



SOME INSIGHTS INTO LLM REASONING

JULIA KEMPE

NYU CDS AND COURANT INSTITUTE

META FUNDAMENTAL AI RESEARCH (FAIR) PARIS

APRIL 26, 2026

SCIENCE FOR DEEP LEARNING WORKSHOP, ICLR 2026

COLLABORATORS AND REFERENCES

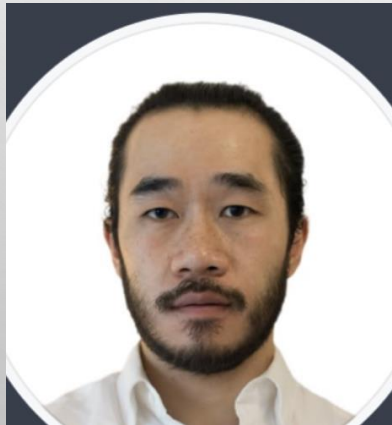
How Reinforcement Learning After Next-Token Prediction Facilitates Learning, ICLR'26

Nikolaos Tsilivis, Eran Malach, Karen Ullrich, JK



Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability, RSI workshop at ICLR'26

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, JK

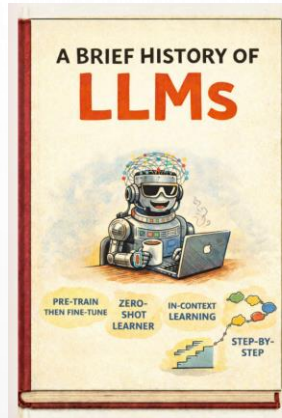


A BRIEF (PRE) HISTORY OF LLMS

Pre-2020: “Pre-train then fine-tune” (GPT-1, GPT-2)

Improving Language Understanding by Generative Pre-Training (Radford et al., 2018)

GPT-2 hinted at “zero-shot” abilities: model could perform tasks it wasn’t specifically trained on



2020: In-Context Learning (ICL)

Language Models are Few-Shot Learners (Brown et al., 2020)

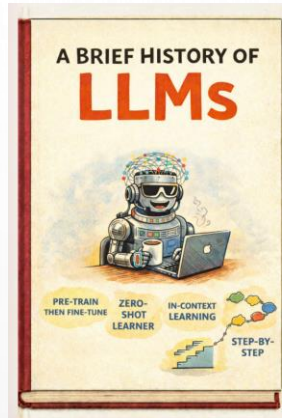
At 175B parameters GPT-3 could learn patterns/take instructions from a prompt

2022: Chain-of-Thought (CoT) “Let’s think step-by-step”

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Wei et al., 2022)

For math/logic ICL wasn’t enough. Prompting the model to "show its work" or "think step-by-step" skyrocketed performance on reasoning tasks

A BRIEF HISTORY OF LLMS FOR REASONING



Pre-2020: “Pre-train then fine-tune” (GPT-1, GPT-2)

Improving Language Understanding by Generative Pre-Training (Radford et al., 2018)

GPT-2 hinted at “zero-shot” abilities: model could perform tasks it wasn’t specifically trained on

2020: In-Context Learning (ICL)

Language Models are Few-Shot Learners (Brown et al., 2020)

At 175B parameters GPT-3 could learn patterns/take instructions from a prompt

2022: Chain-of-Thought (CoT) “Let’s think step-by-step”

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Wei et al., 2022)

For math/logic ICL wasn’t enough. Prompting the model to “show its work” or “think step-by-step” skyrocketed performance on reasoning tasks

2024-25: Reinforcement learning from verifiable rewards (RLVF)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (DeepSeek-AI, 2024/2025 / OpenAI o1 /Kimi)

Use RL & verifiable outcomes (math/code) to force the model to develop internal reasoning and self-correction

A BRIEF HISTORY OF LLMS FOR REASONING

2024-25: Reinforcement learning from verifiable rewards (RLVF)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (DeepSeek-A

2024/2025 / OpenAI o1 /Kimi)

Use RL & verifiable outcomes (math/code) to force the model to develop internal reasoning and self-correction

End 25 - 2026: Agents.

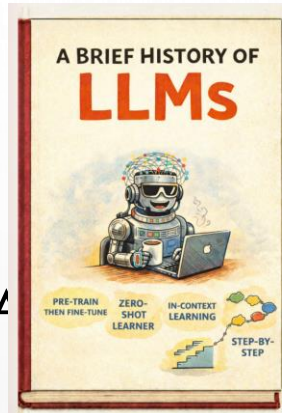
Training-free (in context) RL.

“In skills”-Learning.

“In harness”-RL.

Meta-harness learning.

...



REINFORCEMENT LEARNING FOR LLM POST-TRAINING

Fantastically successful! The O1 (Sep 2024) and DeepSeek/Kimi (Jan 2025) “Revolution”

Introducing OpenAI o1

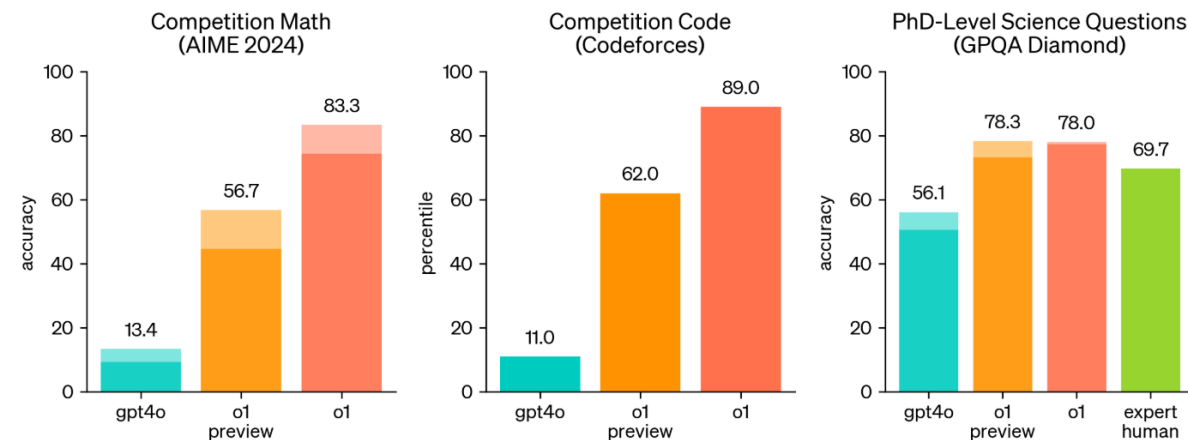
A new series of AI models designed to spend more time thinking before they respond.

Open Source / open method (GRPO)



DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI



o1 greatly improves over GPT-4o on challenging reasoning benchmarks. Solid bars show pass@1 accuracy and the shaded region shows the performance of majority vote (consensus) with 64 samples.

KIMI K1.5: SCALING REINFORCEMENT LEARNING WITH LLMs

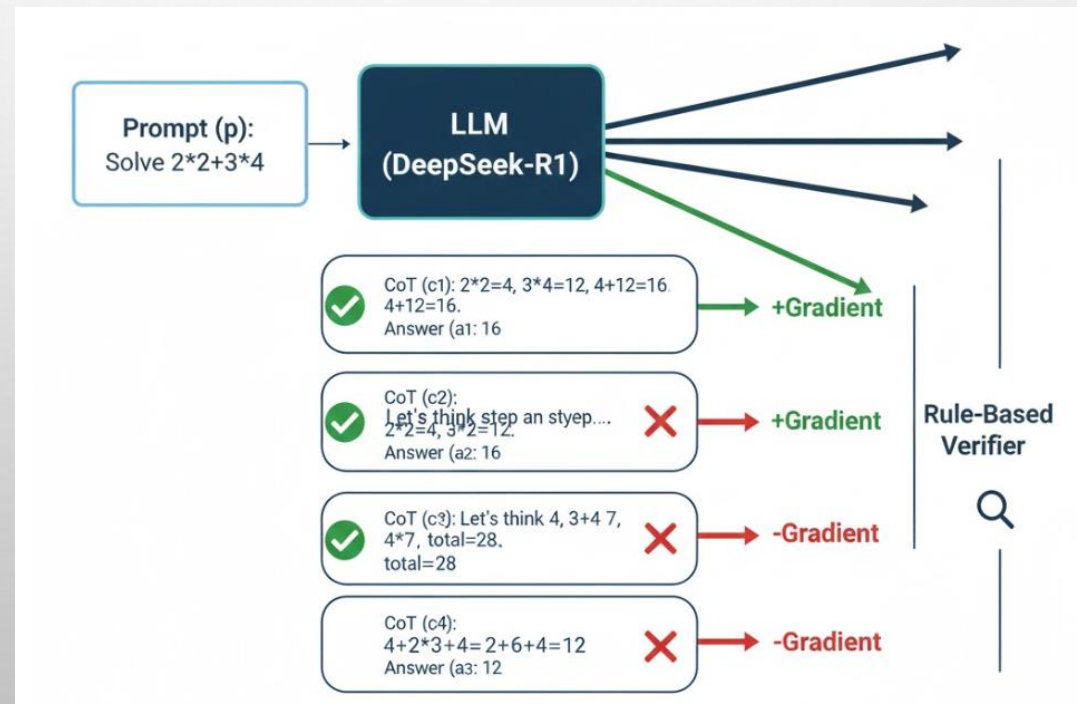
TECHNICAL REPORT OF KIMI K1.5

Kimi Team

REINFORCEMENT LEARNING FOR LLM POST-TRAINING

Group Relative Policy Optimization (GRPO) & “Test-time compute paradigm”

- The model generates a **group** of multiple CoT outputs (say, 16 different attempts) for the same prompt and compares them against each other. The outputs that are better than the group average are rewarded; the ones below average are penalized.
- For reasoning tasks the reward is rule-based (“verifiable”), no “critic” model or “human-in-the-loop” needed.



