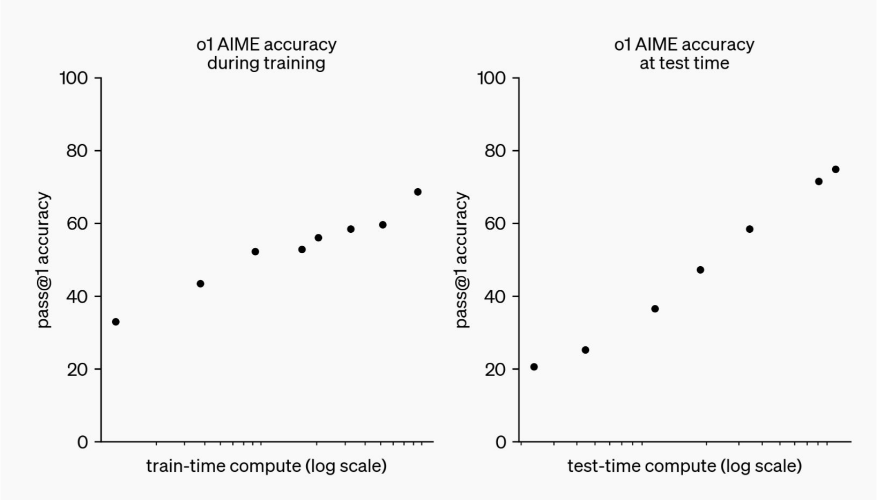
Understanding Reinforcement Learning inside DeepSeek-R1

Tianbing Xu

Jan. 2025

Learning to reason with LLMs (OpenAI 2024)

Our large-scale **reinforcement learning** algorithm teaches the model how to think productively using its **chain of thought** in a highly *data-efficient training* process.



Deepseek-R1

Benchmark Results (DeepSeek-AI 2025)

