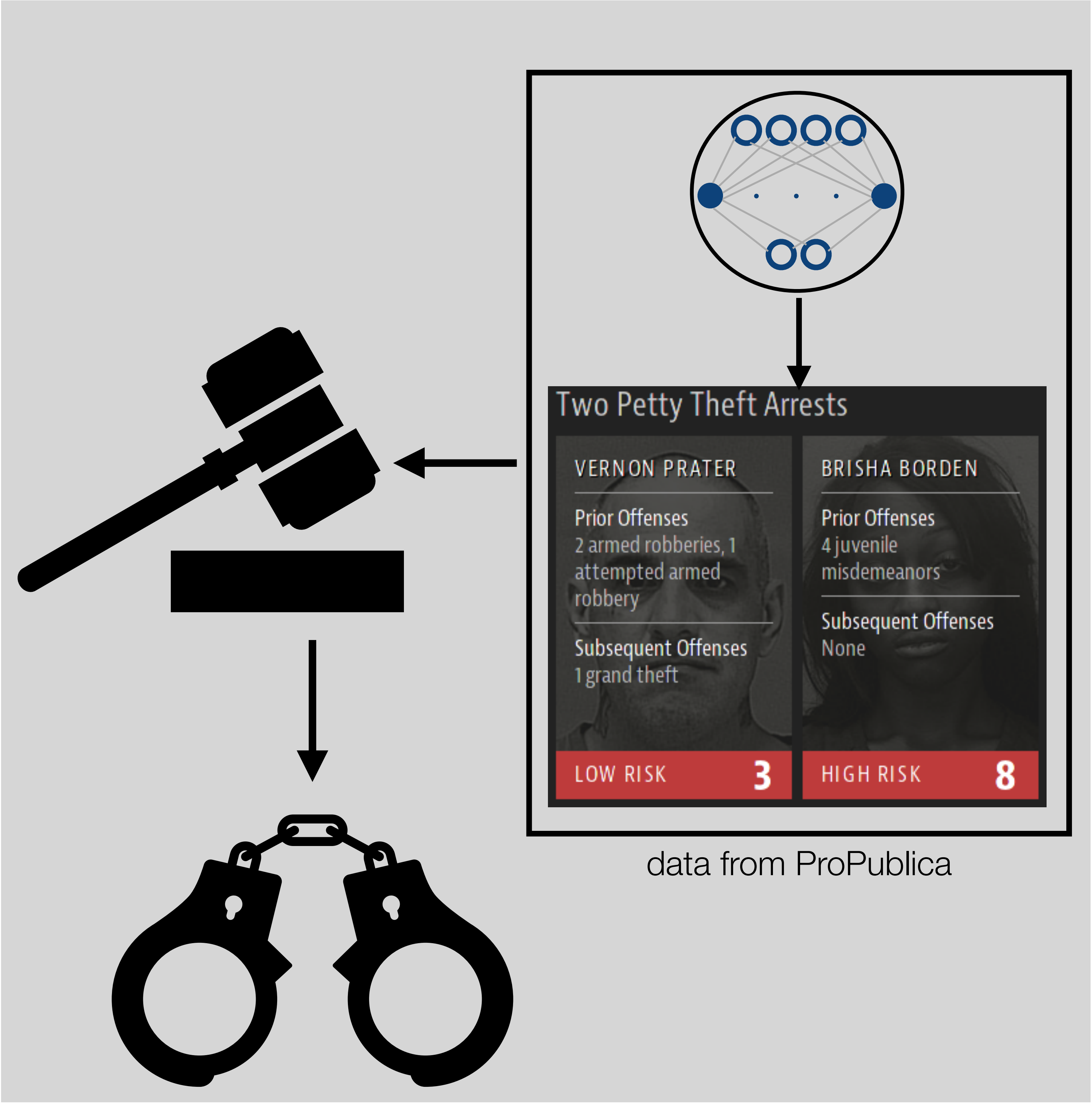
Do NNs in critical systems work as intended?