GRPO FOR RLVF: OBSERVATIONS

- LLMs use lots of test time tokens to reason and improve the performance.
- From train-time scaling to test-time scaling

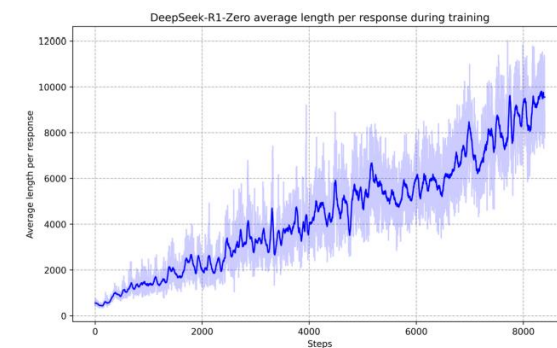
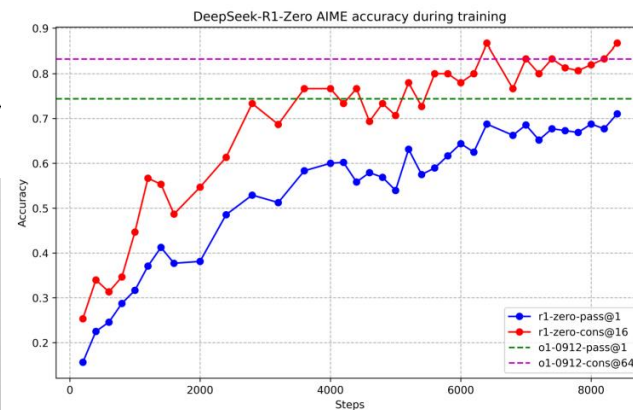
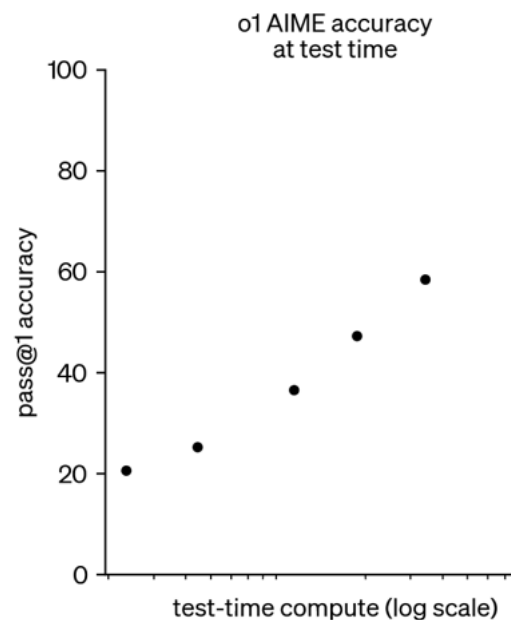
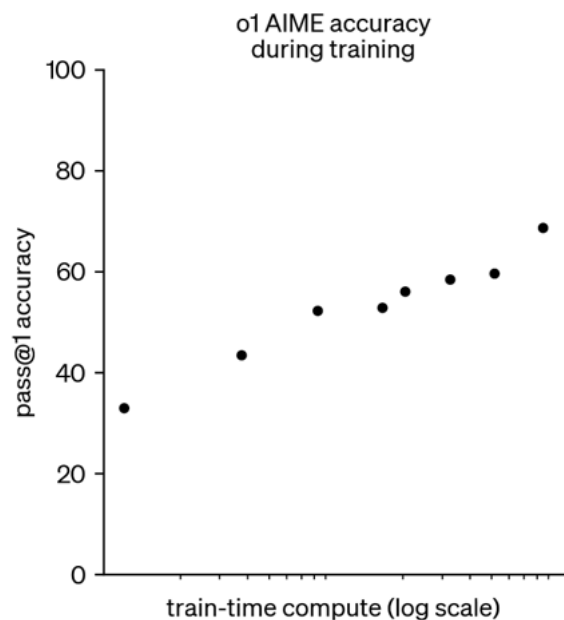


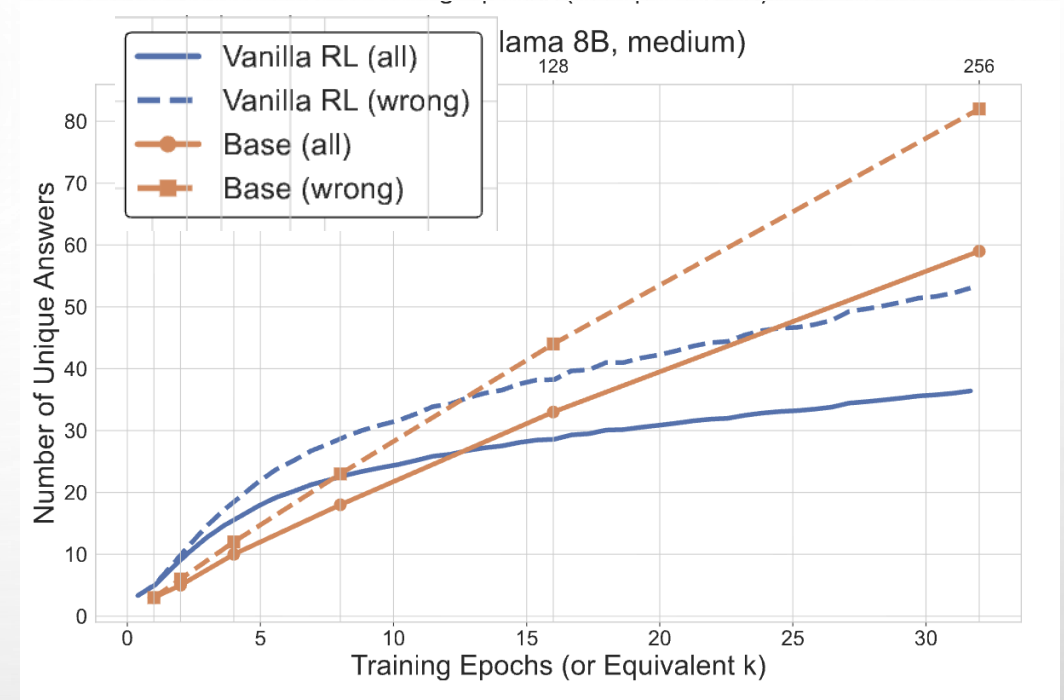
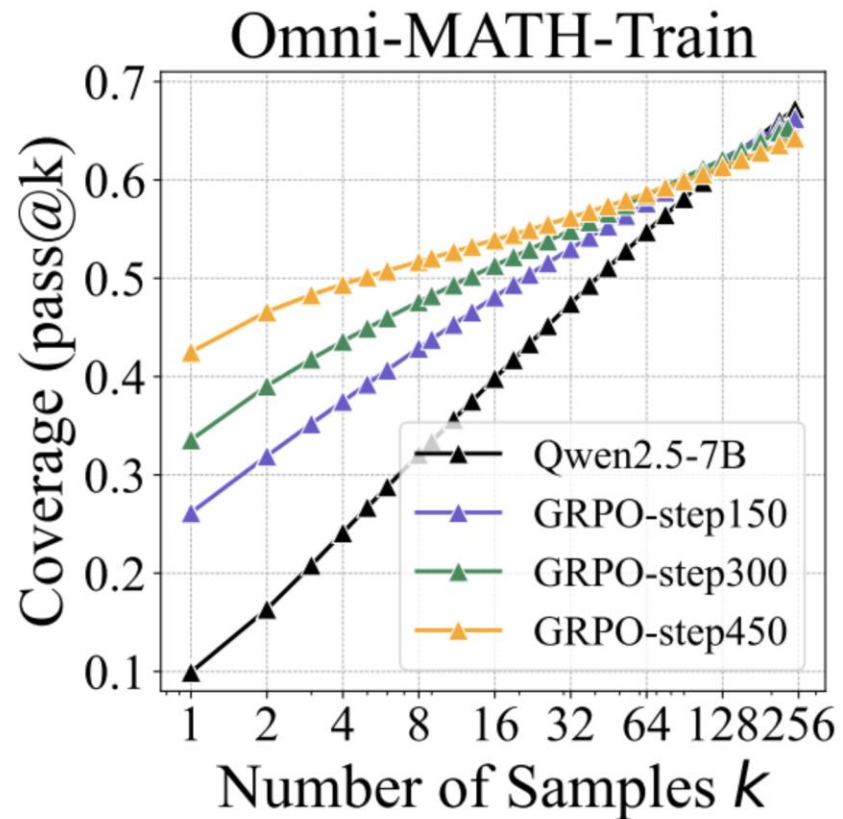
Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

LEARNING OR RETRIEVING?

“The intelligence already exists in the pre-trained model, it is all about extracting it.”

Sebastian Bubeck (November 2024)

LEARNING OR “SHARPENING”?



Outcome-based Exploration for LLM Reasoning

Yuda Song^{1,2}, Julia Kempe^{1,3}, Remi Munos¹

¹FAIR at Meta, ²CMU, ³NYU

A CONTROLLED STUDY

Echo Chamber: RL Post-training Amplifies Behaviors Learned in Pretraining

Rosie Zhao*

Harvard University
Kempner Institute

Alexandru Meterez*

Harvard University
Kempner Institute

Sham Kakade

Harvard University
Kempner Institute

Cengiz Pehlevan

Harvard University
Kempner Institute

Samy Jelassi[†]

Harvard University

Eran Malach[†]

Harvard University
Kempner Institute

Abstract

Reinforcement learning (RL)-based fine-tuning has become a crucial step in post-training language models for advanced mathematical reasoning and coding. Following the success of frontier reasoning models, recent work has demonstrated that RL fine-tuning consistently improves performance, even in smaller-scale models; however, the underlying mechanisms driving these improvements are not well-understood. Understanding the effects of RL fine-tuning requires disentangling its interaction with pretraining data composition, hyperparameters, and model scale, but such problems are exacerbated by the lack of transparency regarding the training data used in many existing models. In this work, we present a systematic end-to-end study of RL fine-tuning for mathematical reasoning by training models entirely from scratch on different mixtures of fully open datasets. We investigate the effects of various RL fine-tuning algorithms (PPO, GRPO, and Expert Iteration) across models of different scales. Our study reveals that RL algorithms consistently converge towards a dominant output distribution, amplifying patterns in the pretraining data. We also find that models of different scales trained on the same data mixture will converge to distinct output distributions, suggesting that there are scale-dependent biases in model generalization. Moreover, we find that RL post-training on simpler questions can lead to performance gains on harder ones, indicating that certain reasoning capabilities generalize across tasks. Our findings show that small-scale proxies in controlled settings can elicit interesting insights regarding the role of RL in shaping language model behavior.¹

Abstract

Reinforcement learning (RL)-based fine-tuning has become a crucial step in post-training language models for advanced mathematical reasoning and coding. Following the success of frontier reasoning models, recent work has demonstrated that RL fine-tuning consistently improves performance, even in smaller-scale models; however, the underlying mechanisms driving these improvements are not well-understood. Understanding the effects of RL fine-tuning requires disentangling its interaction with pretraining data composition, hyperparameters, and model scale, but such problems are exacerbated by the lack of transparency regarding the training data used in many existing models. In this work, we present a systematic end-to-end study of RL fine-tuning for mathematical reasoning by training models entirely from scratch on different mixtures of fully open datasets. We investigate the effects of various RL fine-tuning algorithms (PPO, GRPO, and Expert Iteration) across models of different scales. Our study reveals that RL algorithms consistently converge towards a dominant output distribution, amplifying patterns in the pretraining data. We also find that models of different scales trained on the same data mixture will converge to distinct output distributions, suggesting that there are scale-dependent biases in model generalization. Moreover, we find that RL post-training on simpler questions can lead to performance gains on harder ones, indicating that certain reasoning capabilities generalize across tasks. Our findings show that small-scale proxies in controlled settings can elicit interesting insights regarding the role of RL in shaping language model behavior.¹

AMPLIFY PATTERNS IN THE TRAINING DATA...