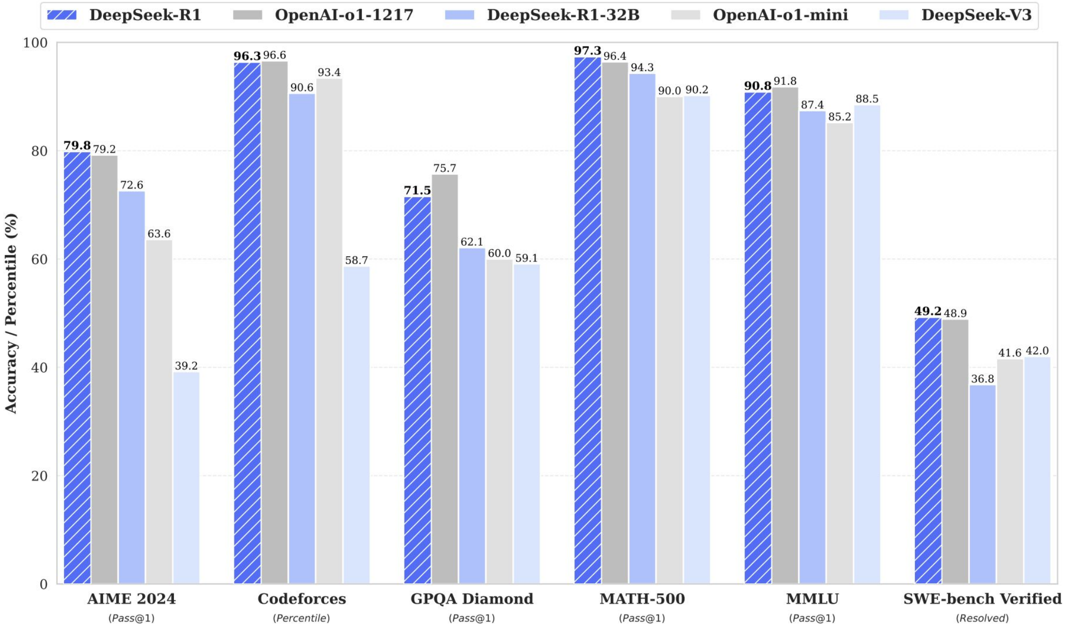
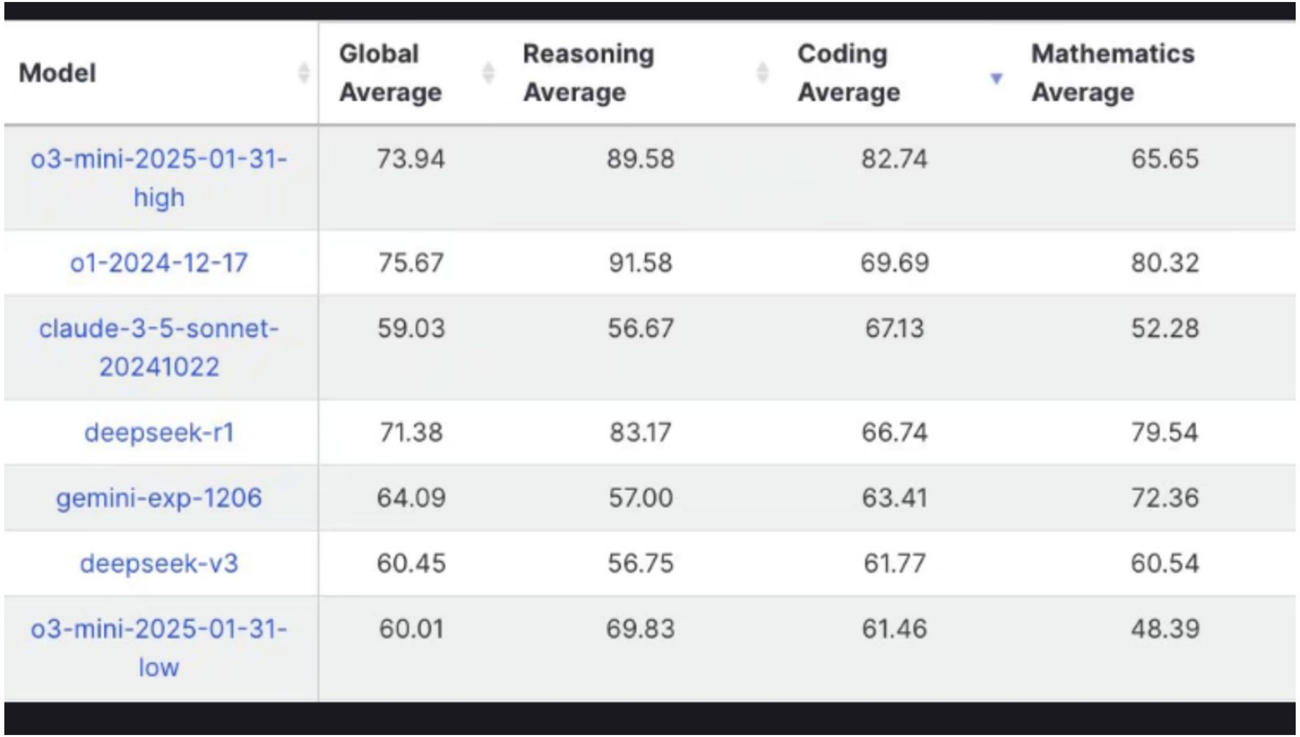


Figure 1 | Benchmark performance of DeepSeek-R1.

Benchmark Results (O3, O1, R1)



Model	Global Average	Reasoning Average	Coding Average	Mathematics Average
o3-mini-2025-01-31-high	73.94	89.58	82.74	65.65
o1-2024-12-17	75.67	91.58	69.69	80.32
claude-3-5-sonnet-20241022	59.03	56.67	67.13	52.28
deepseek-r1	71.38	83.17	66.74	79.54
gemini-exp-1206	64.09	57.00	63.41	72.36
deepseek-v3	60.45	56.75	61.77	60.54
o3-mini-2025-01-31-low	60.01	69.83	61.46	48.39