*images created with copilot



*Autonomous Driving

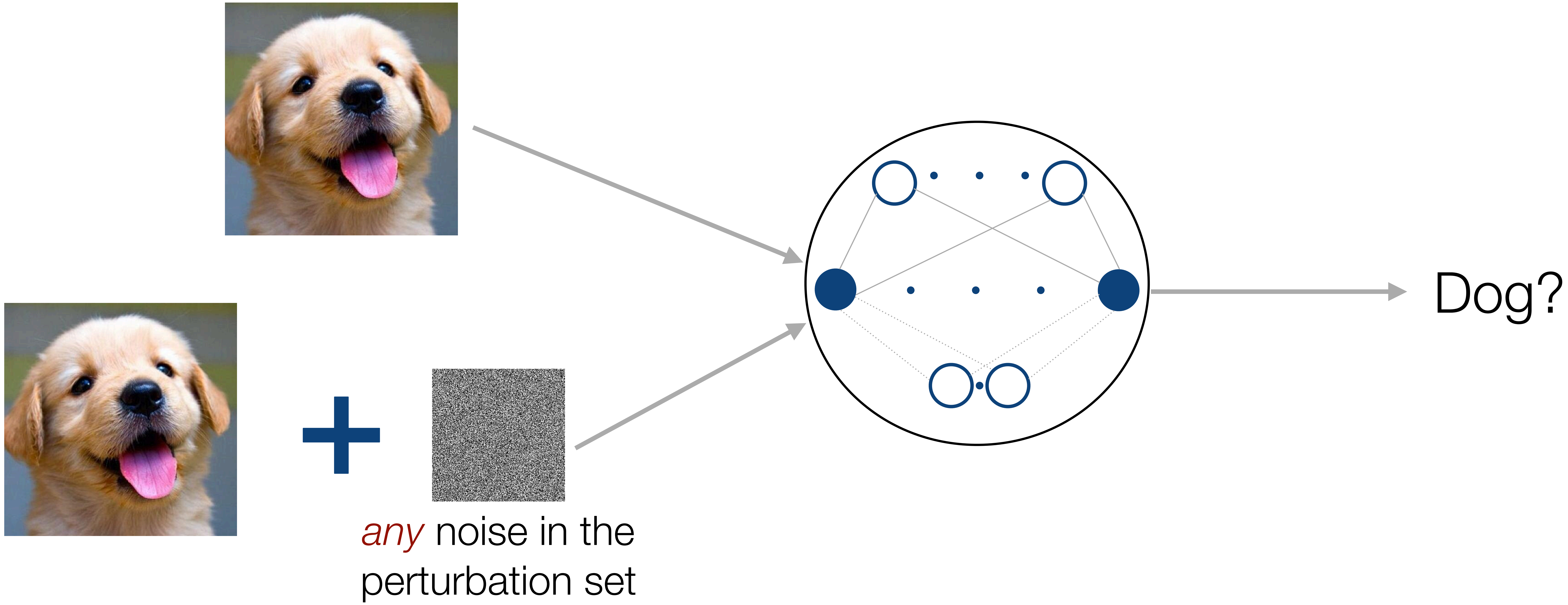
*AI-powered Medical Equipment



AI-powered Decision Making

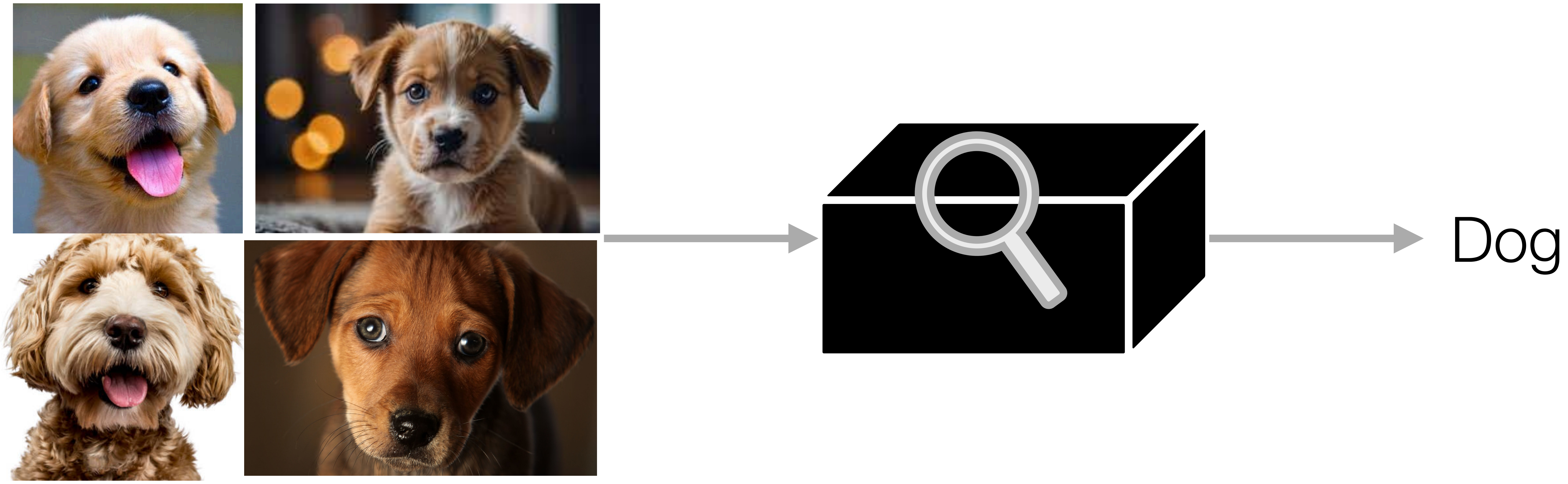
We aim to **prove** that NNs have certain desired properties!

Property 1: Robustness (Advisor: Philipp)



- How can we verify if the network is robust to input perturbations?