SETUP:

We assume that **generalizing “long” patterns are present in the pre-training data** (but they are perhaps very rare). This allows us to explain:

1. Why it is difficult for a model to generalize during pre-training for some tasks.
2. How reinforcement learning leads to a rapid improvement in terms of samples.
3. What optimization pressures cause increase in the length of the response.

How Reinforcement Learning After Next-Token Prediction Facilitates Learning

Nikolaos Tsilivis^{1,*}, Eran Malach^{2,‡}, Karen Ullrich^{3,†}, Julia Kempe^{1,3,†}

¹New York University, ²Harvard University, ³FAIR, Meta

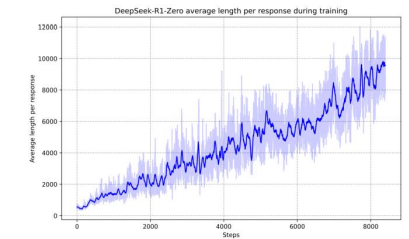
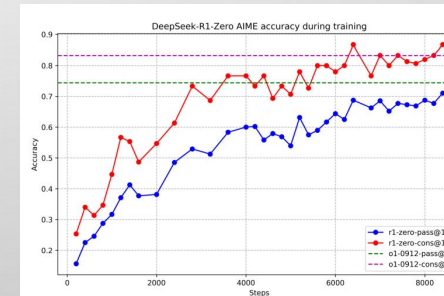
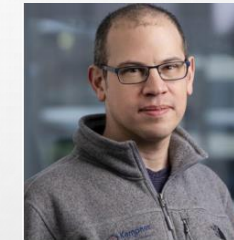
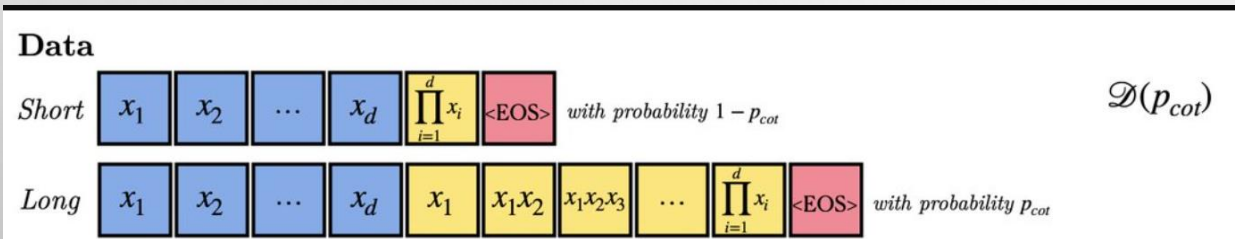


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

ISOLATING THE MECHANISM: CONTROLLED EXPERIMENTS WITH PARITY

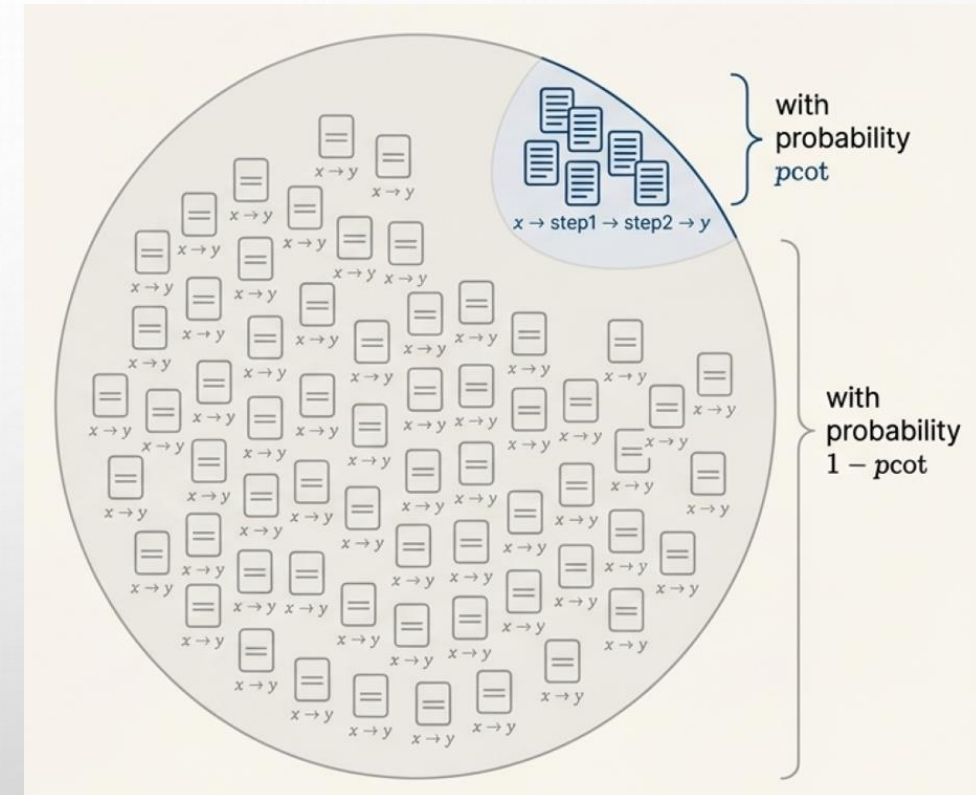
SETUP:

- Parity of d bits (XOR): classic hard learning problem for neural nets.
- Train on controlled mixture distribution $\mathcal{D}(p_{cot})$:
 - **Long demonstrations (rare)**: input bits + full chain-of-thought with intermediate parities, with probability p_{cot}
 - **Short demonstrations (common)**: input bits + only final answer, with probability $1 - p_{cot}$



HYPOTHESIS:

Mimics the internet where step-by-step reasoning examples are rare amongst a sea of direct answers.



PRETRAINING FAILS WHEN GOOD EXAMPLES ARE RARE

Setup: GPT2, depth=4, #heads=32, embedding-dim=128

Next-token prediction (pre-training)

Training data (d=5)

-1, +1, +1, -1, +1, +1, <EOS>
-1, -1, -1, +1, +1, -1, <EOS>
-1, +1, -1, +1, +1, -1, -1, +1, +1, +1, <EOS>
+1, -1, -1, +1, +1, +1, <EOS>
+1, +1, +1, -1, +1, -1, <EOS>
+1, +1, +1, +1, +1, +1, <EOS>



Transformer

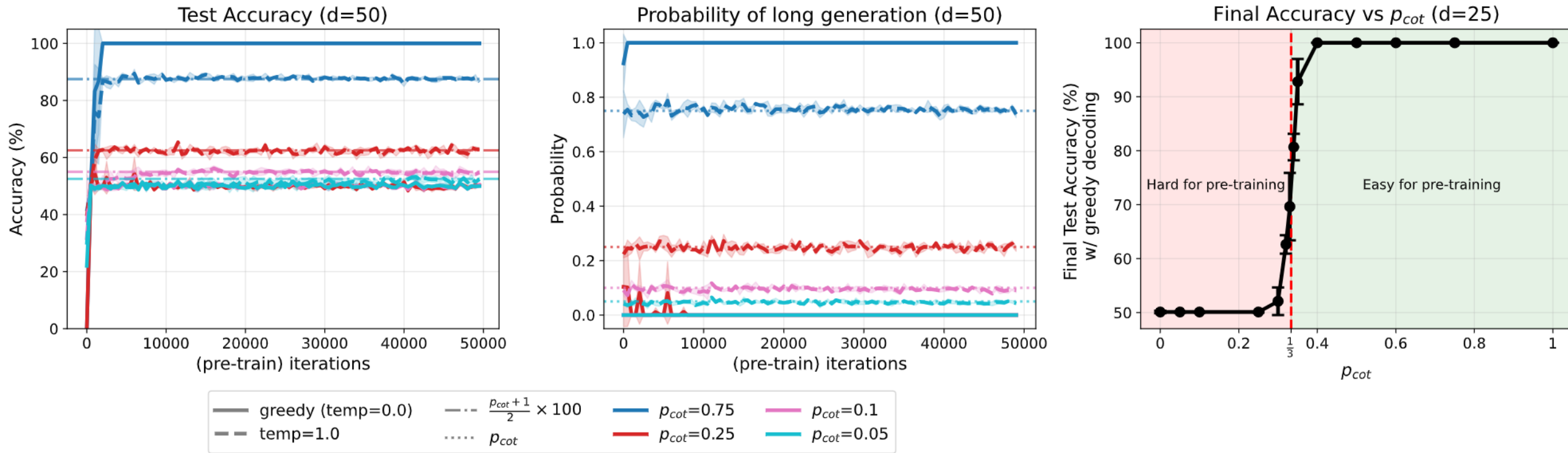


Inference

+1, +1, -1, -1, +1, -1, <EOS> ✗
-1, -1, -1, +1, -1, -1, <EOS> ✗
-1, +1, -1, -1, +1, -1, -1, +1, -1, -1, <EOS>
-1, -1, -1, +1, +1, -1, <EOS>
+1, -1, +1, -1, +1, -1, <EOS> ✗
-1, -1, +1, +1, +1, +1, <EOS>

PRETRAINING WITH NTP

Setup: GPT2, depth=4, #heads=32, embedding-dim=128



Generated distribution after NTP remains **length-calibrated!**

Same fraction of long and short.