Reinforcement Learning from Strong Base Model

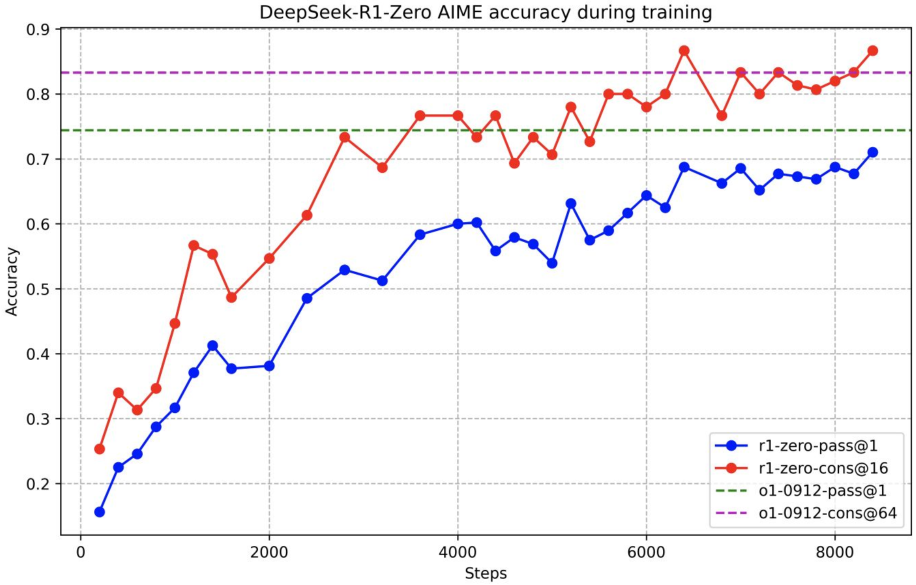


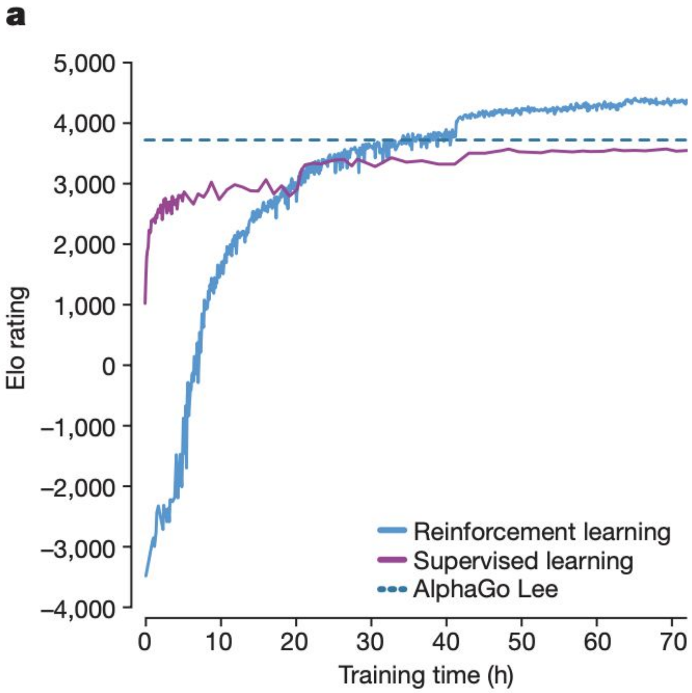
Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

Open Questions for **Learning to Reason via RL**

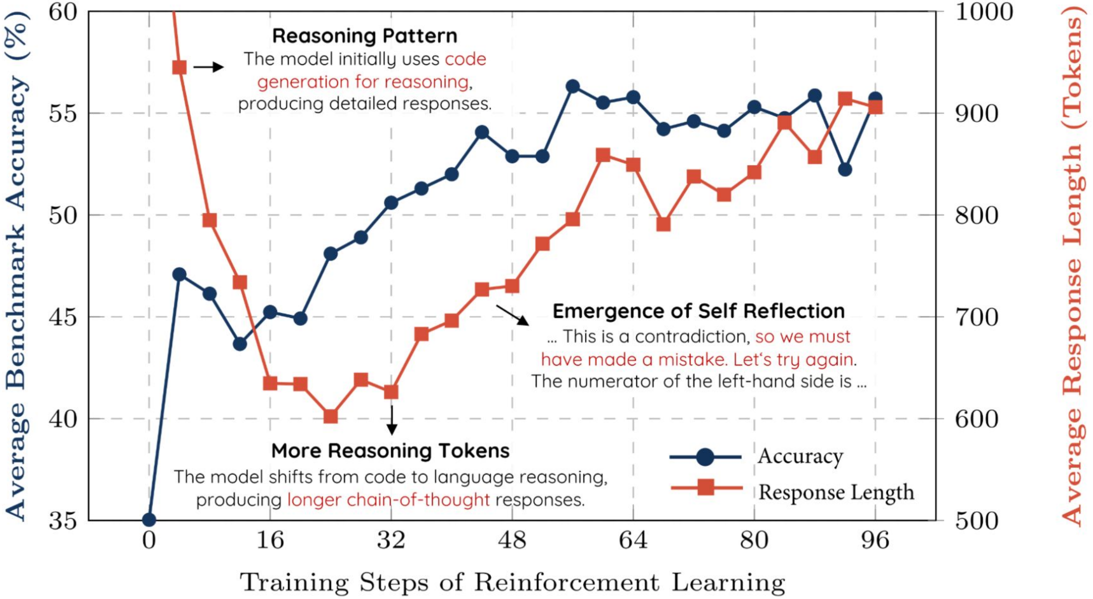
1. Can Emergent Reasoning Arise via Reinforcement **Learning from Scratch** (Using a Random Base Model)?
2. Can Emergent Reasoning Arise via Reinforcement **Learning from Weak Base Model**?
3. How Do Small Reasoning Models Compare to Large General Models?