- What are the challenges in this verification problem?

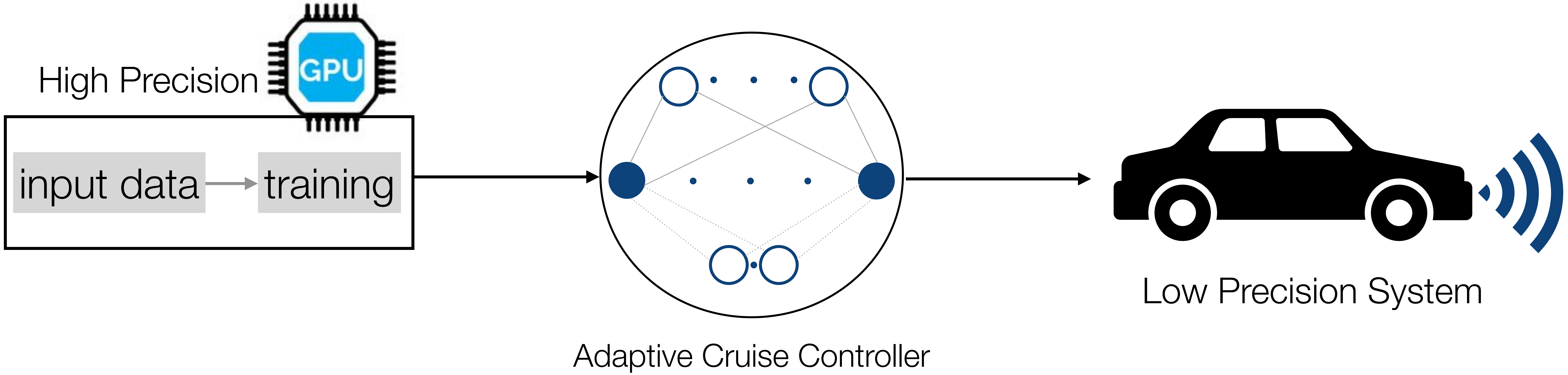
Property 2: Explainability (Advisor: Philipp)



Goal: explain the decision!

- What makes a good explanation, and how can we compute it efficiently?
- Which parts/features of inputs are the most critical for prediction?