If p_{cot} is small: model generates short, random guesses

PRETRAINING, THEN POST-TRAINING

Setup: GPT2, depth=4, #heads=32, embedding-dim=128

Next-token prediction (pre-training)

Training data (d=5)

-1, +1, +1, -1, +1, +1, <EOS>
-1, -1, -1, +1, +1, -1, <EOS>
-1, +1, -1, +1, +1, -1, -1, +1, +1, +1, <EOS>
+1, -1, -1, +1, +1, +1, <EOS>
+1, +1, +1, -1, +1, -1, <EOS>
+1, +1, +1, +1, +1, +1, <EOS>

Transformer

Inference

+1, +1, -1, -1, +1, -1, <EOS> ✗
-1, -1, -1, +1, -1, -1, <EOS> ✗
-1, +1, -1, -1, +1, -1, -1, +1, -1, -1, <EOS>
-1, -1, -1, +1, +1, -1, <EOS>
+1, -1, +1, -1, +1, -1, <EOS> ✗
-1, -1, +1, +1, +1, +1, <EOS>

Reinforcement learning (post-training)

Transformer

Inference

-1, +1, -1, -1, -1, +1, <EOS>
+1, +1, -1, -1, -1, +1, +1, -1, +1, -1, <EOS>
+1, -1, +1, +1, -1, +1, -1, -1, -1, +1, <EOS>
-1, +1, -1, +1, +1, -1, <EOS> ✗
+1, -1, +1, -1, +1, -1, <EOS> ✗
+1, -1, -1, -1, +1, -1, <EOS>

Transformer

Inference

-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, <EOS>
-1, +1, -1, +1, -1, -1, -1, +1, +1, -1, <EOS>
+1, +1, -1, -1, -1, +1, +1, -1, +1, +1, <EOS>
+1, -1, -1, +1, -1, +1, -1, +1, +1, -1, <EOS>
-1, +1, +1, +1, +1, -1, -1, -1, -1, -1, <EOS>
+1, +1, -1, +1, +1, +1, +1, -1, -1, -1, <EOS>

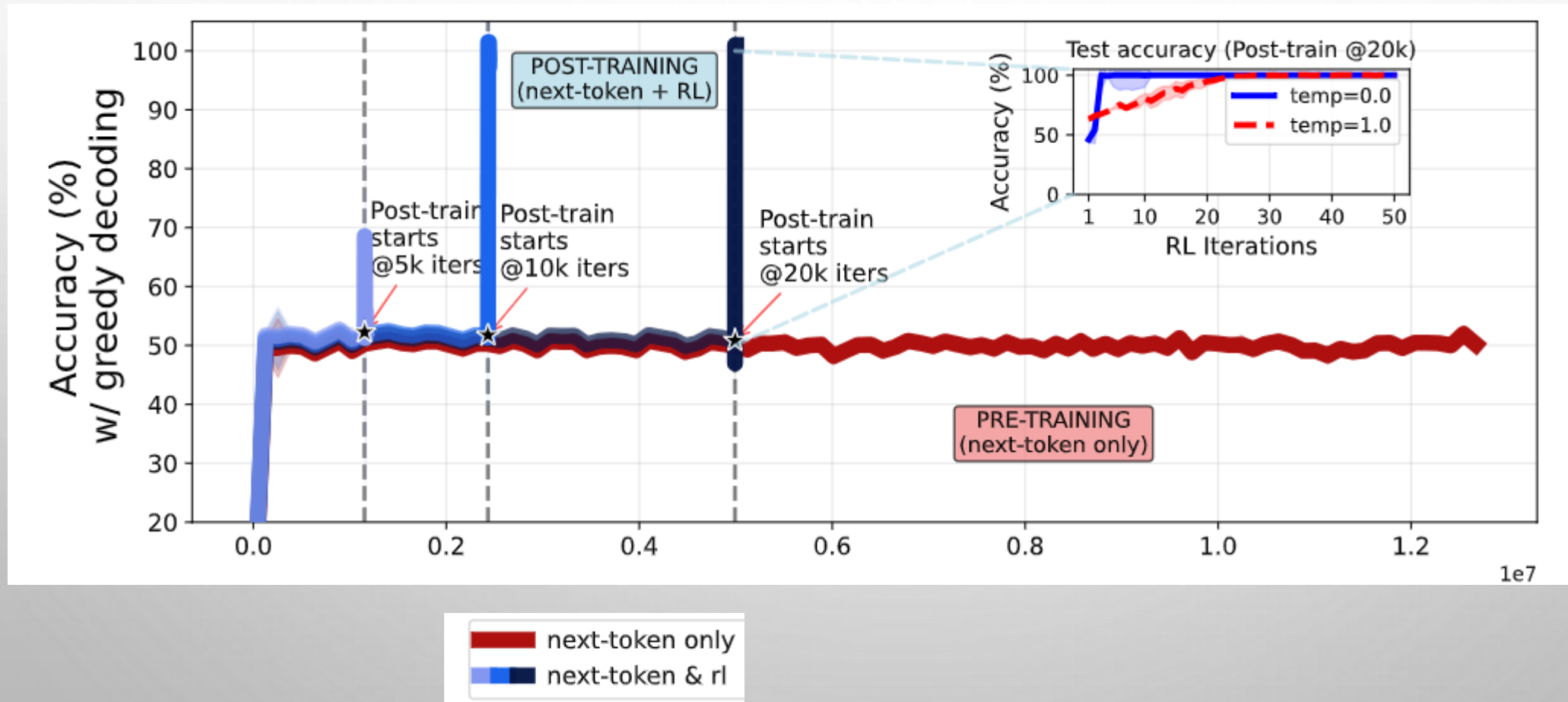
| reward |

RL iterations

| reward |

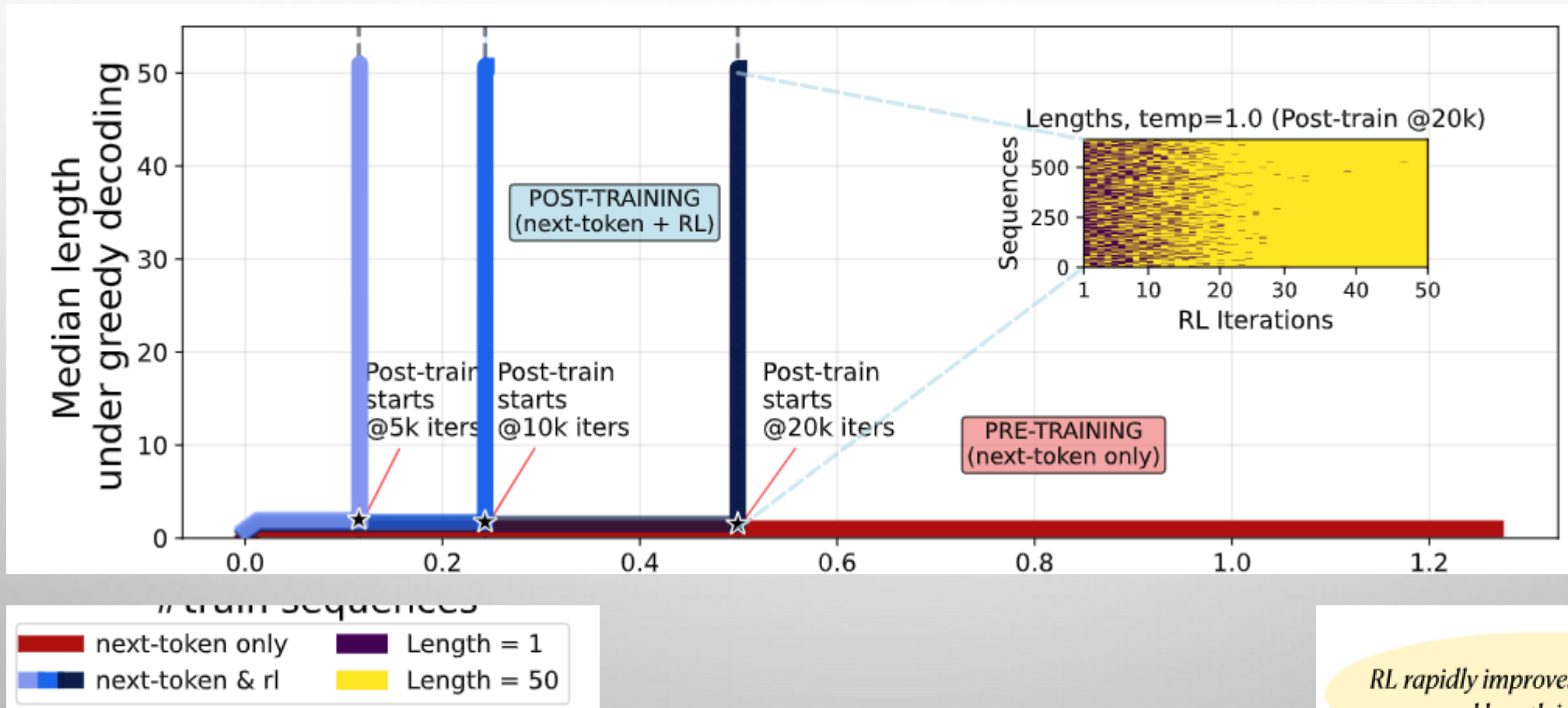
RAPID LIFT-OFF IN PERFORMANCE!

For small p_{CoT} switch from NTP to RL after some (5K, 10K, 20K) steps:



AND LENGTH INCREASES!

For small p_{CoT} switch from NTP to RL after some (5K, 10K, 20K) steps:



RL rapidly improves performance and length increases

MECHANISM (PROOF FOR SIMPLIFIED BUT EXPRESSIVE AUTOREGRESSIVE LINEAR MODEL)

DURING PRETRAINING (NTP):

- Model learns to **quickly generate long-CoT responses** (answer easy to compute)
- **Fails to learn from short responses** (parity is hard) and resorts to random guesses
- **Length calibrated:** model generates long response w.p. p_{CoT} and short w.p. $1 - p_{CoT}$

It learns successfully from long rather than short sequences but **doesn't change relative weight**.

DURING POST-TRAINING (RLVF / REST/ STAR):

- RL rewards correctness: reward-weighted loss **effectively up-samples the long-CoT data**
- This **shifts the model policy**, causing it to **generate long responses more frequently**, which in turn leads to generalization
- Generalization happens after a number of **rounds logarithmic in $1 / p_{CoT}$** !

