Post-Training Reasoning Models: How LLMs Learn to Think and Act

Zhi Wang

UC San Diego

July 21, 2025

Outline

Introduction

- **Why do we need post-training?** *Prior to 2025: alignment, RLHF; Post 2025: give LLM time to think.*

Introduction

- **Why do we need post-training?** *Prior to 2025: alignment, RLHF; Post 2025: give LLM time to think.*
- **A Better Analogy:** It's unfair to compare infant learning to pretraining. It's more suitable to compare it to post-training, where innate knowledge (biological priors) already lays the ground.

Introduction

- **Why do we need post-training?** *Prior to 2025: alignment, RLHF; Post 2025: give LLM time to think.*
- **A Better Analogy:** It's unfair to compare infant learning to pretraining. It's more suitable to compare it to post-training, where innate knowledge (biological priors) already lays the ground.
- **Current Landscape:**
  - Scaling pre-training is hitting a wall.
  - Open-source models (e.g., DeepSeek R1) prove post-training's effectiveness.
  - Chain-of-Thought (CoT) gives the model time to think.

Key Questions

- How to introduce the dimension of time to LLMs?
- What are the paradigms for training LLMs to reason?
- Does RL lead to generalization? Where does the hype outpace science?
- What are the pathways forward in post training and inference time scaling?

From Static Answers to Dynamic Reasoning

Standard inference is atemporal, effectively a single computational step.

$$P(\text{answer}|\text{prompt})$$

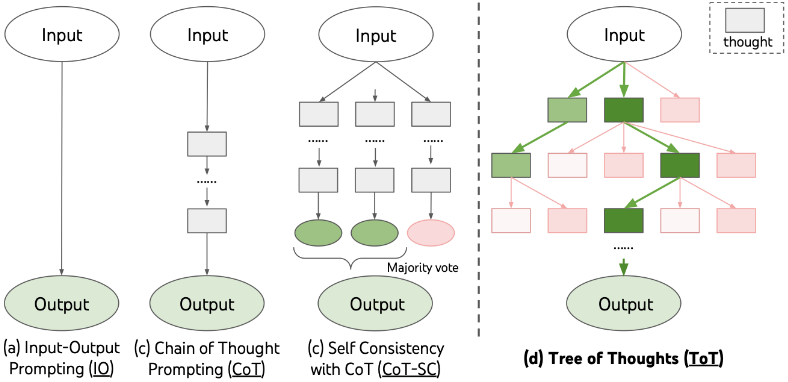
Chain of Thought (CoT): A Linear Timeline

Chain of Thought (CoT) refers to generating intermediate reasoning steps as part of the answer before producing the final output. Two main approaches include:

- **Few-shot prompting:** Including examples with reasoning steps to encourage the model to mimic the format.
- **Post-training:** Fine-tuning the model on CoT-annotated data using supervised or reinforcement learning.

The effectiveness is primarily empirical.

Beyond CoT


Input
Output
(a) Input-Output Prompting (IO)
Input
Output
(c) Chain of Thought Prompting (CoT)
Input
Majority vote
Output
(c) Self Consistency with CoT (CoT-SC)
Input
thought
Output
(d) Tree of Thoughts (ToT)

What is Supervised Fine-Tuning (SFT)?

Definition

Supervised Fine-Tuning (SFT) refers to the process of training a **pretrained language model** on a **labeled dataset** using **supervised learning**, typically to improve task-specific performance or teach desired behavior.

What is Supervised Fine-Tuning (SFT)?

Definition

Supervised Fine-Tuning (SFT) refers to the process of training a **pretrained language model** on a **labeled dataset** using **supervised learning**, typically to improve task-specific performance or teach desired behavior.

Formal Objective

Given a pretrained model f_θ, SFT updates parameters θ to minimize:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\log P_\theta(y \mid x)\right]$$

where (x, y) are input-output pairs from labeled dataset $\mathcal{D}$.

What is Supervised Fine-Tuning (SFT)?

