STAT 453: Introduction to Deep Learning and Generative Models

Ben Lengerich

Lecture 21: Reasoning Models and RL from Verifiable Rewards

November 19, 2026

Logistics

- HW4 due tomorrow (Fri 11/20); HW5 due Fri 12/4
- Project presentations: Tue 12/1 and Thu 12/3
  - Sign up for a slot if you haven't yet
  - 4-minute talks; upload slides to Canvas 24 hours before your slot
- Project final report due Tue 12/8
- Final exam: Saturday, December 12, 5:05–7:05 PM

Ben Lengerich © University of Wisconsin-Madison 2026

Last Time: Aligning LLMs

- **SFT:** imitate demonstrations (loss on response tokens only)
- **Reward model:** Bradley–Terry on preference pairs; only reward differences matter
- **RLHF:** maximize $r_\phi - \beta\, \mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})$; policy gradient on the model's own samples
  - Push too hard and the policy exploits the reward model (over-optimization)
- **DPO:** the same objective as a classification loss on pairs; no reward model, no RL loop

Today: Reasoning Models

- **Thinking at test time:** chain of thought, sampling many answers, voting
- **RL from verifiable rewards:** replace the reward model with a checker
- **GRPO:** policy gradient where the group of samples is the baseline
- **DeepSeek-R1 and o1:** what emerged, and the full training recipe
- **Open questions:** new reasoning or sharpened sampling? Is the chain of thought faithful?

Thinking at Test Time

Scaling Test-Time Inference

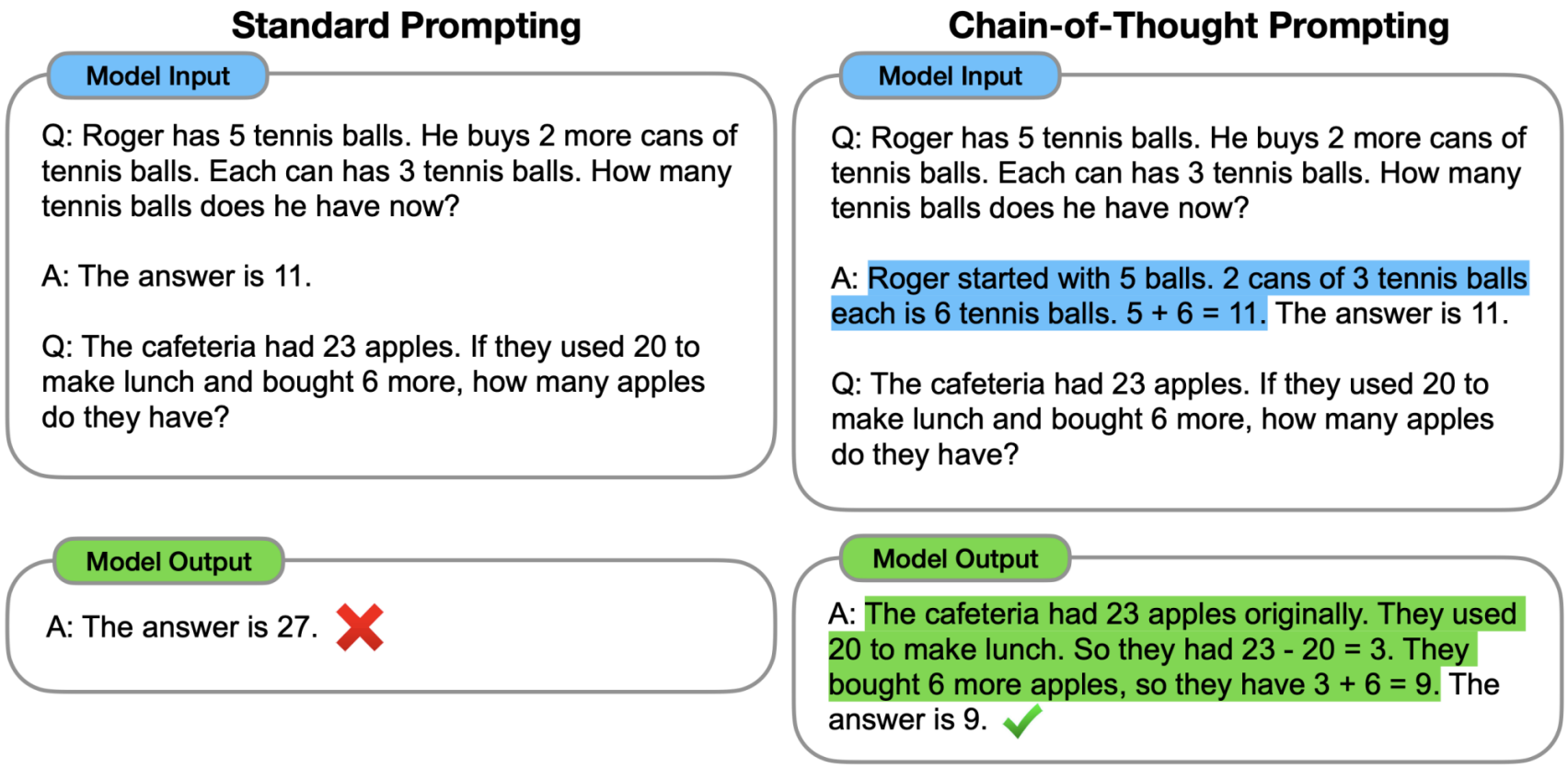
Accuracy can scale with **inference compute**, not just model size.