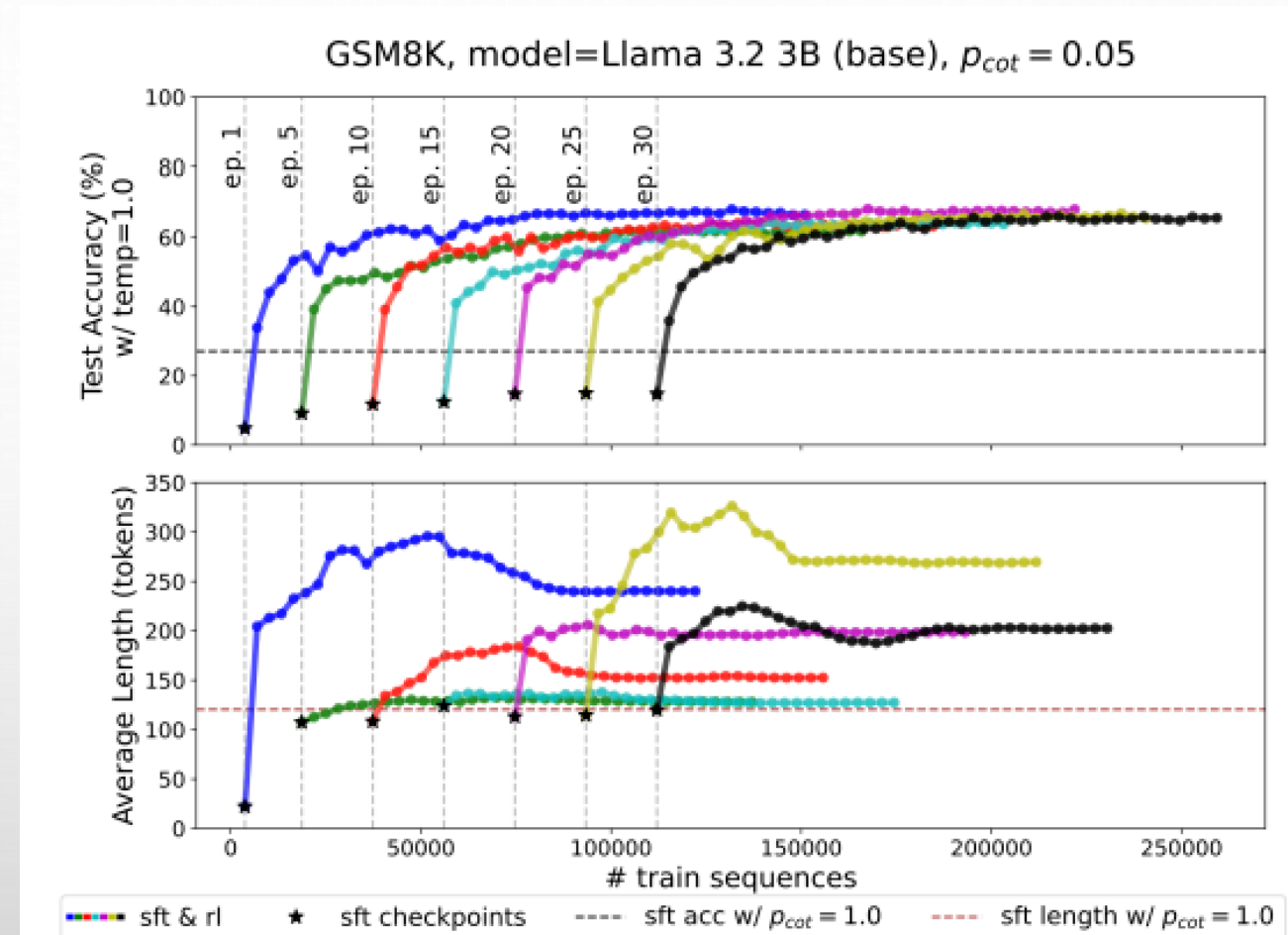
RL is a highly effective tool for “mining” and capitalizing on rare, high-quality subset of data. Length increase is a learning phenomenon (not just a “scratchpad”).

WITH PRETRAINED LLMS: AMPLIFICATION MECHANISM IN THE WILD

LLAMA3B:

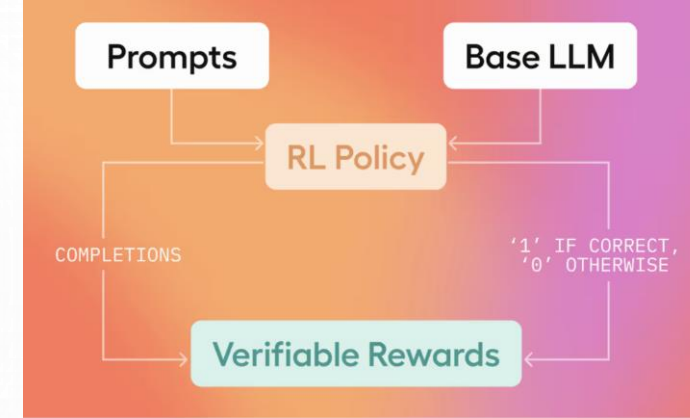
- SFT (next token prediction) on a mixture from GSM8K
- p_{CoT} fraction with chain-of-thought, $1 - p_{CoT}$ fraction with answer only, $p_{CoT}=0.05$

We observe a similar phenomenon!



Our framework provides a principled explanation for the success (and sample efficiency) of the post-training paradigm in state-of-the-art models.

RLVR'S "INVISIBLE LEASH"*:



“CRITICAL CONSENSUS”:

- **THE CURRENT RLVR RECIPE PRIMARILY PRESERVES RATHER THAN EXPANDS THE BASE MODEL'S SOLUTION COVERAGE.**
 - Learning process is largely confined to re-weighting probabilities within the base model's pre-existing trajectories
 - Cannot discover fundamentally new reasoning trajectories outside this initial capability.
 - Diversity of answers declines.
- **HARD PROBLEMS CANNOT BE LEARNED**
 - What happens when a model gets stuck on problems it fails 100% of the time? There is no learning signal to begin. “Reasoning plateau”.

*Wu et al.'25, “THE INVISIBLE LEASH? WHY RLVR MAY OR MAY NOT ESCAPE ITS ORIGIN”

Image credit: <https://labelstud.io/blog/reinforcement-learning-from-verifiable-rewards/>

META-RL: HOW TO SOLVE HARD PROBLEMS

Edge of Learnability:

- RL training often **plateaus**, “hard” problems in the training set remain unsolved
- Low initial success rate (e.g. fail@128) means **no training signal** and no learning
- Recall: RLVF learns in a sparse reward world , “wanders indefinitely” without positive reward

How to escape the “All-Zero” region?

META-RL: HOW TO SOLVE HARD PROBLEMS

Edge of Learnability:

- RL training often **plateaus**, “hard” problems in the training set remain unsolved
- Low initial success rate (e.g. fail@128) means **no training signal** and no learning
- Recall: RLVF learns in a sparse reward world , “wanders indefinitely” without positive reward

How to escape the “All-Zero” region?

- **RL training on simpler questions can kickstart learning on hard datasets**
- **But how to design a “stepping stone curriculum” ?**
 - Curated additional data (expert-driven process, doesn’t scale)
 - Filtering of existing data (but what criteria? How do we know what easier questions transfer learning to hard ones? Also doesn’t scale.)

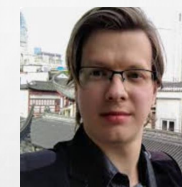
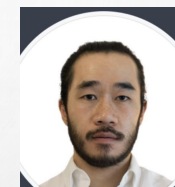
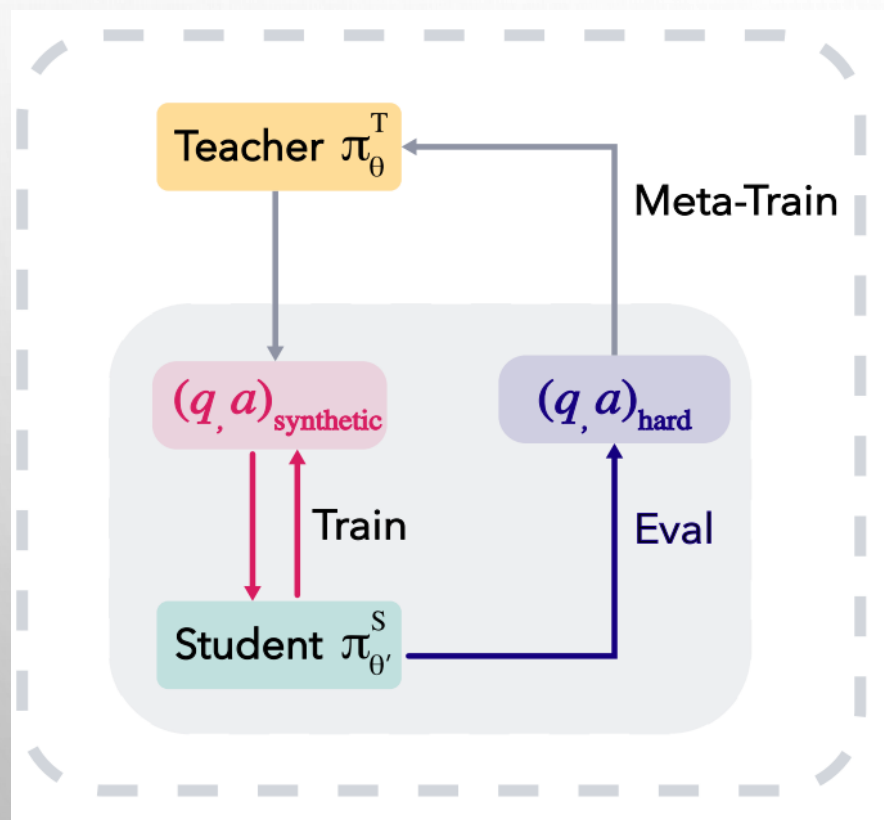
Pretrained LLMs themselves are an untapped resource (teacher) for learnable questions to unlock learning on hard datasets. RL on the teacher can “mine” pretraining corpus.

CAN A MODEL TEACH ITSELF TO SOLVE PROBLEMS THAT ARE, INITIALLY, COMPLETELY OUTSIDE ITS CAPABILITIES?

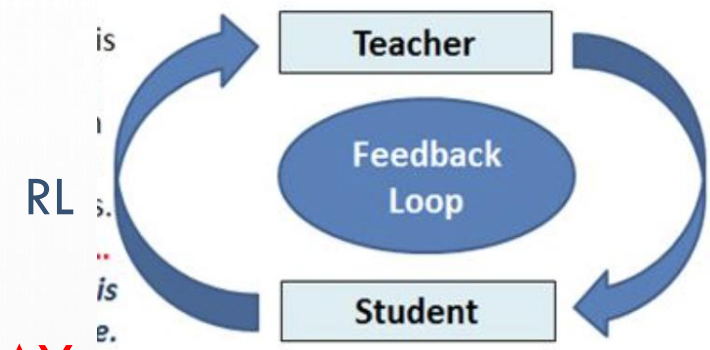
Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability,