Definition

Supervised Fine-Tuning (SFT) refers to the process of training a **pretrained language model** on a **labeled dataset** using **supervised learning**, typically to improve task-specific performance or teach desired behavior.

Formal Objective

Given a pretrained model f_θ, SFT updates parameters θ to minimize:

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[-\log P_\theta(y \mid x)\right]$$

where (x, y) are input-output pairs from labeled dataset $\mathcal{D}$.

In the Context of Reasoning

Fine-tuning on reasoning examples like **Chain-of-Thought (CoT)**. Teaches the model to *imitate* reasoning patterns.

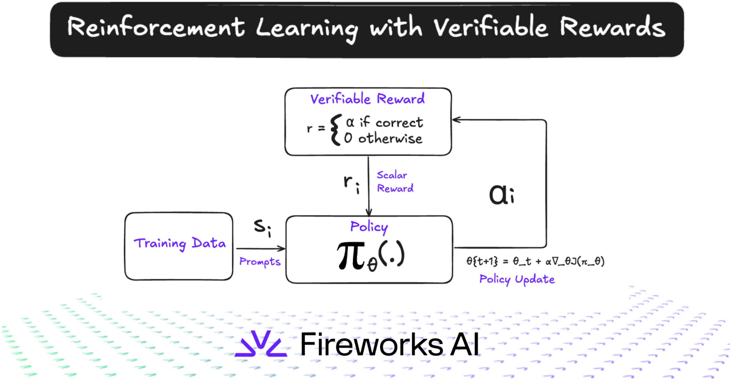
Reinforcement Learning with Verifiable Rewards (RLVR)

What is RLVR?

Rewarding the model for correctness based on binary, verifiable checks (e.g., does the code compile? is the math answer correct?).

What is RLVR?

Rewarding the model for correctness based on binary, verifiable checks (e.g., does the code compile? is the math answer correct?).



What is GRPO in RLVR?

- action = generating a new token
- binary rewards (e.g., correct = 1, incorrect = 0).
- No extra reward model — only requires verifiable correctness.
- Rewards are normalized **within a group** (not necessary).
- Updates keep the new policy close to a reference model.

Why "Group"?

- GRPO uses **relative performance** within each group to determine which rollout is desired.

GRPO Objective

Minimize Clipped Objective with Normalized Advantage

$$Surrogate_{i,t} = \min\left(r_{i,t} \cdot \hat{A}_{i,t},\ \text{clip}(r_{i,t}) \cdot \hat{A}_{i,t}\right)$$

$$\mathcal{L} = \frac{1}{num\ rollouts} \sum_t \frac{1}{|seq\ length|} (\sum_t surrogate_{i,t} - \beta \cdot per\text{-}token\ KL)$$

- $r_{i,t}$: token-level importance weight (new policy / old policy).
- $\hat{A}_{i,t}$: **normalized group advantage within group i**:

$$\hat{A}_{i,t} = \frac{r_i - \mu}{\sigma}$$

- KL: measures the *distance* between two distributions.

GRPO Intuition in a Group

Example: Group of 4 Responses

A: wrong $\rightarrow$ 0
B: right $\rightarrow$ 1
C: right $\rightarrow$ 1
D: wrong $\rightarrow$ 0

- Group mean: 0.5, std: 0.5
- Normalized advantage:

$$\hat{A}_{\mathrm{B,C}} = +1, \quad \hat{A}_{\mathrm{A,D}} = -1$$

- Policy is updated to favor B and C over A and D.

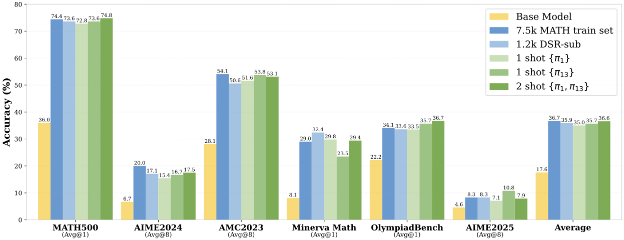
Reinforcement Learning for Reasoning in Large Language Models with One Training Example

(Wang et al. 2025)

- Explores data selection for RLVR – just ONE training example is enough.
- Focuses on mathematical reasoning capabilities.
- New phenomena like post-saturation generalization and the role of different loss components.