Property 3: Quantization (Advisor: Debasmita)



Quantization trades off precision for improved efficiency!

- How many bits are sufficient to ensure the safety of the quantized model?
- How can we scale the verification process efficiently?

Property 4: Fairness (Advisor: Samuel)

ML algorithms make critical predictions
but studies have shown potential **biases**!

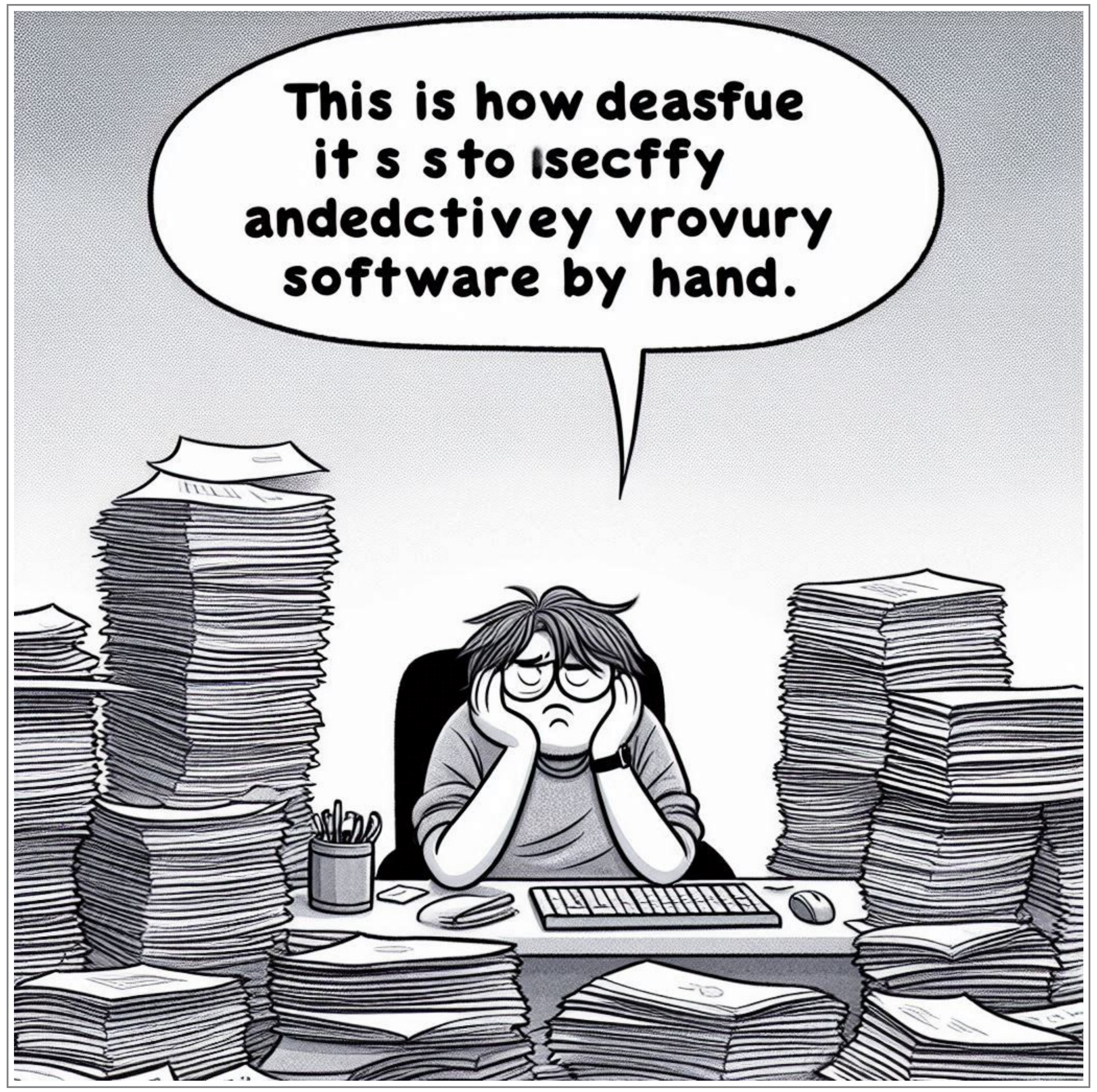
data from ProPublica

- How do we analyze influence of protected attributes?
- How do we verify at inference time?