ICLR'26 RSI Workshop

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, JK

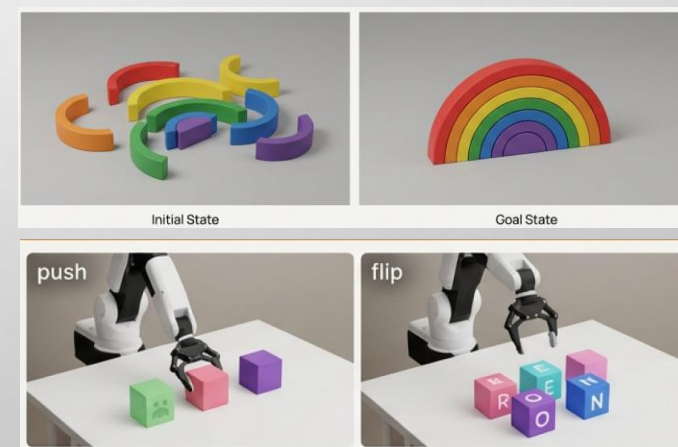
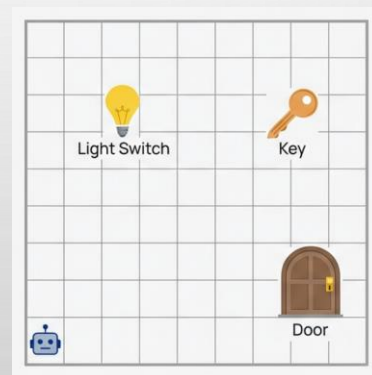
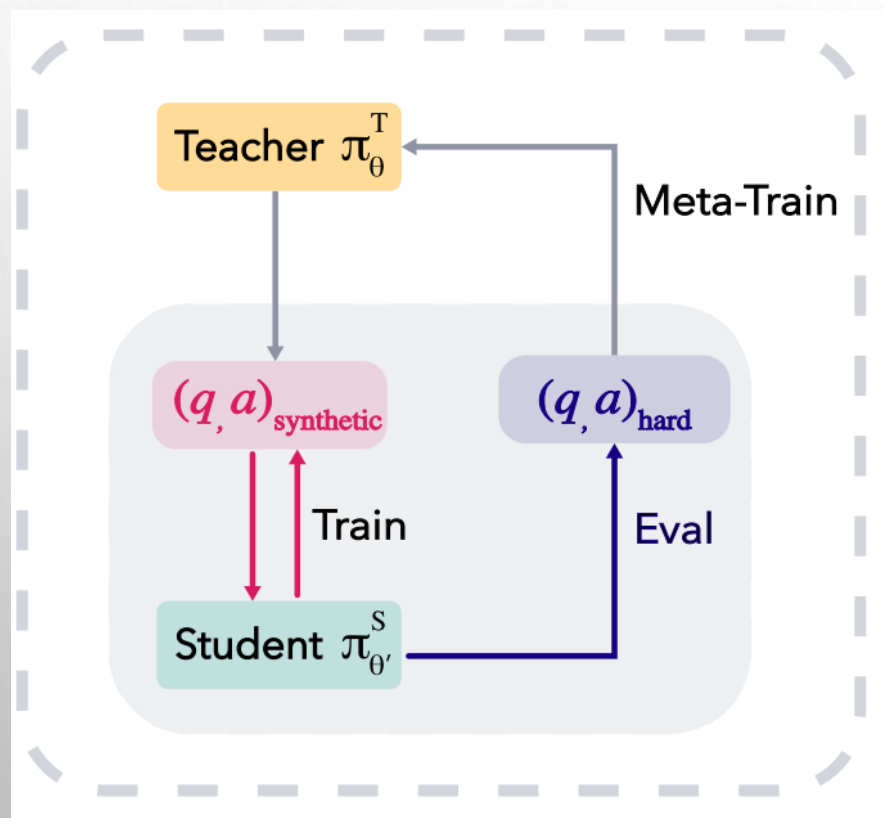


META-RL: HOW TO TEACH MODELS TO TEACH THEMSELVES



ASYMMETRIC SELF-PLAY:

Has proven to be a powerful engine for **skill-discovery** and **exploration** in physical domains (mazes, robotics).



Sukhbaatar et al.'17: Intrinsic motivation and automatic curricula via asymmetric self-play

OpenAI'21: "Asymmetric Self-Play for Automatic Goal Discovery..."

GROUNDING SELF-PLAY IN MEASURABLE PROGRESS

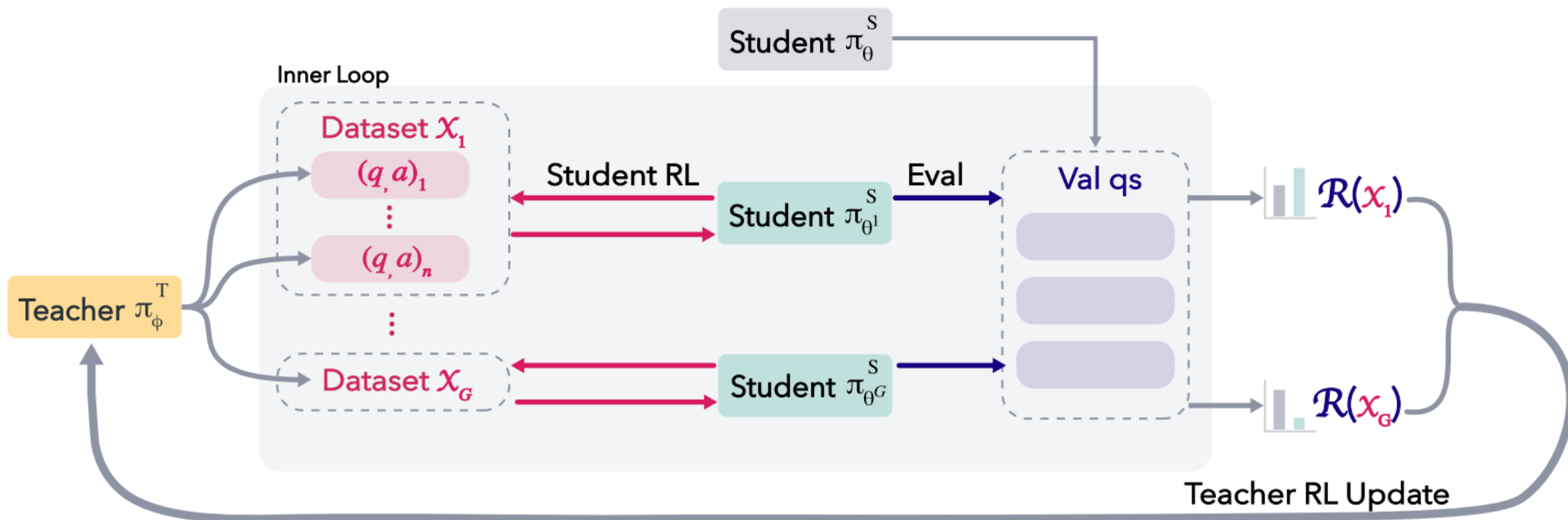
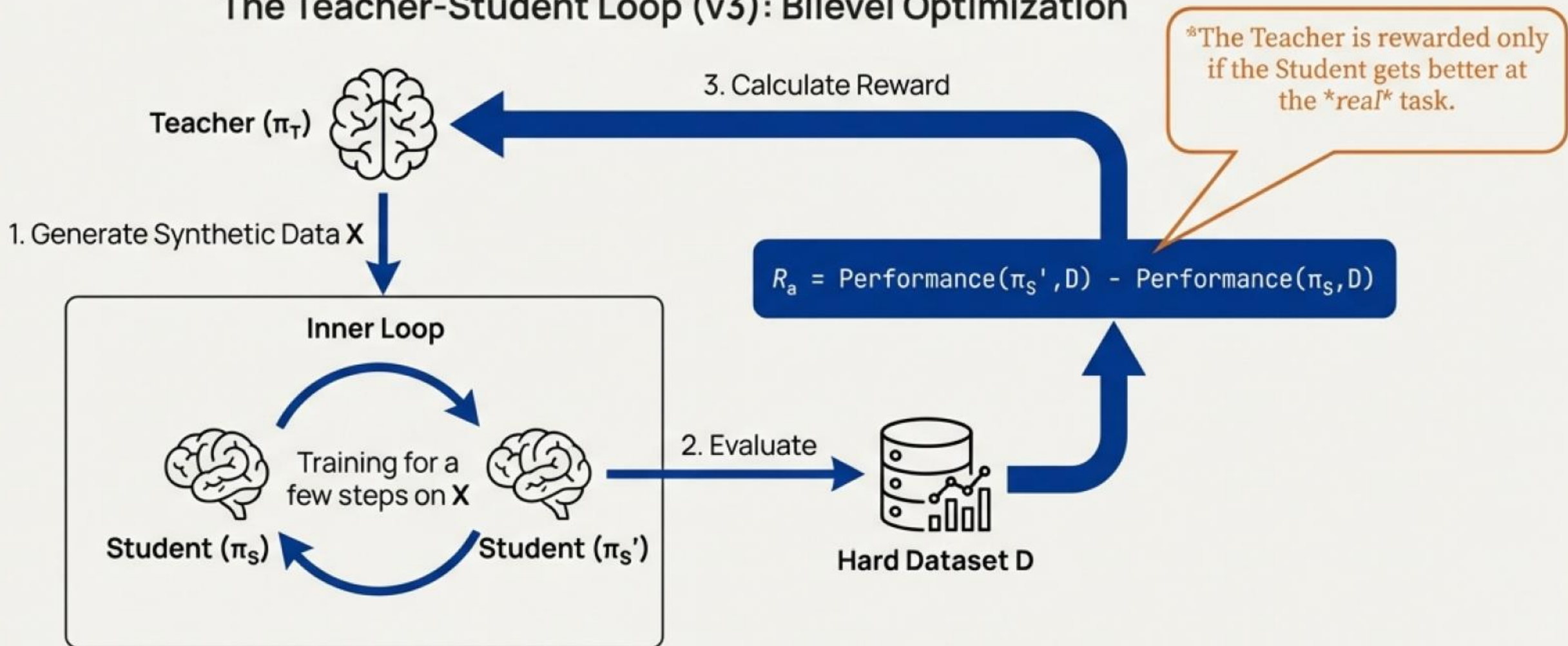


Figure 2 The soar Meta-RL Loop. The teacher and student are initialized from the same model. In the **outer RL loop** the teacher generates candidate question-answer pairs that are partitioned into datasets. In the **inner RL loop**, the student is trained for 10 steps on the candidate problems and evaluated on sampled hard problems. The teacher is rewarded based on the resulting student improvement over the student baseline, grounding the synthetic curriculum in real learning progress.