Test-time scaling techniques:

- Chain-of-thought reasoning
- Self-consistency sampling
- Tool use / search
- Verifiers: sample many answers, keep the best

Chain-of-Thought

Asking the model to generate reasoning tokens improves problem-solving accuracy.



Wei, et al. (2023) Chain-of-Though Prompting Elicits Reasoning in LLMs

Ben Lengerich © University of Wisconsin-Madison 2026

Why does writing out steps help?

Answer immediately

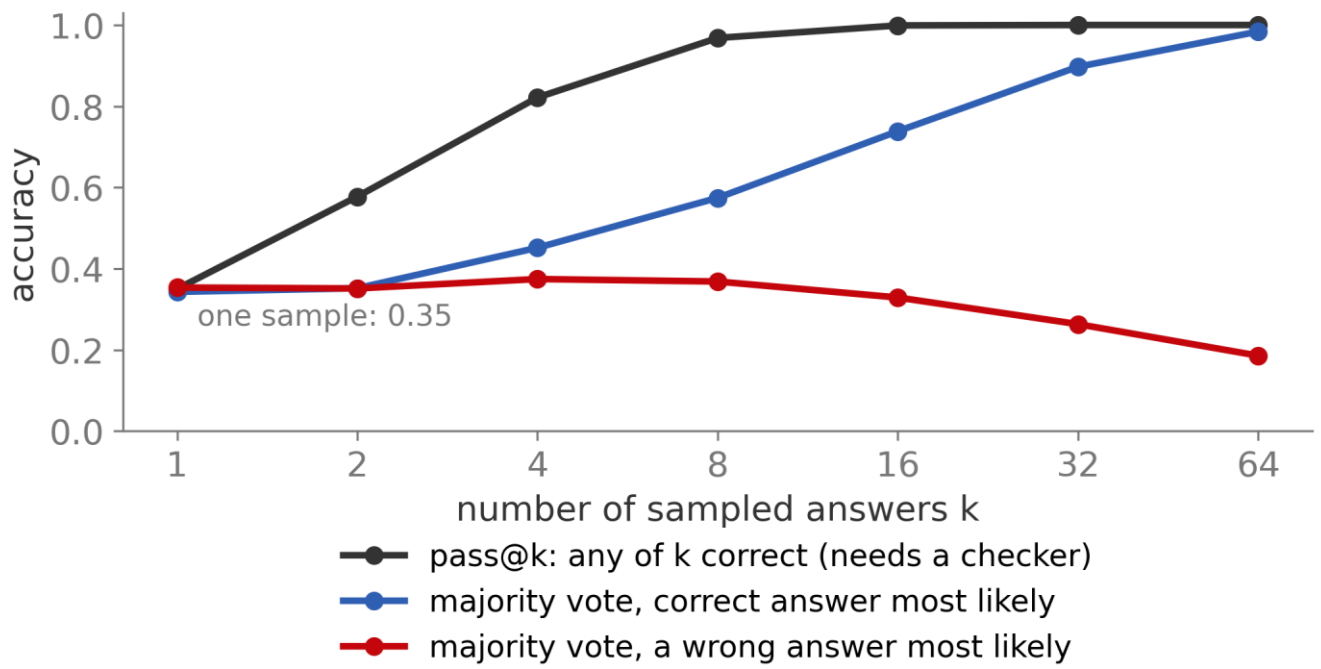
"23 × 47 = " → one forward pass
to produce the answer