Remarkable Performance with One Example

- **Key Finding:** RLVR with a single example (1-shot RLVR) can match performance of training with thousands. This matched training on 1.2k DSR-sub; 2-shot RLVR slightly exceeded it. Base model is Qwen2.5-Math-1.5B.



Role of Exploration & Entropy Loss

- Policy gradient loss is the main driver of improvement.
- Critically, promoting exploration (e.g., via entropy loss and temperature) improves model performance.
- *Comment:* Learning is likely driven by trying out different variations which leads to non-trivial policy gradient.

Table 6: **Entropy loss alone with π_1 can still improve model performance.**

Model	MATH 500	AIME24 2024	AMC23 2023	Minerva Math	Olympiad-Bench	AIME 2025	Avg.
Qwen2.5-Math-1.5B	36.0	6.7	28.1	8.1	22.2	4.6	17.6
+Entropy Loss, Train 20 step	63.4	8.8	33.8	14.3	26.5	3.3	25.0
Llama-3.2-3B-Instruct	40.8	8.3	25.3	15.8	13.2	1.7	17.5
+Entropy Loss, Train 10 step	47.8	8.8	26.9	18.0	15.1	0.4	19.5
Qwen2.5-Math-7B	51.0	12.1	35.3	11.0	18.2	6.7	22.4
+Entropy Loss, Train 4 step	57.2	13.3	39.7	14.3	21.5	3.8	25.0

1-Shot RLVR The "Reranking" Hypothesis

- The success of 1-shot RLVR suggests that RL is "activating" or making more accessible latent capabilities rather than teaching entirely new ones from scratch with just one example.
- If one example can trigger such broad improvements, those improved reasoning paths were likely already possible for the base model, just not efficiently sampled.

Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild
(Zeng et al. 2025)

- Explores "zero RL training": RL directly on pretrained base LLMs.
- No initial Supervised Fine-Tuning (SFT) for instruction following.
- Investigated across 10 diverse open base models (LLama3, Mistral, DeepSeek-Math, Qwen2.5 series).

Zero RL: Broad Effectiveness

Achieved with simple rule-based rewards (+1 correct, 0 incorrect) $\sim$8K samples.

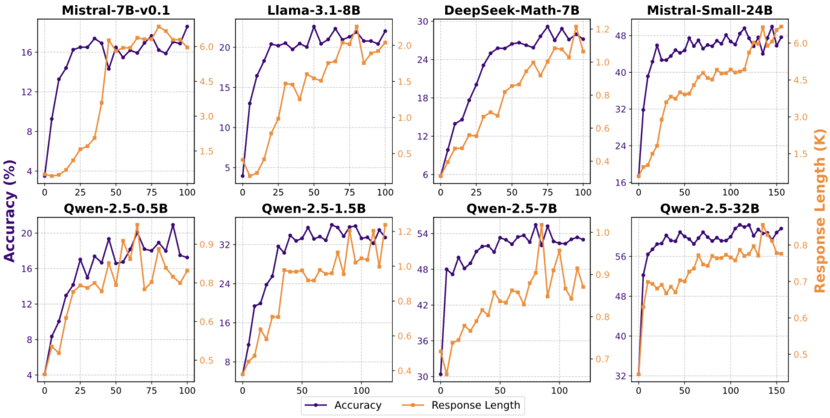


Figure 1: Accuracy and response length across training iterations for different models, averaged on GSM8K, MATH500, Minerva Math, OlympiadBench, AIME24, and AMC23. Per-benchmark results are in Figure 11 (Appendix D). All training starts from base models.