Reinforcement Learning from Scratch (AlphaGo-Zero, 2017)

without human knowledge (~80M model)



Reinforcement Learning from Weak Base Model (Simple R1, 2025)



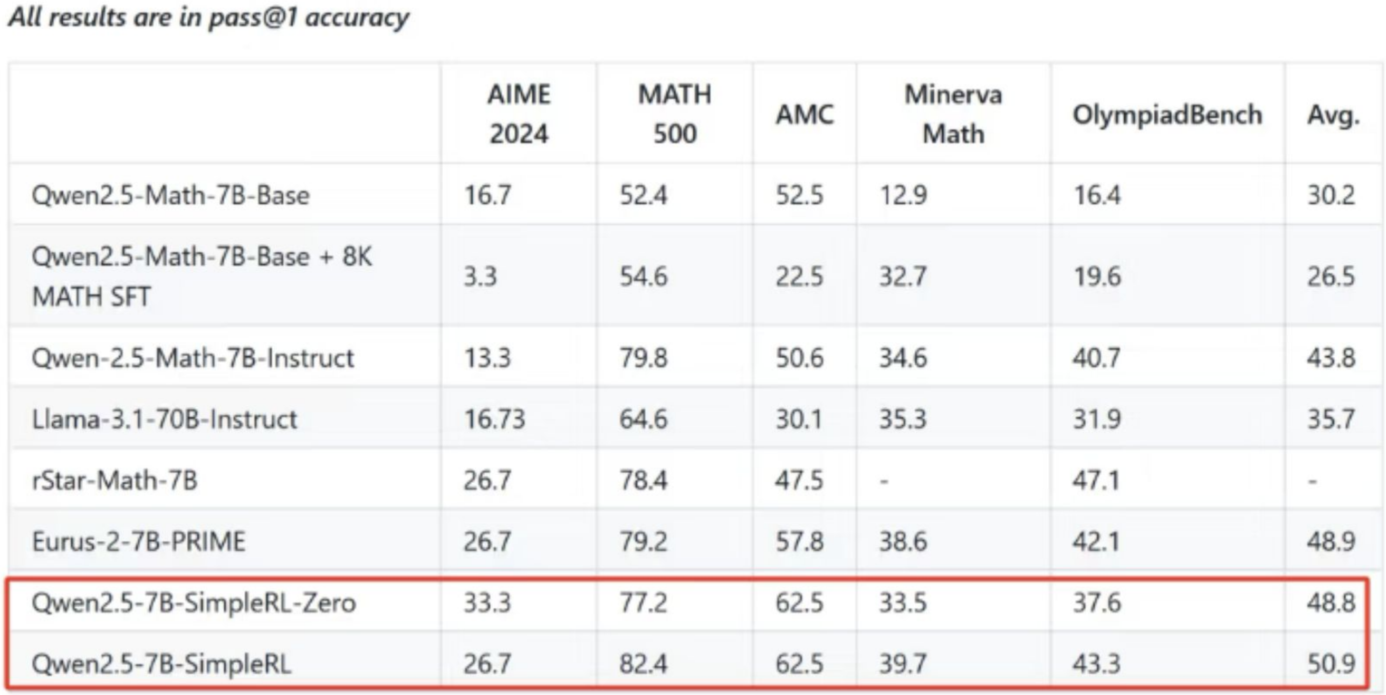
Reasoning Pattern
The model initially uses code generation for reasoning, producing detailed responses.

Emergence of Self Reflection
... This is a contradiction, so we must have made a mistake. Let's try again. The numerator of the left-hand side is ...