Write it out

"23 × 47 = 23 × 40 + 23 × 7 = 920 + 161 = 1081"
one forward pass per token

- Each token gets a **fixed amount of computation** (L layers)
- Every generated token is **another forward pass** that can read all earlier tokens
- Intermediate results live in the text: a **scratchpad**
- So more tokens = more sequential computation: compute spent at **test time**

Sampling many answers: best-of-k and majority vote



- **With a checker:** keep any correct sample
  - pass@k = $1 - (1 - p)^k$
  - p = 0.35, k = 8 → 0.97
- **Without one:** majority vote ("self-consistency")
  - Converges to the model's most likely answer: helps only if that answer is right

RL from Verifiable Rewards

From RLHF to RLVR: swap in a checker

$$r(x, y) = \mathbf{1}\big[\text{final answer in } y \text{ passes the check}\big]$$

Task	Checker
Math with a known answer	Compare the final answer
Code	Run the unit tests
Proofs / logic	Proof checker
Output format	Pattern match

- Same objective as RLHF: maximize reward, KL to π_{ref}
- **No learned reward model** to over-optimize
- Unlimited labels wherever answers can be checked
- Can’t grade an essay or a tone: those still need RLHF

What can you verify?

Which of these could we train with a verifiable reward?

Sudoku · SQL queries · French translation · summarizing a paper · diagnosis from a chest X-ray report · a math proof

- For each one you can verify: what exactly does the checker compute?
- How could a model get the reward without doing the task?
- For the ones you can't verify: what would you use instead?

Outcome rewards vs. process rewards

Outcome reward

- Grade only the final answer
- Cheap: a rule or a known answer
- Sparse: a long chain gets one bit of feedback
- A lucky wrong chain still gets credit

Process reward

- Grade every step
- Dense feedback: says where it went wrong
- Needs step labels or a learned step grader
- A learned grader can be hacked

- Lightman et al. 2023: process supervision beat outcome supervision on MATH (800K step labels)
- DeepSeek-R1: outcome rewards from rules only, to avoid reward hacking

GRPO: the group is the baseline

$$A_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G)} \qquad \text{for } G \text{ answers } y_1, \ldots, y_G \sim \pi_\theta(\cdot \mid x)$$

- Policy gradient with baseline b (Tuesday): b = **the group's average reward**
- No value model to train (PPO needs one as big as the policy)
- Raise log $\pi_\theta(y_i|x)$ when $A_i > 0$, lower it when $A_i < 0$

Rewards (G = 4)	Advantages
1, 0, 0, 1	+1, −1, −1, +1
1, 0, 0, 0	+1.7, −0.6, −0.6, −0.6
1, 1, 1, 1	0, 0, 0, 0 (no signal)

Problems that are always or never solved teach nothing: train on problems of the right difficulty

Shao et al., 2024 (DeepSeekMath)

DeepSeek-R1 and o1

DeepSeek-R1-Zero: RL straight from the base model

Prompt template: … The assistant first thinks about the reasoning process and then provides the answer, inside <think> … </think> and <answer> … </answer> tags.