- **Key Finding:** Over-reliance on rigid format rewards (e.g., '\boxxed') is detrimental. Can lead to lower performance ceilings and "overthinking."
- Also penalizes exploration.

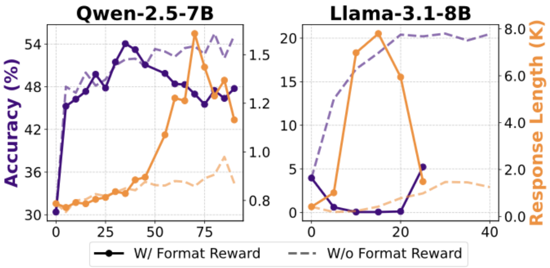


Figure 6: Accuracy and response length with and without format rewards.

SFT's Impact on Performance in Reasoning

- Leads to diminished post-RL performance (lower max accuracy/length).
- Negative impact more severe with more initial SFT steps (using NuminaMath).

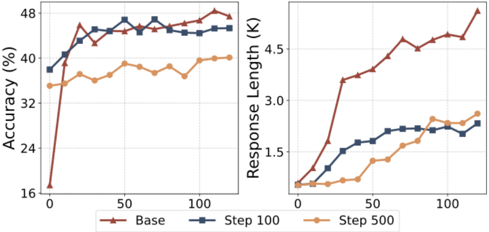


Figure 9: Accuracy and response length averaged on the six benchmarks over RL training iterations after running different SFT steps as starting points.

The DeepSeek R1 Pipeline (Part 1: Building the Engine)

From a generalist model to a specialized reasoner

Base Model: DeepSeek-V3

Stage 1: Cold-Start SFT

Goal: Avoid the "cold start" problem of pure RL and teach the model the **basic output format.** **Method:** A light round of

Supervised Fine-Tuning on **a small, human-refined dataset** of reasoning examples.

Stage 2: Reasoning-Oriented RL

Goal: Develop the core problem-solving and reasoning abilities. **Method:** Large-scale

Reinforcement Learning using **GRPO** (Group Relative Policy Optimization) with rule-based rewards (e.g., **accuracy, format checks**).

The DeepSeek R1 Pipeline (Part 2: Refinement)

From a specialist to a robust, general-purpose reasoner

Stage 3: Rejection Sampling + SFT

Goal: Internalize the best reasoning paths generated by the model itself. **Method:** Automatically select the highest-scoring outputs from Stage 2 and **use this "golden" data for another round of SFT.**

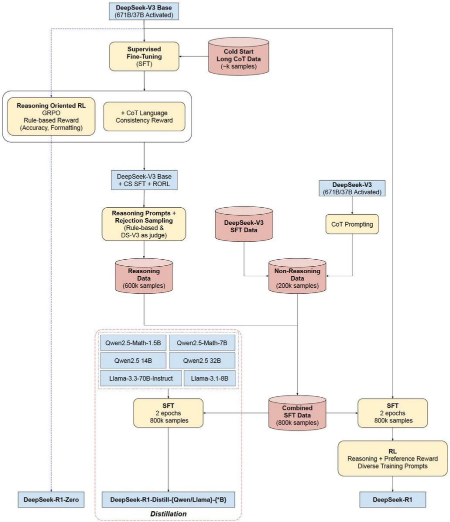
Stage 4: Final RL for All Scenarios

Goal: Ensure the model is helpful and harmless across all tasks, not just reasoning. **Method:** A final RL phase on a diverse set of prompts, combining **reasoning rewards with general preference scores.**

Key Takeaway

The pipeline cleverly alternates between RL (to explore and discover reasoning) and SFT (to distill and stabilize the learned behaviors).