About the Topics

- ☑ Verification of Neural Networks
- ☑ Neural Networks for Verification

How can we use NNs to reduce verification efforts?



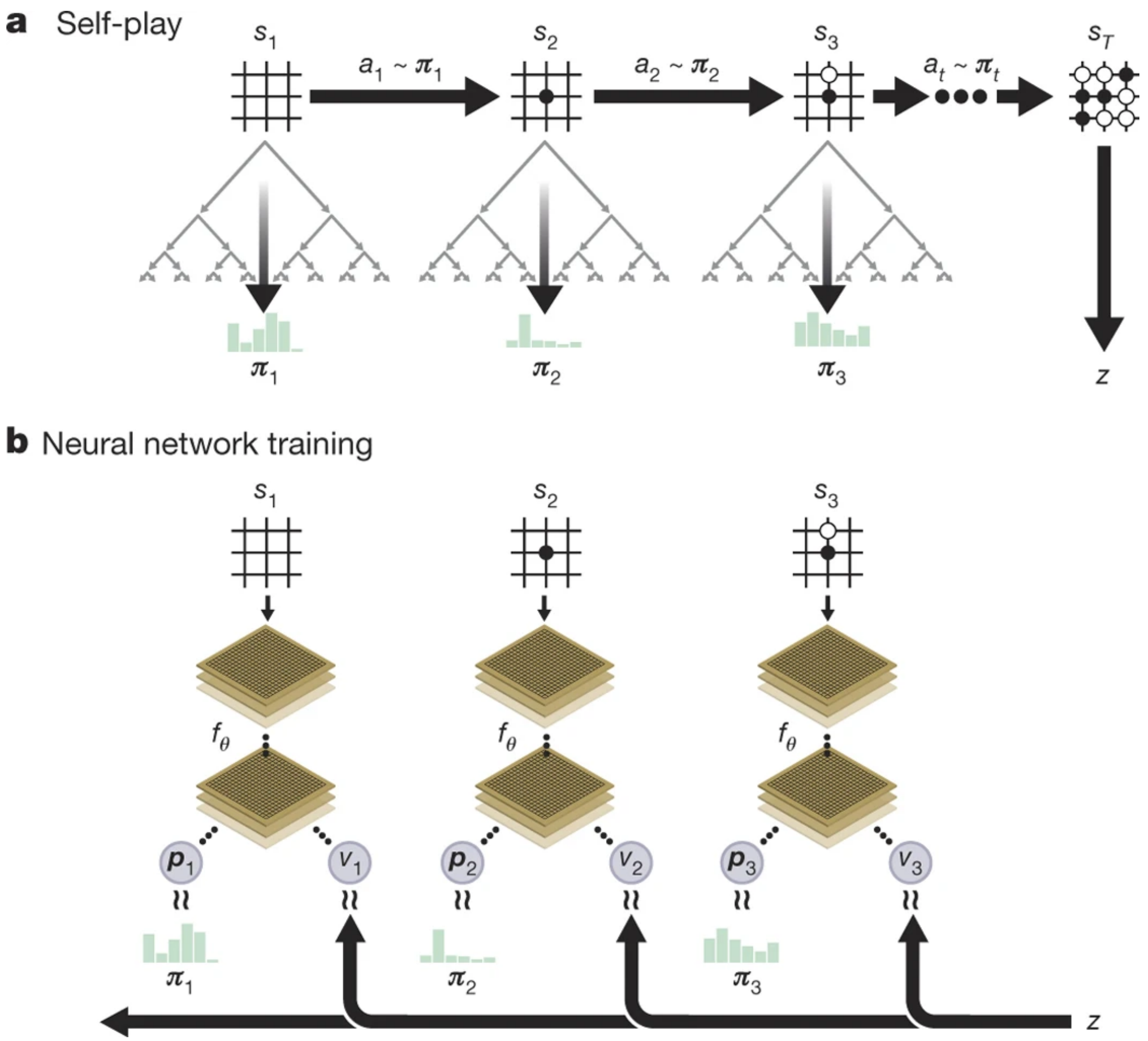
Prompt: *Generate a picture showing how tedious it is to specify and deductively verify software by hand.*
The picture should show how terribly boring the job is. So much so, that anyone seeing the picture is scared of doing this task by hand.