More Reasoning Tokens
The model shifts from code to language reasoning, producing longer chain-of-thought responses.

> Training dynamics of our Qwen2.5-SimpleRL-Zero training starting from the Qwen2.5-Math-7B, without SFT or reward models.

How Good: Small Reasoning Models vs Large General Models?



All results are in pass@1 accuracy

	AIME 2024	MATH 500	AMC	Minerva Math	OlympiadBench	Avg.
Qwen2.5-Math-7B-Base	16.7	52.4	52.5	12.9	16.4	30.2
Qwen2.5-Math-7B-Base + 8K MATH SFT	3.3	54.6	22.5	32.7	19.6	26.5
Qwen-2.5-Math-7B-Instruct	13.3	79.8	50.6	34.6	40.7	43.8
Llama-3.1-70B-Instruct	16.73	64.6	30.1	35.3	31.9	35.7
rStar-Math-7B	26.7	78.4	47.5	-	47.1	-
Eurus-2-7B-PRIME	26.7	79.2	57.8	38.6	42.1	48.9
Qwen2.5-7B-SimpleRL-Zero	33.3	77.2	62.5	33.5	37.6	48.8
Qwen2.5-7B-SimpleRL	26.7	82.4	62.5	39.7	43.3	50.9

Learning to improve reasoning with more thinking

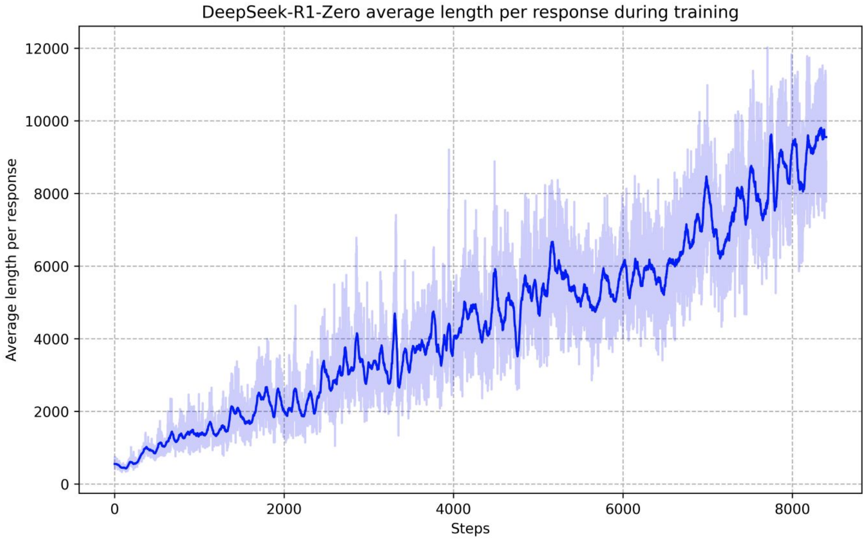
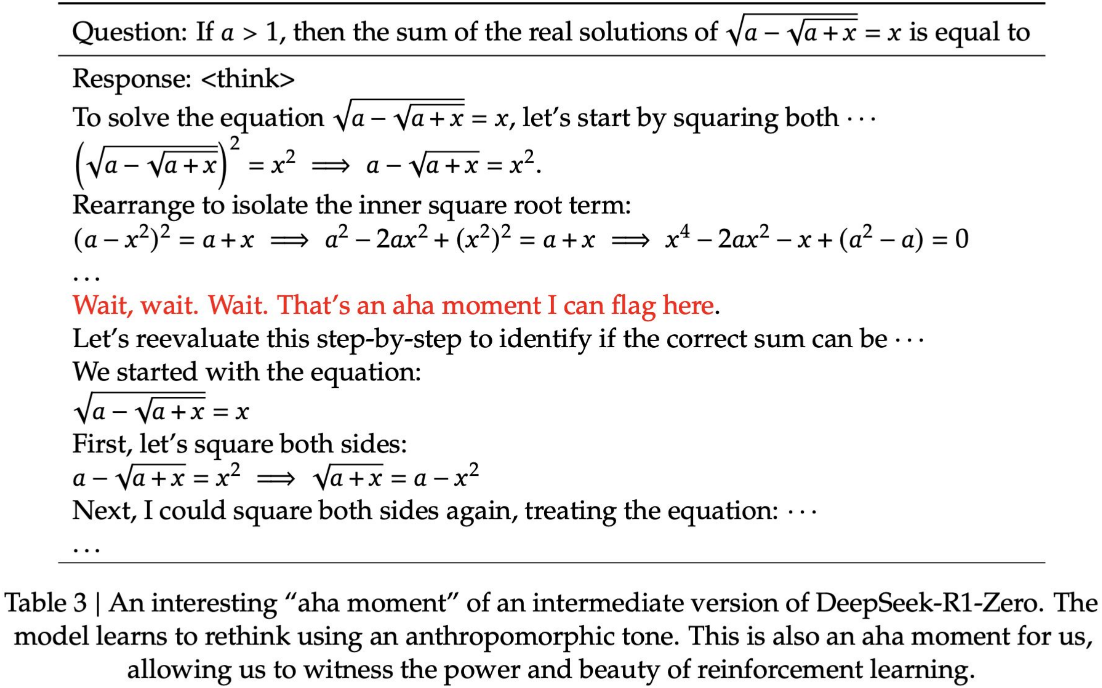