DeepSeek R1



DeepSeek-V3 Base (671B/37B Activated)
Supervised Fine-Tuning (SFT)
Cold Start Long CoT Data (~k samples)
Reasoning Oriented RL GRPO Rule-based Reward (Accuracy, Formatting)
+ CoT Language Consistency Reward
DeepSeek-V3 Base + CS SFT + RORL
DeepSeek-V3 (671B/37B Activated)
Reasoning Prompts + Rejection Sampling (Rule-based & DS-V3 as judge)
DeepSeek-V3 SFT Data
CoT Prompting
Reasoning Data (600k samples)
Non-Reasoning Data (200k samples)
Qwen2.5-Math-1.5B
Qwen2.5-Math-7B
Qwen2.5 14B
Qwen2.5 32B
Llama-3.3-70B-Instruct
Llama-3.1-8B
SFT 2 epochs 800k samples
Combined SFT Data (800k samples)
SFT 2 epochs 800k samples
RL Reasoning + Preference Reward Diverse Training Prompts
DeepSeek-R1-Zero
DeepSeek-R1-Distill-{Qwen/Llama}-{*B}
Distillation
DeepSeek-R1

Inference-Time Scaling – Metrics & Techniques

Definition

Improving reasoning performance at inference time, without additional training.

Inference-Time Scaling – Metrics & Techniques

Definition

Improving reasoning performance at inference time, without additional training.

Core Metrics

- pass@k – success if any of k generated outputs is correct.
- maj@k – accuracy determined by majority vote among k candidates.
- avg@k – average correctness across k samples.

Inference-Time Scaling – Metrics & Techniques

Definition

Improving reasoning performance at inference time, without additional training.

Core Metrics

- pass@k – success if any of k generated outputs is correct.
- maj@k – accuracy determined by majority vote among k candidates.
- avg@k – average correctness across k samples.

Techniques

- Chain-of-Thought Prompting
- Self-Consistency (voting across multiple samples)
- Temperature & Top-k/Top-p Tuning
- Tree-of-Thoughts Search

GenSelect from AIMO-2 (Moshkov et al., 2025)

What is GenSelect?

- An inference-time algorithm that selects the best answer from k generated candidates using a learned selector model.
- Trained on tuples of `<problem, k candidates, correctness>`.
- Designed to approach the performance of **pass@**k, while outputting a single answer.

GenSelect from AIMO-2 (Moshkov et al., 2025)

What is GenSelect?

- An inference-time algorithm that selects the best answer from k generated candidates using a learned selector model.
- Trained on tuples of `<problem, k candidates, correctness>`.
- Designed to approach the performance of **pass@**k, while outputting a single answer.

How It Works at Inference

1. Generate k candidate reasoning chains (CoT or tool-integrated).
2. Use GenSelect to rank and select the best candidate.
3. Return the selected candidate as final output.

No model retraining is needed—GenSelect operates entirely at inference time.

Citation: Moshkov et al. (2025), *AIMO-2 Winning Solution*, arXiv:2504.16891

GenSelect: Bridging Metric and Deployment

- **pass@**k is an idealized metric:
  - Measures the chance that *at least one* of k generations is correct.
  - Assumes access to a perfect verifier (e.g., test cases or oracle).
- **Limitation:** Not usable directly during inference.
- **GenSelect:** Turns pass@k into a *deployable algorithm*.
  - Trains a selector to choose the best from k candidates.
  - Uses learned signals to approximate the oracle.

Unexplained Phenomena:

- One-shot RLVR
- Self-post-train without examples
- Intuitor (RLIF): Self-certainty as reward
- Spurious rewards

Shaky Scientific Ground?

Is the hype real?

Recent claims of RL's effectiveness are being questioned.

Shaky Scientific Ground?

Is the hype real?

Recent claims of RL's effectiveness are being questioned.

- **Incorrect Baseline Evaluations:** Do these studies use the same set of **temperature, prompt, and answer extractor for benchmarking? YES.**

Shaky Scientific Ground?