Prompt: *Now create an image demonstrating how incredibly easy the task becomes if you use clever AIs like yourself.*
The person should be extremely happy. Anyone seeing the image should be motivated to verify their software using AI.

But, AI is not as reliable as we would like it to be…

Images are created with copilot!

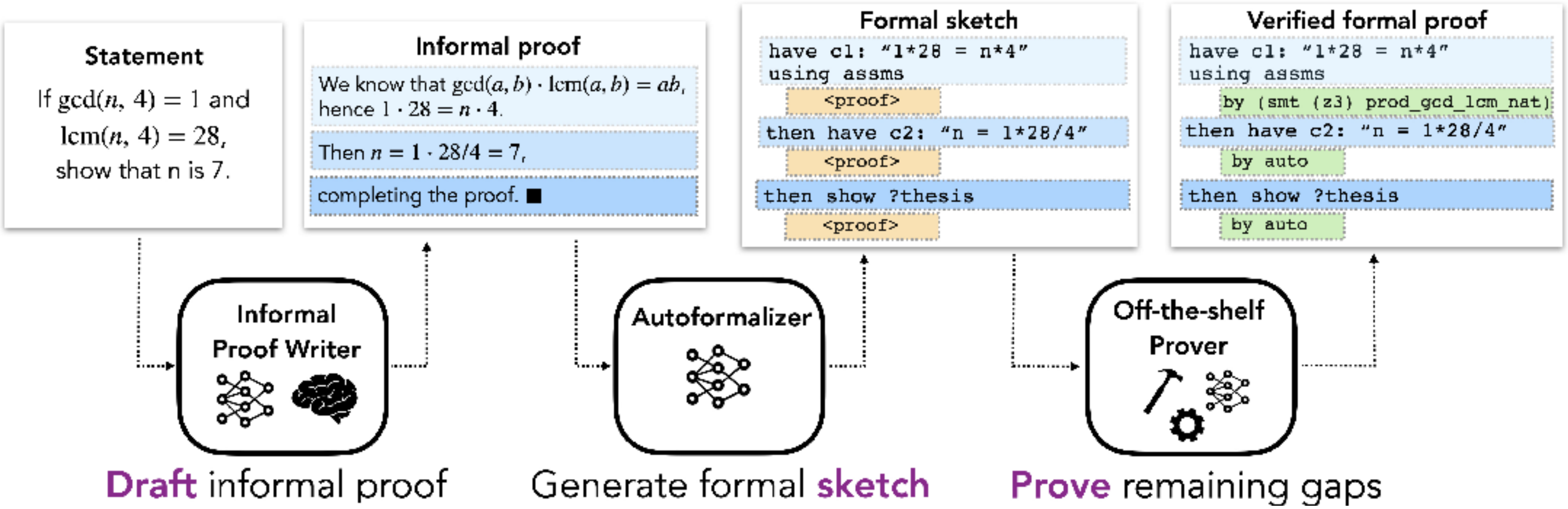
Reinforcement Learning (RL) for Theorem Proving (Advisor: Samuel)



Silver, David, et al. "Mastering the game of go without human knowledge." Nature 550.7676 (2017): 354-359