- Base model + GRPO, **no SFT first**
- Rule-based rewards only:
  - accuracy: is the final answer right? do the tests pass?
  - format: is the reasoning inside <think> tags?
- No neural reward model: it “may suffer from reward hacking”

AIME 2024	accuracy
Start of RL (pass@1)	15.6%
End of RL (pass@1)	71.0%
+ majority vote	86.7%

DeepSeek-AI, 2025

What emerged: longer thinking, and an “aha moment”

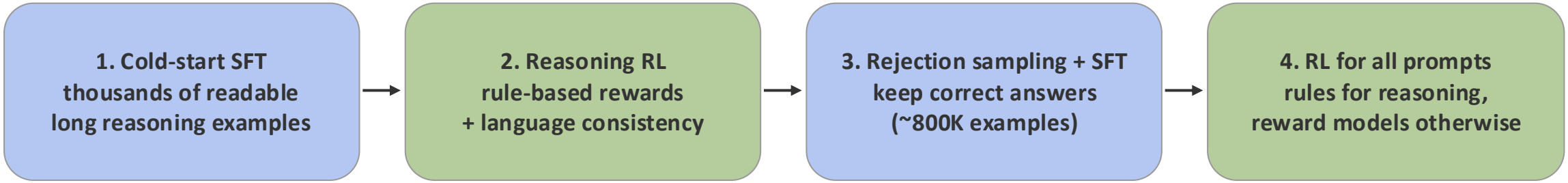
- Nobody rewarded length, yet **responses grew** from hundreds to thousands of tokens
- Reflection and trying alternative approaches **appeared on their own**
- The only reward: was the final answer right?
- Side effects: hard-to-read reasoning, **mixed languages**

“… Wait, wait. Wait. That’s an aha moment I can flag here. Let’s reevaluate this step-by-step …”

DeepSeek-R1-Zero, intermediate checkpoint

Longer reasoning = more test-time compute, learned rather than prompted

DeepSeek-R1: the full recipe



- Tuesday's stack, twice: **SFT → RL → SFT → RL**
- Stage 1 fixes R1-Zero's readability; stage 3 is SFT on the model's own **filtered** samples
- Verifiable rewards where answers can be checked, human-preference reward models where they can't

DeepSeek-AI, 2025

For small models, distillation beats RL

Qwen-32B, AIME 2024	pass@1
Large-scale RL on the base model	47.0
SFT on DeepSeek-R1's reasoning traces	72.6

- Distillation (Tuesday) = SFT on a stronger model's outputs
- The big model **discovers** reasoning patterns; small models **copy** them cheaply
- Many open reasoning models are distilled this way