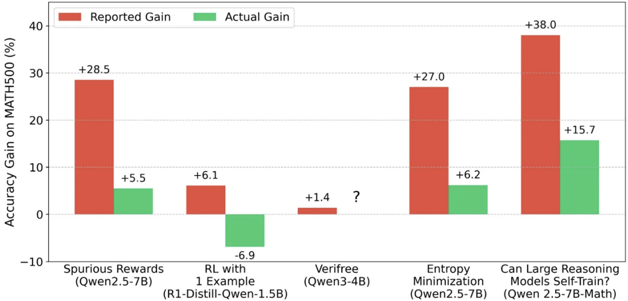
Is the hype real?

Recent claims of RL's effectiveness are being questioned.

- **Incorrect Baseline Evaluations:** Do these studies use the same set of **temperature, prompt, and answer extractor for benchmarking? YES.**
- **The Qwen 2.5 Problem:** This model is consistently used to show RL's math prowess.
  - Does it have inherently strong math skills? **YES.**
  - Or has it been contaminated with benchmark data? **YES.**

Blog: Incorrect Baseline Evaluations

Snapshot: MATH 500 Results



▼ Table for the above plot with paper links, and links to sources for actual pre-RL accuracies

Method/Paper	Reported RL Acc	Reported Pre-RL Acc	Actual Pre-RL Acc
Spurious Rewards (Qwen2.5-7B)	70.1	41.6	64.6
RL with 1 Example (DeepSeek-R1-Distill-1.5B)	78.0	71.9	83.9
Verifree (Qwen3-4b)	74.8	73.4	??
Entropy Minimization (Qwen2.5-7B)	70.8	43.8	64.6

Paper: Data Contamination

Example from MATH-500

Problem: For how many positive integers $n > 1$ is it true that 2^{24} is a perfect n^{th} power?
Answer: $\boxed{7}$

Partial Prompt

Prompt: For how many positive integers $n > 1$ is

Qwen2.5-7B Output

it true that 2^{24} is a perfect n^{th} power?

To determine for how many positive integers $n > 1$ the number 2^{24} is a perfect n-th power...
First, we need to find...
...
Let's count these values. There are 7 such values...
...
The final answer is $\boxed{7}$.

Llama3.1-8B Output

the set of positive divisors of n (including 1 and n) less than 50 ?
(A) 15 **(B)** 16 **(C)** 17 **(D)** 18 **(E)** 19

The number of positive divisors of n is given by...
...
The number of factors of each of these numbers is...
...
The answer is $\boxed{\textbf{(B)}\ 16}$

contamination-free dataset construction

5-Step Calculation

Problem: Evaluate this LaTeX numerical expression step-by-step and give the final value within \boxed{}:

$$45^2 - \frac{94}{6}/(\frac{76}{4}/\frac{19}{5} - 35^3) + 81^2$$

Answer: $\boxed{8586.00036544592}$

10-Step Calculation

Problem: Evaluate this LaTeX numerical expression step-by-step and give the final value within \boxed{}:

$$\frac{94}{2} + \left(\frac{73^2 \cdot (62 - 10)}{\left(\frac{\frac{65}{9}+47}{\frac{\frac{49}{7}\cdot 81}{62^2}}\right)}\right) \cdot \left(\frac{41}{6} + \frac{12}{7}\right)$$

Answer: $\boxed{6490.42220471333}$

Figure 2: Examples of *RandomCalculation* dataset.

RandomCalculation shows only correct signal works

correct → steady improvment, random → unstable, inverted → collapse.
No surprise!

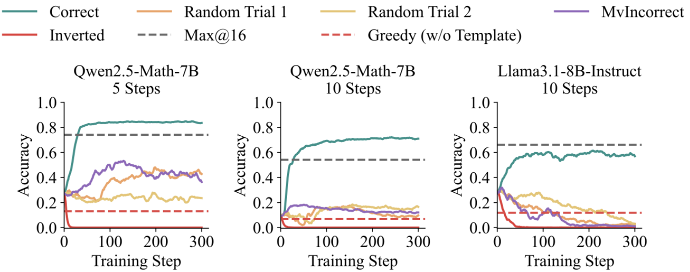
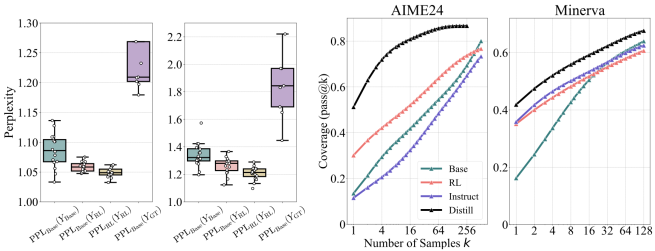