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

aha moment



Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$

$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$

Rearrange to isolate the inner square root term:

$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$

We started with the equation:

$\sqrt{a - \sqrt{a + x}} = x$

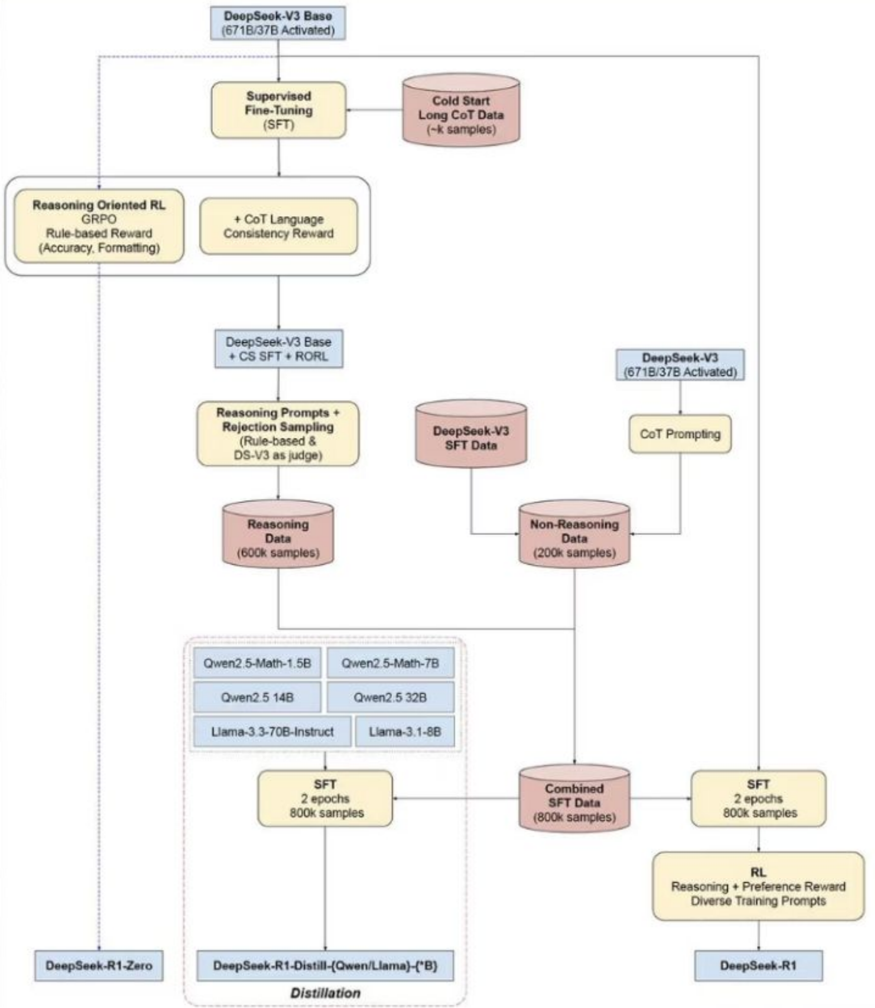
First, let's square both sides:

$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$

Next, I could square both sides again, treating the equation: $\cdots$

...

Table 3 | An interesting "aha moment" of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.