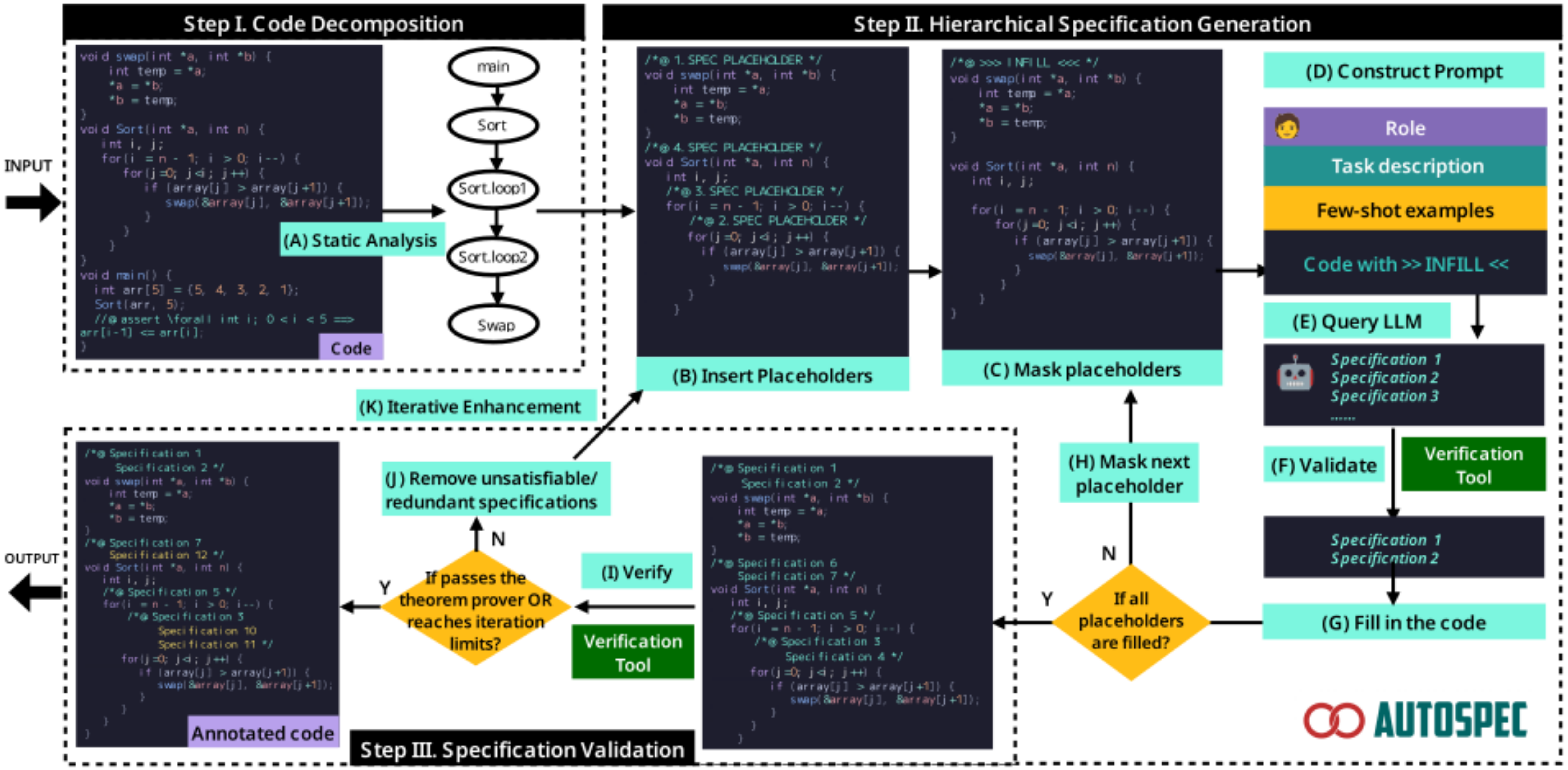
- Problem Solving as a game that RL agent can win
- Generation of training problem is just another game

Large Language Models (LLM) for Theorem Proving (Advisor: Michael)



- Can LLMs help in drafting formal theorems and proofs, and finalize existing proofs?
- Can LLMs come up with valid new theorems from existing proof libraries?