SOAR (SELF-OPTIMIZATION VIA ASYMMETRIC RL): GROUNDING SELF-PLAY IN MEASURABLE PROGRESS

The Teacher-Student Loop (v3): Bilevel Optimization



TEACHING MODELS TO TEACH THEMSELVES: TARGETED EXPERIMENTS

Model: LLAMA-3.2-3B-INSTRUCT

Create Hard Dataset: Filter from MATH or HARP
for failure to pass @128

Setup: inner student loop – 10 steps of GRPO
outer teacher loop – sample 4 groups of
64 (q,a) pairs (GRPO with group size 4),
reward is student improvement over
untrained student

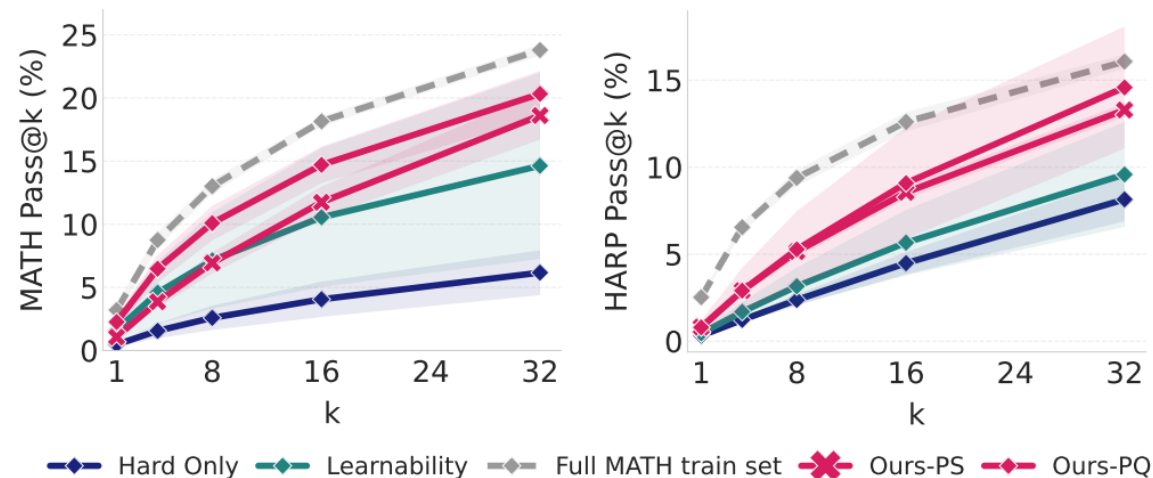


Figure 3 Performance on MATH/HARP fail@128 subsets. Synthetic curricula generated with SOAR (PQ) and inference with the promoted student (PS) improve performance on hard problems where the base model plateaus, outperforming direct training on the hard problems and teacher-training with intrinsic learnability rewards.

The ability to generate a useful curriculum does not require the pre-existing ability to solve the target problem!

TEACHING MODELS TO TEACH THEMSELVES: TARGETED EXPERIMENTS

Model: LLAMA-3.2-3B-INSTRUCT

Create Hard Dataset: Filter from MATH or HARP
for failure to pass @128

Setup: inner student loop – 10 steps of GRPO
outer teacher loop – sample 4 groups of
64 (q,a) pairs (GRPO with group size 4),
reward is student improvement over
untrained student

The learned stepping-stone curriculum
transfers **out of domain**.

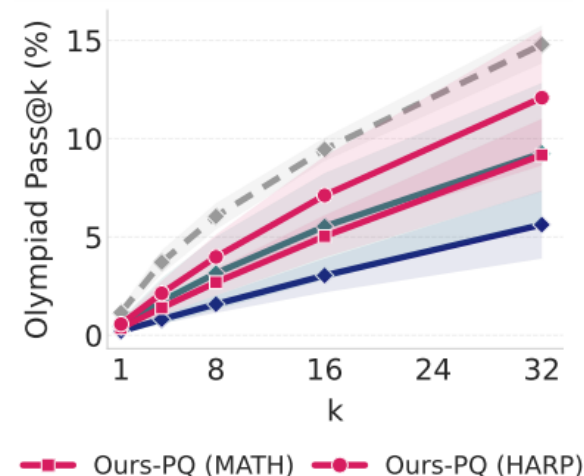


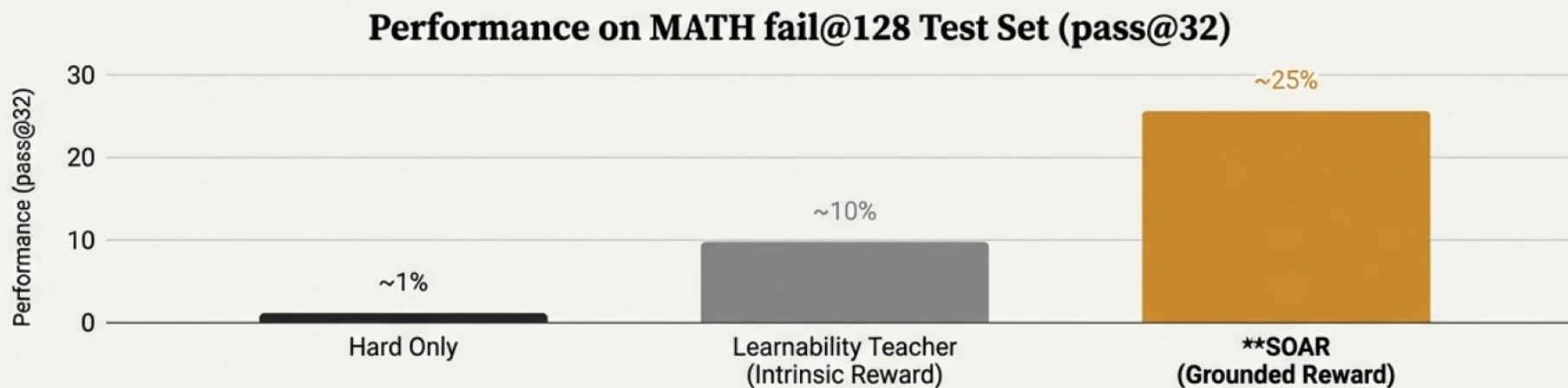
Figure 4 Transfer performance to OlympiadBench fail@128 subset. Questions optimized for MATH and HARP transfer to improving student performance on an external dataset.

The ability to generate a useful curriculum does not require the pre-existing ability to solve the target problem!

SELF-PLAY: INTRINSIC VERSUS GROUNDED TEACHER REWARDS

- Contemporary “zero-data” LLM self-play studies (R-Zero, AbsoluteZero, Language-Self-Play (LSP), Self-questioning LLMs (SQLM), Self-Play RL (SeRL) ...) use intrinsic teacher rewards (learnability, uncertainty,...).
- Lead to “Reward Hacking”: Curriculum can “drift”, generating interesting but irrelevant tasks, stalling student progress or collapsing it completely

Grounding the reward in real progress is critical to generating a useful curriculum.



WHAT MAKES A GOOD “STEPPING-STONE” PROBLEM?

Stage 0

Q: Compute the value of x for the equation:

$$f(x) = (x^2 + 1) / (x - 1)?$$

A: 2

Stage 1

Q: What is the probability of randomly picking a blue or green item from a bag containing 8 red, 5 blue, and 5 green items?

A: 5/9

Stage 2

Q: Given the polynomial $f(x) = x^3 - 6x^2 + 11x - 6$, factorize and solve for its roots.

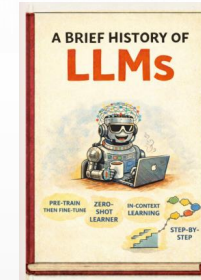
A: (2, 3, 1)

A "good" stepping-stone problem doesn't have to be perfectly correct or well-posed.

Analysis shows only **~36%** of the most effective generated problems (PQ) have correct solutions.

The structural and conceptual content of the question provides the crucial learning signal, even if the answer itself is noisy.

A BRIEF HISTORY OF LLMs FOR REASONING



Pre-2020: “Pre-train then fine-tune” (GPT-1, GPT-2)

Improving Language Understanding by Generative Pre-Training (Radford et al., 2018)

GPT-2 hinted at “zero-shot” abilities: model could perform tasks it wasn’t specifically trained on

2020: In-Context Learning (ICL)

Language Models are Few-Shot Learners (Brown et al., 2020)

At 175B parameters GPT-3 could learn patterns/take instructions from a prompt

2022: Chain-of-Thought (CoT) “Let’s think step-by-step”

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models (Wei et al., 2022)

For math/logic ICL wasn’t enough. prompting the model to "show its work" or "think step-by-step" skyrocketed performance on reasoning tasks

2024-25: Reinforcement learning from verifiable rewards (RLVF)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (DeepSeek-AI, 2024/2025) / OpenAI o1 / Kimi

Use RL & verifiable outcomes (math/code) to force the model to develop internal reasoning and self-correction

2026: Agents, “Self-Improvement” and “Autonomous AI”?

Models start to teach themselves. Human is left by the roadside.... (?)

LOOKING FORWARD

2026: Agents, “Self-Improvement” and “Autonomous AI”?

Models start to teach themselves.

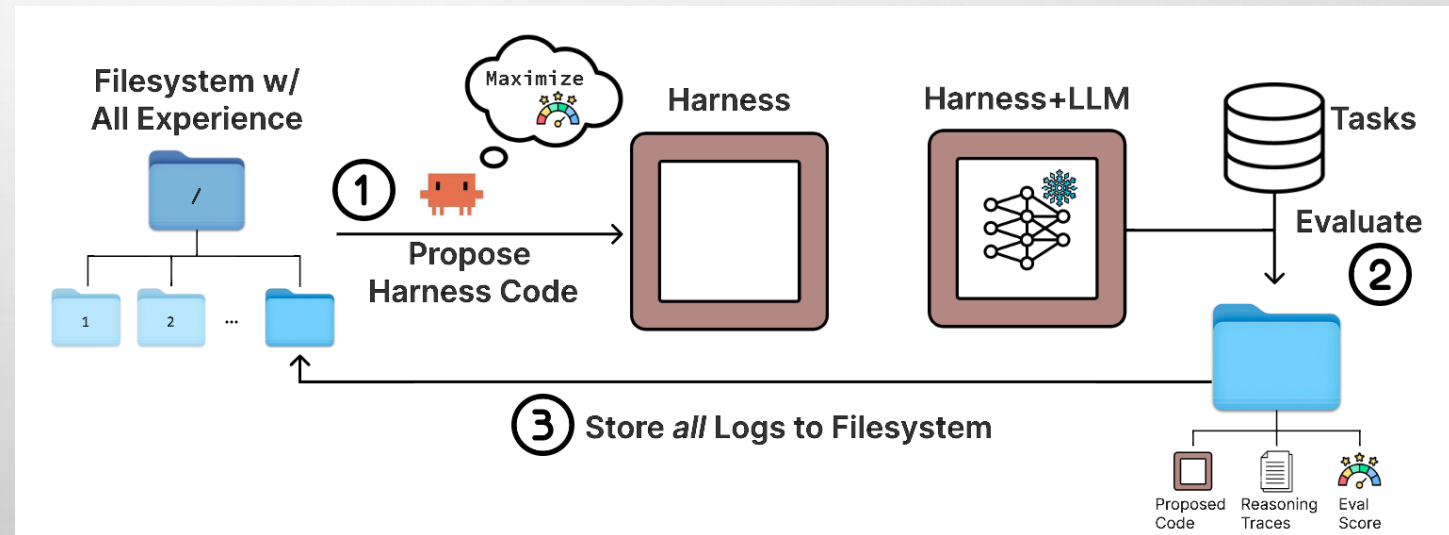
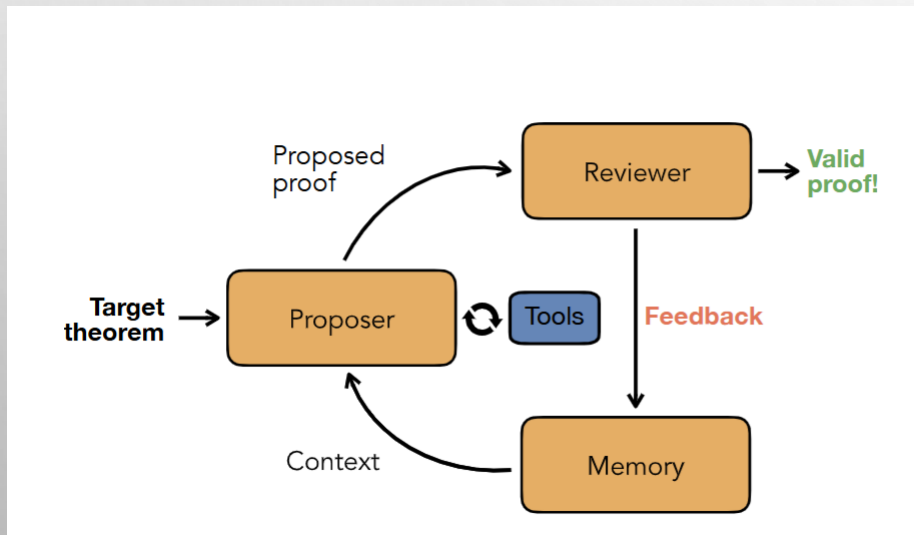
- Harnesses and multi-agent workflows
- (Some form of) continual learning and differentiation through contexts and skill.md files
- Allows for “meta-RL” frameworks

LOOKING FORWARD

2026: Agents, “Self-Improvement” and “Autonomous AI”?