DeepSeek-AI, 2025

o1: a second axis for scaling

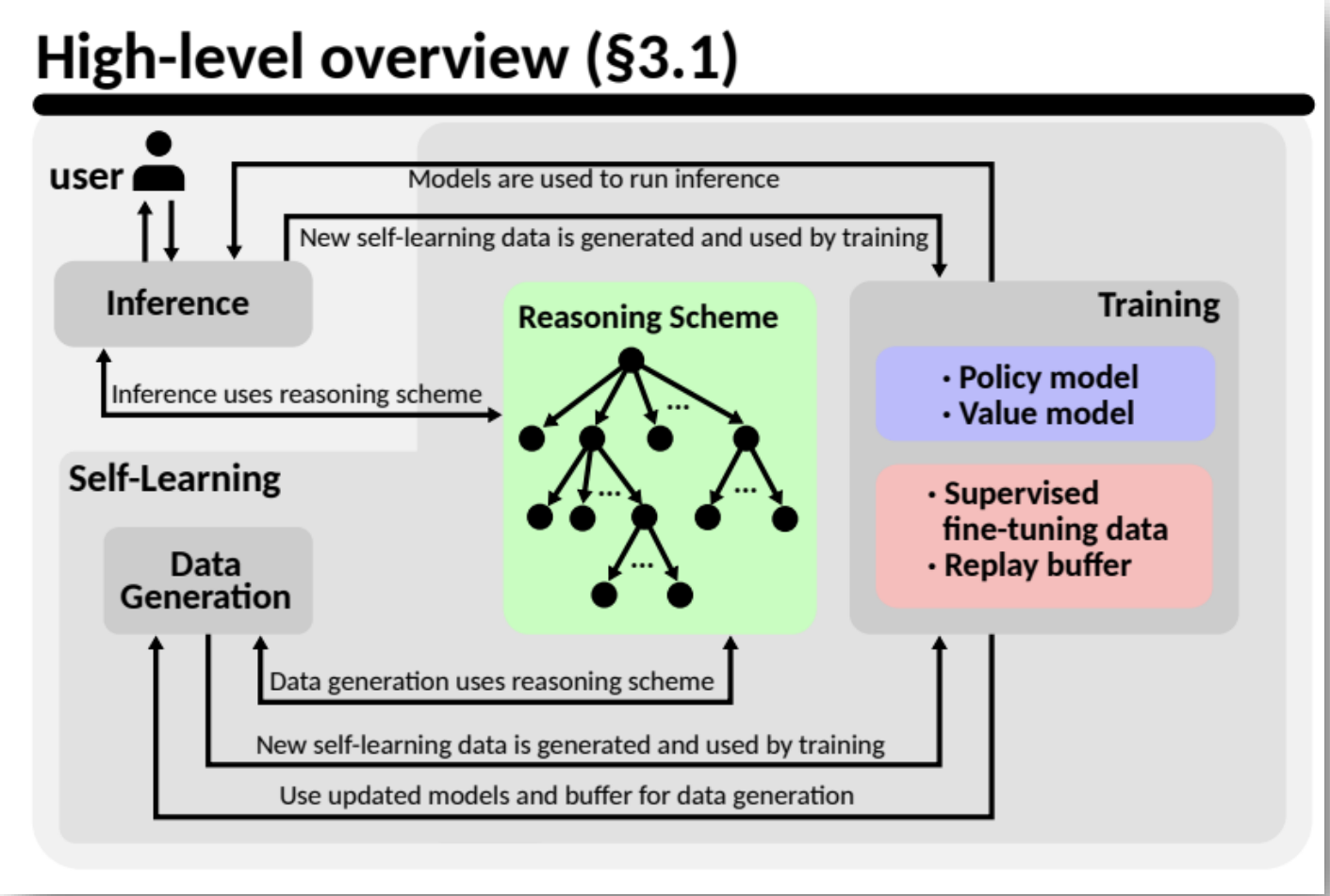
- Accuracy improves with **more RL** (train-time compute)
- … and with **more thinking** (test-time compute)
- Earlier lectures: scale parameters and data. Now: scale **tokens spent per question**

AIME 2024	accuracy
GPT-4o	12%
o1, one sample	74%
o1, vote over 64	83%
o1, rerank 1000	93%