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

Reinforcement Learning: From PPO to GRPO

Reinforcement Learning

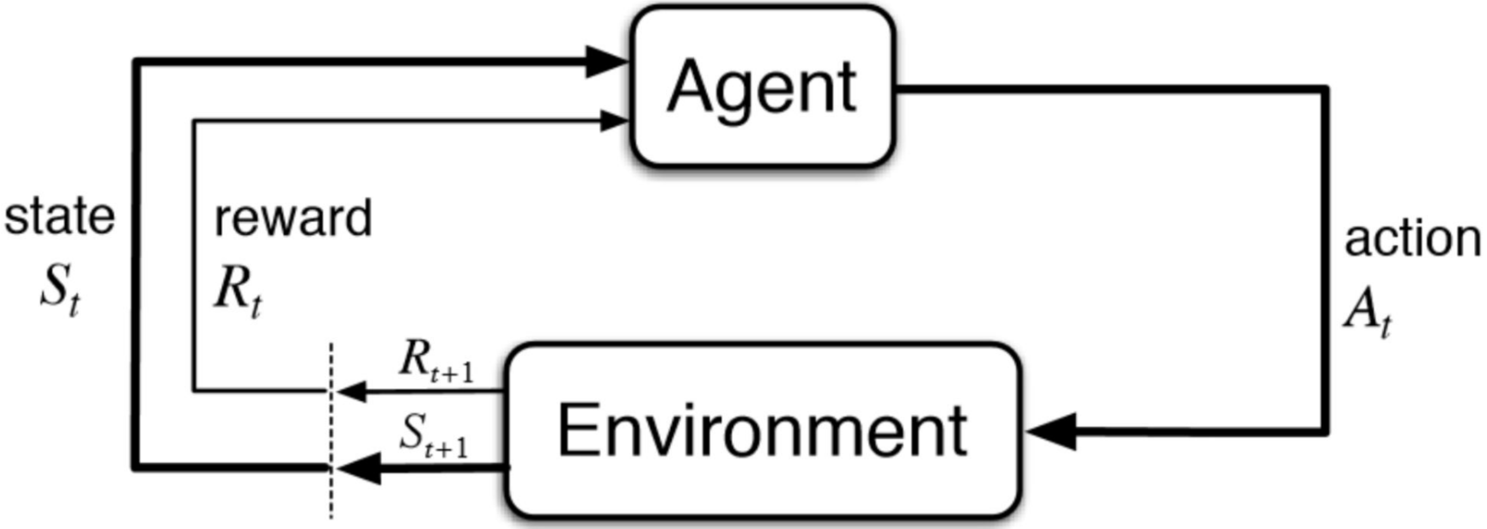
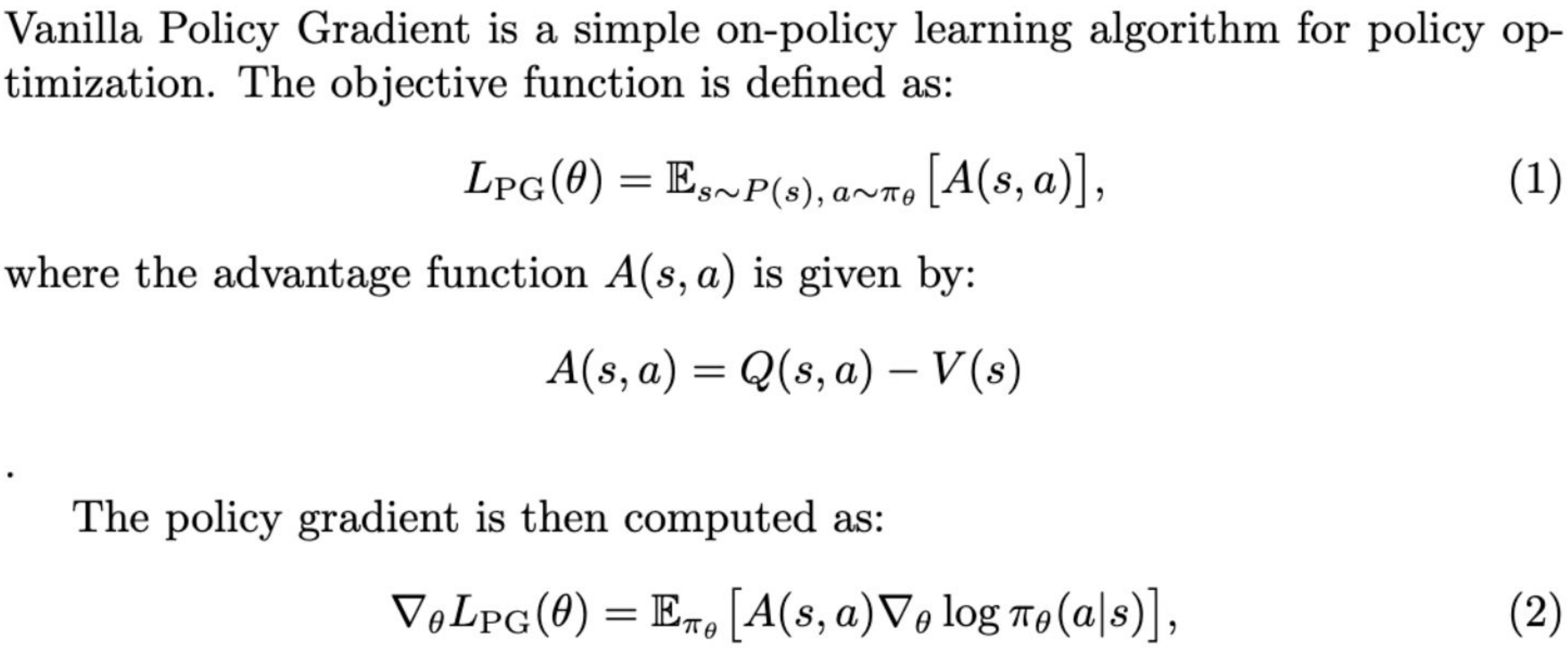