Large Language Models (LLM) for Formal Specifications (Advisor: Michael)

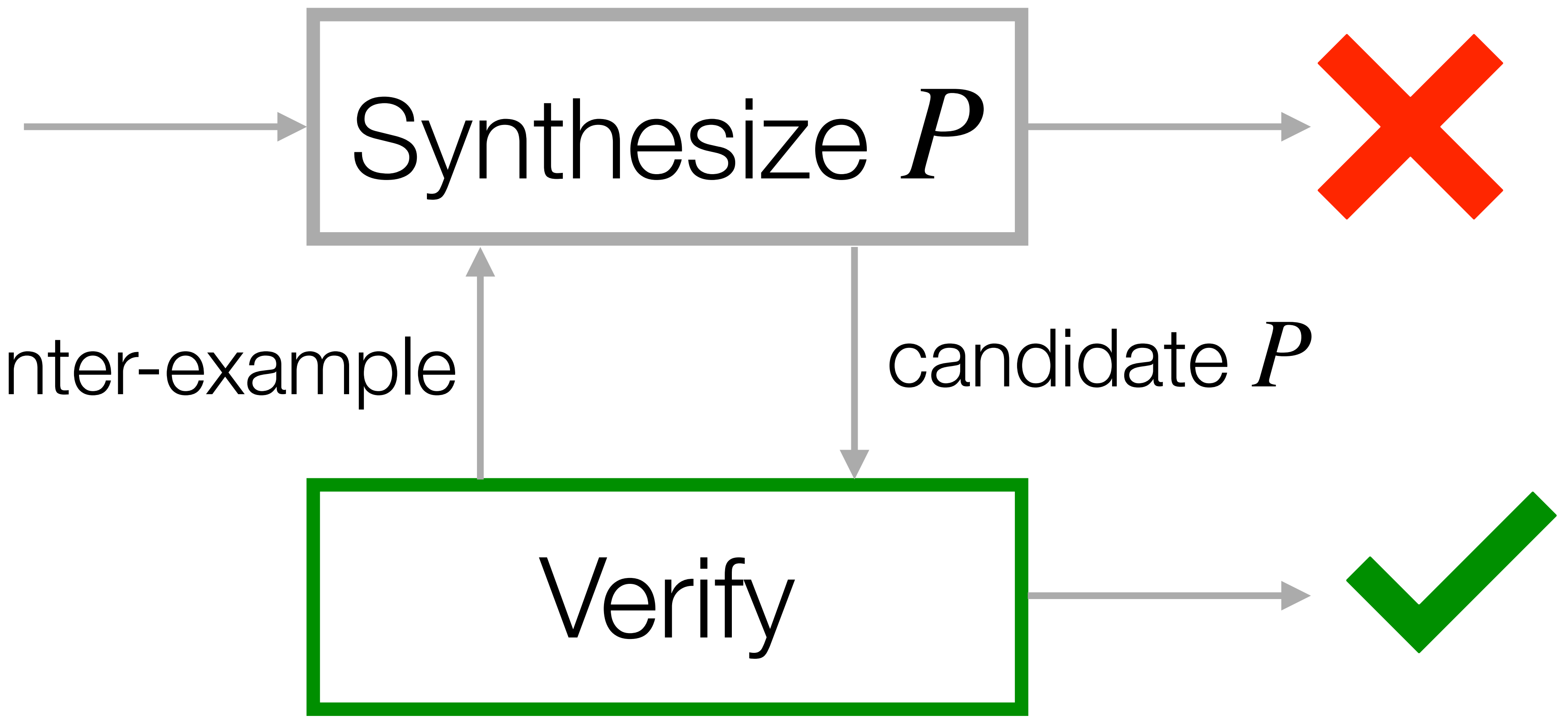


- Can LLMs provide intermediate specifications e.g., loop invariants?
- Can LLMs provide further specifications to guide program verification tools?

Large Language Models (LLM) for Program Synthesis (Advisor: Debasmita)

$$\exists P \forall x \, . \, \sigma(P, x)$$

Does there exist a function P such that
for all possible inputs x,
the specification σ will evaluate to true for P and x?



- How can LLMs assist in synthesizing specific code?
- How can we ensure the generated code is correct and consistent?