OpenAI, 2024

Reasoning Models

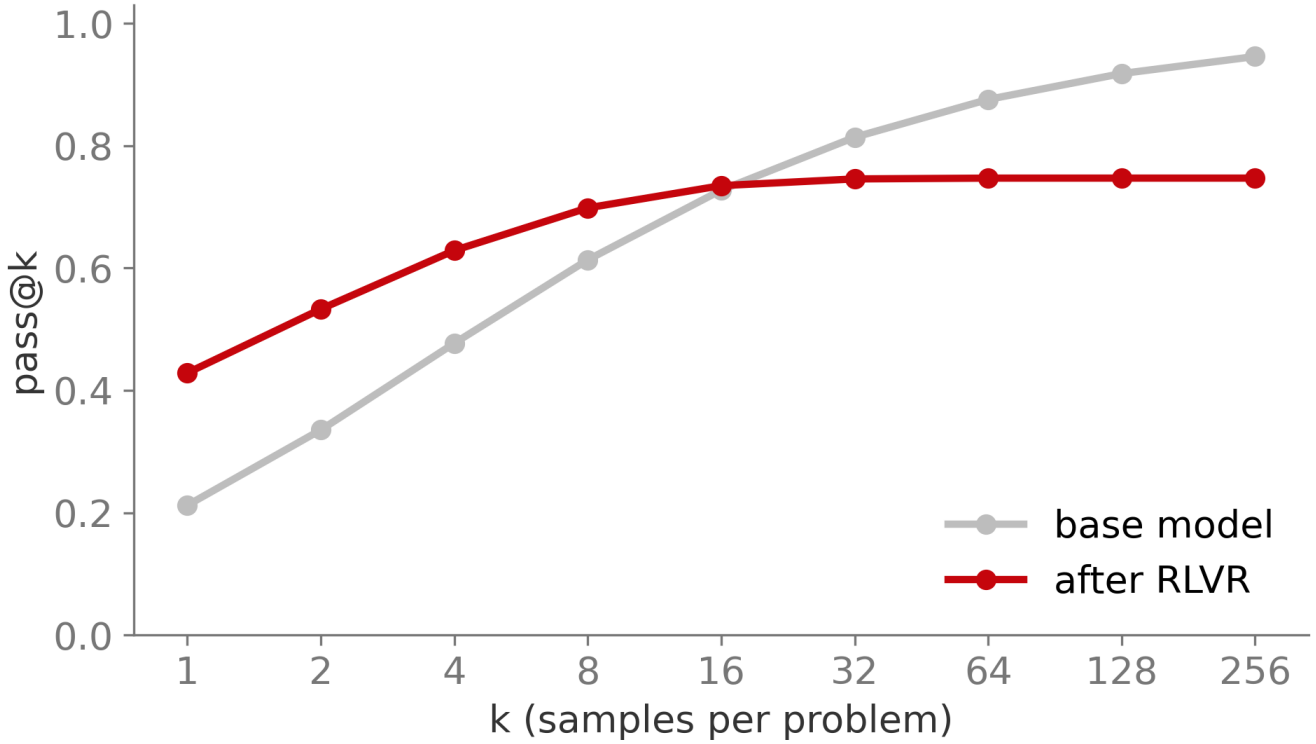
Reasoning models increase **test-time compute** by generating intermediate reasoning steps.



https://arxiv.org/pdf/2501.11223

Open Questions

New reasoning, or sharper sampling?



- Small k: the RL model wins (it picks good paths more often)
- Large k: the **base model wins** (it covers more problems)
- RL mostly **sharpens** what the base model could already sample
- Debated: longer RL runs may change this

Schematic after Yue et al., 2025

Is the chain of thought the real reason?

- Experiment: in the few-shot examples, make the right answer **always (A)**
- Models drift toward (A), and their explanations **justify the new answer without mentioning the pattern**
- Accuracy fell by up to **36%** across 13 reasoning tasks
- A chain of thought is text the model generates, **not a trace of its computation**
- Matters when we read reasoning to check or monitor a model

Turpin et al., NeurIPS 2023

Should we read the chain of thought?

If a model's written reasoning isn't a faithful trace of its computation, what is it good for? Should users see it?

- o1 shows users a summary; DeepSeek-R1 shows the raw reasoning. Which would you choose, and why?
- If we penalize "bad thoughts" during training, what might the model learn instead?
- Is this a fixed law of nature, or a architectural artifact?

Limits and failure modes

- **The checker can be gamed:** code that special-cases the visible tests passes them
- **Cost:** thousands of reasoning tokens per answer; overthinking easy questions
- **Coverage:** only tasks with checkable answers; essays and advice still need RLHF
- **Difficulty matters:** problems that are always or never solved give no gradient
- **Faithfulness:** the written reasoning may not be why the model answered

Ben Lengerich © University of Wisconsin-Madison 2026

Summary: the post-training stack, updated

Stage	Signal	What it adds
Pretraining	Next token of raw text	A model of language
SFT / distillation	Demonstrations (human or model)	Follows instructions; copies reasoning style
RLHF / DPO	Human preferences	Helpful, harmless, preferred style
RLVR	A program that checks the answer	Longer, more reliable reasoning
Test-time compute	More samples, more thinking tokens	Accuracy without new weights

Ben Lengerich © University of Wisconsin-Madison 2026

Next Time: Parameter-Efficient Fine-Tuning

LoRA, Adapters, and Prompting

Fine-tune a billion-parameter model by training a tiny fraction of it

Ben Lengerich © University of Wisconsin-Madison 2026

Questions?