Figure 7: Training performance of Qwen2.5-Math-7B and Llama3.1-8B-Instruct using the RLVR algorithm on the ***RandomCalculation*** dataset. Results are presented for datasets with 5-step and 10-step calculations.

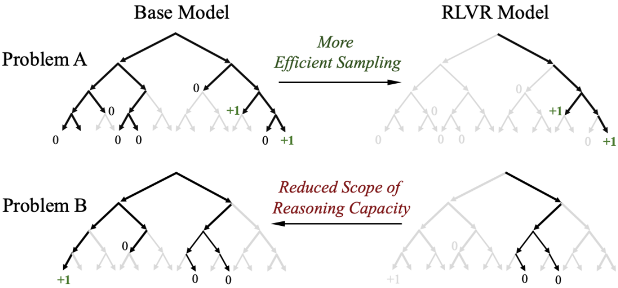
RLVR: No Fundamentally New Reasoning Patterns

- **Key Finding:** "Surprisingly, our findings demonstrate that RLVR does not elicit fundamentally new reasoning patterns."
- **Reasoning paths from RLVR models are largely already present within the base model's potential outputs.**
- Lower perplexity indicates that the model has a higher likelihood of generating this response.
- This is reported last year in DeepSeekMath paper as well.



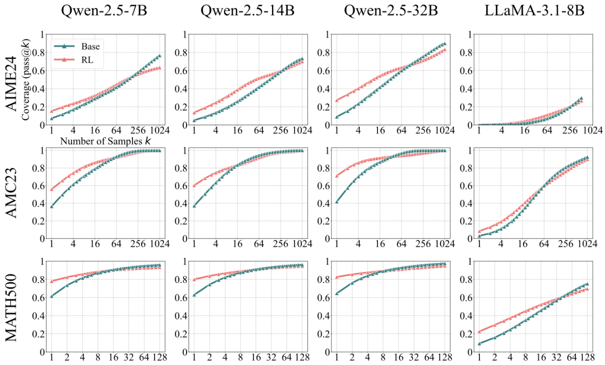
RL's Main Role: Enhanced Sampling Efficiency

- "Instead, RL primarily enhances the efficiency of LLMs in sampling existing correct reasoning paths encoded in the base model."
- RLVR improves pass@1 by making it easier to find these existing correct paths.



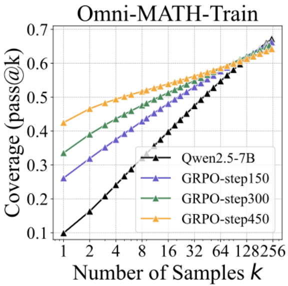
Base Models' Potential at Large 'k'

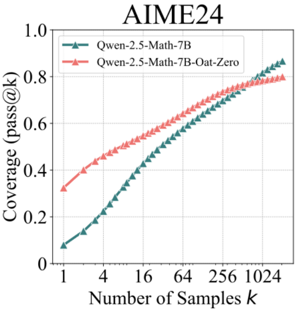
- While RL-trained models lead at small 'k' (e.g., pass@1), base models often match or exceed them at large 'k' values.
- This indicates base models can solve these problems if allowed more attempts.



Reasoning Boundary Capped by Base Model

- **Key Finding:** "Consequently, the reasoning boundary remains limited by the base model's capabilities."
- Coverage (pass@k) for a dataset is the proportion of problems in that dataset that the model can solve within k trials.
- Solvable problems by RL model often form a subset (not just fewer) of the base model's.