Models start to teach themselves.

- Harnesses and multi-agent workflows
- (Some form of) continual learning and differentiation through contexts and skill.md files
- Allows for “meta-RL” frameworks

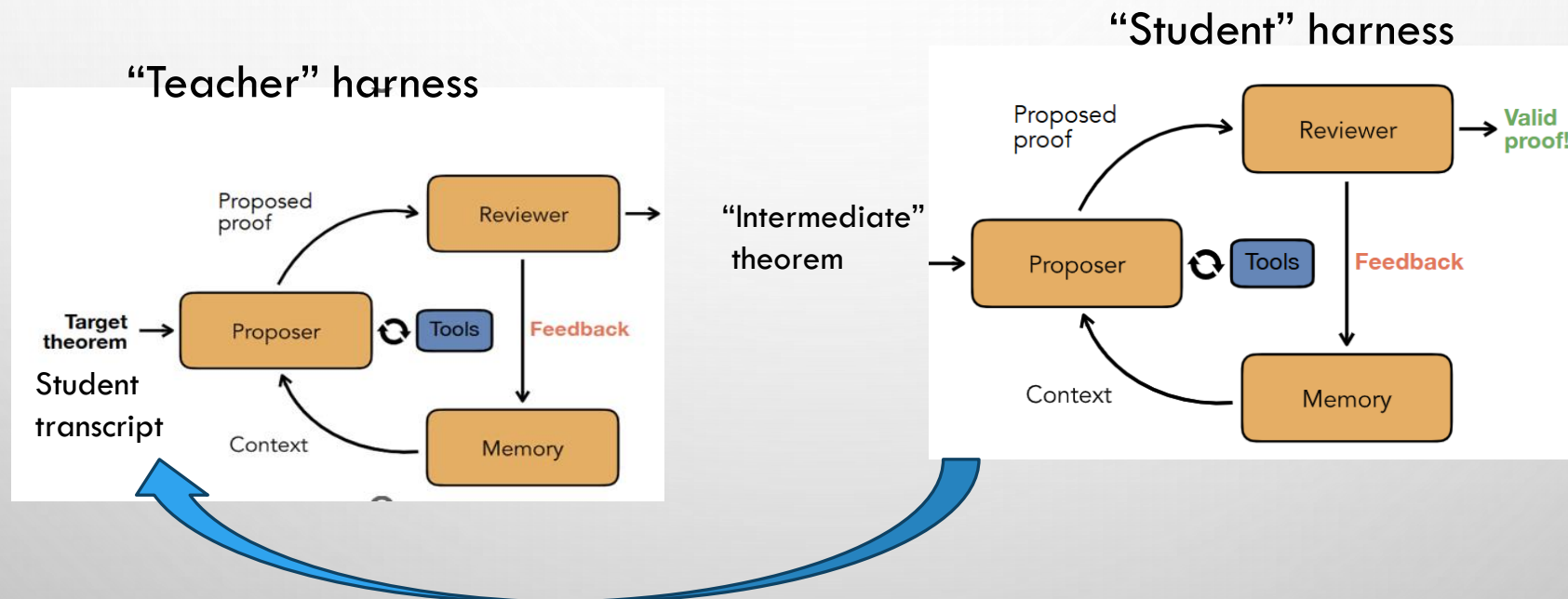


LOOKING FORWARD

2026: Agents, “Self-Improvement” and “Autonomous AI”?

Models start to teach themselves.

- Agentic (multi-) teacher-student setups

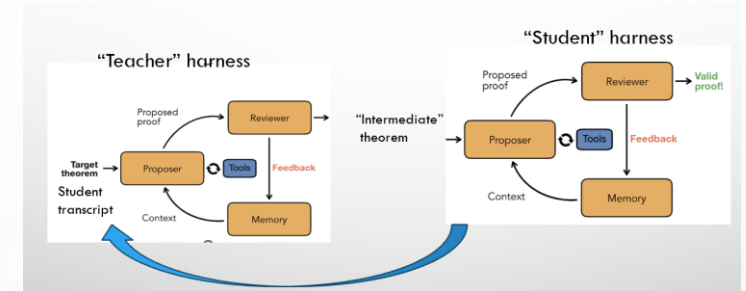


- Multi-agent capabilities > Single agent?

LOOKING FORWARD

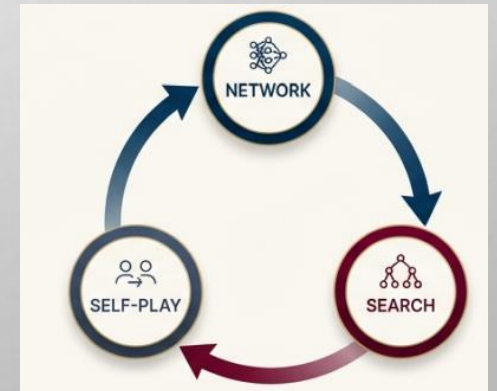
2026: Agents, “Self-Improvement” and “Autonomous AI”?

Models start to teach themselves.

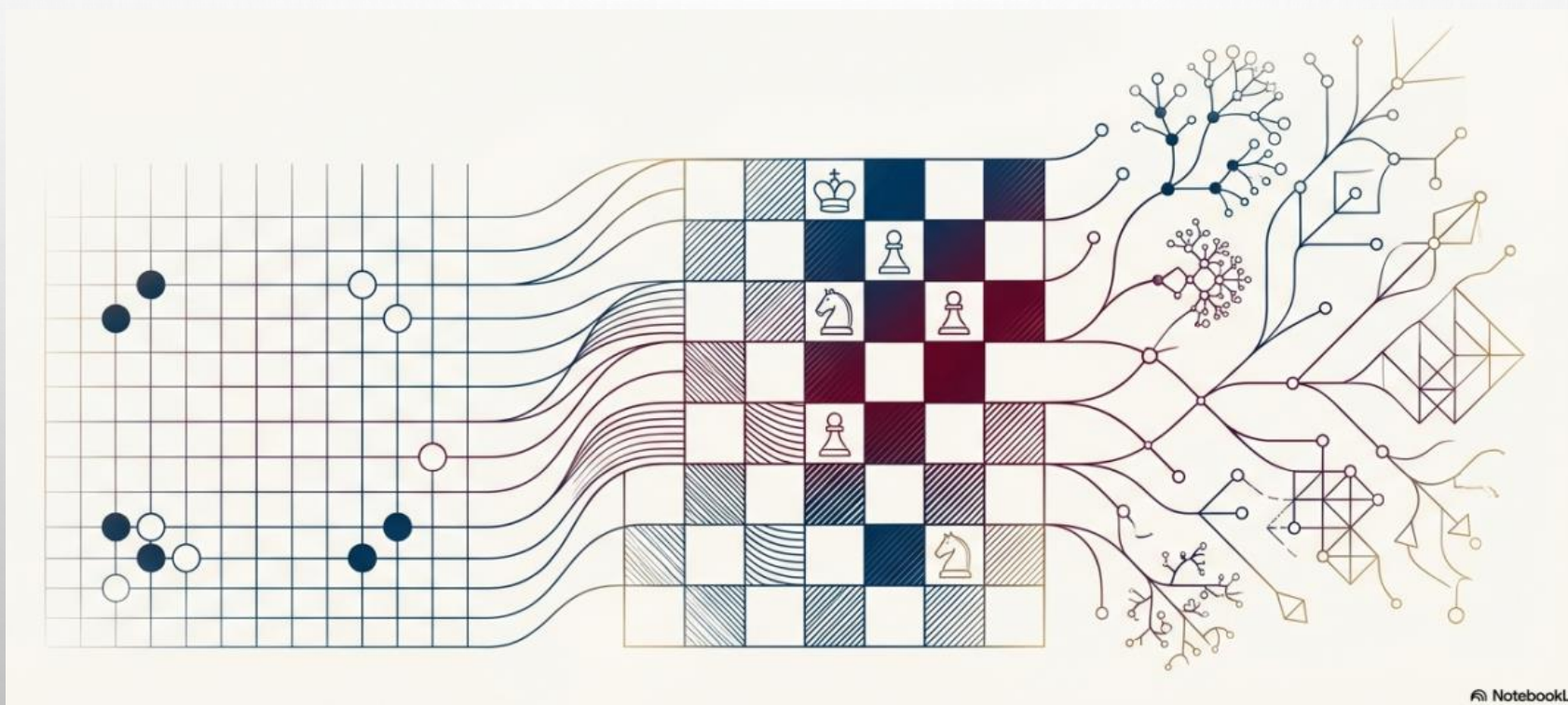


Can these self-improvement (self-play) loops transcend the knowledge in the training data (i.e. the “best” human?)

- **Data annotators are now the highest specialists in their fields.**
- **But what next? In what areas?**
- **Surely, for some domains like Go, games (AlphaGo, AlphaZero,...).**
- **Code.**
- **Math?**
- **General Reasoning?**
- **Planning and world modelling?**



THANKS !



COLLABORATORS AND REFERENCES

How Reinforcement Learning After Next-Token Prediction Facilitates Learning, ICLR'26 & EurIPS workshop oral

Nikolaos Tsilivis, Eran Malach, Karen Ullrich, JK



Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability, RSI workshop

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, JK