Evidence of Subset Relationship

- Analysis of solvable problem sets supports the subset argument.

Table 4: Indices of solvable problems in AIME24 (starting from 0). An approximate subset relationship can be observed: most problems solved by the RL model are also solvable by the base model.

Models	Problem Indices
Qwen-7B-Base	0, 1, 4, 6, 7, 8, 9, 11, 12, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 26, 27, 28, 29
SimpleRL-Qwen-7B	0, 1, 6, 7, 8, 9, 12, 14, 15, 16, 18, 22, 23, 24, 25, 26, 27, 28, 29

Table 5: Indices of solvable problems in LiveCodeBench (ranging from 400 to 450, starting from 0).

Model	Solvable Problem Indices
Qwen-7B-Instruct-1M	400, 402, 403, 407, 409, 412, 413, 417, 418, 419, 422, 423, 427, 432, 433, 436, 438, 439, 440, 444, 445, 448, 449
Coder-R1	400, 402, 403, 407, 412, 413, 417, 418, 419, 422, 423, 427, 430, 433, 438, 439, 440, 444, 445, 449

Current RL Algorithms: Suboptimal Efficiency

- "Furthermore, our in-depth analysis reveals that current RL algorithms are far from achieving the optimal sampling efficiency, defined by the reasoning boundary of the base model."
- A "Sampling Efficiency Gap" ($\Delta_{SE} = \textit{base model's pass@256} - \textit{RL model's pass@1}$) persists across various RL methods.

Open Questions

Combining SFT and RL

How can we best integrate the stability of SFT with the optimization power of RL?

Open Questions

Combining SFT and RL

How can we best integrate the stability of SFT with the optimization power of RL?

Effective RL Training

How do we optimize the RL process itself? (e.g., the 80/20 rule, selective rollouts).

Open Questions

Combining SFT and RL

How can we best integrate the stability of SFT with the optimization power of RL?

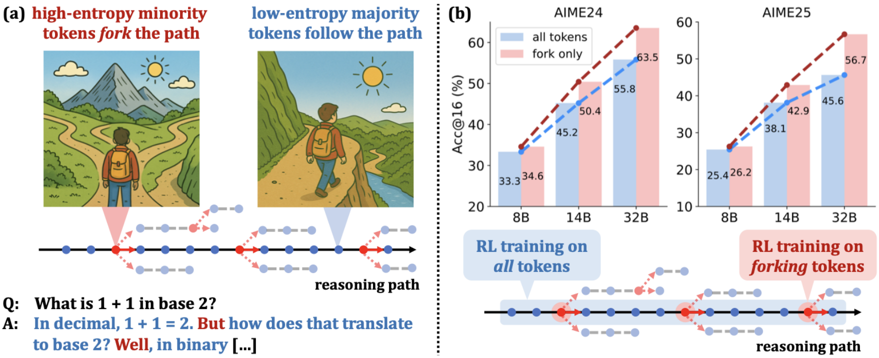
Effective RL Training

How do we optimize the RL process itself? (e.g., the 80/20 rule, selective rollouts).

Latent Reasoning

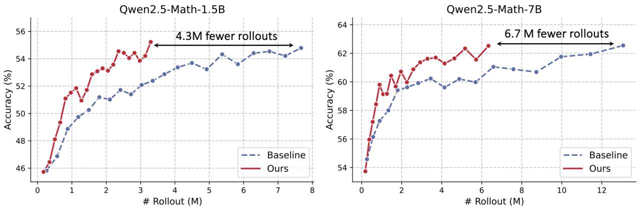
Can we encourage continuous, internal "thought" processes in LLMs? (e.g., recurrent blocks, chain of continuous thoughts).

Paper: The 80/20 Rule



Paper: Selective Rollouts

Our analysis of reward dynamics reveals a strong temporal consistency in prompt value: prompts that are uninformative in one epoch of training are likely to remain uninformative in future epochs.



Thank You & Q&A

Questions?