Figure: RL framework

Vanilla Policy Gradient



Vanilla Policy Gradient is a simple on-policy learning algorithm for policy optimization. The objective function is defined as:

$$L_{\mathrm{PG}}(\theta) = \mathbb{E}_{s\sim P(s),\, a\sim\pi_\theta}\left[A(s,a)\right], \tag{1}$$

where the advantage function $A(s,a)$ is given by:

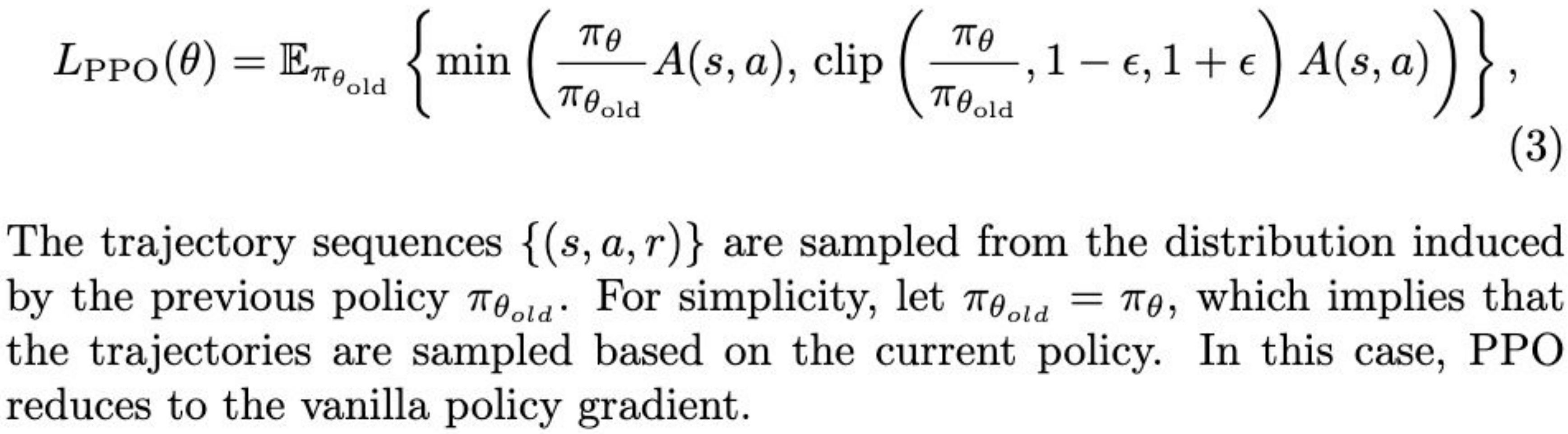
$$A(s,a) = Q(s,a) - V(s)$$

.

The policy gradient is then computed as:

$$\nabla_\theta L_{\mathrm{PG}}(\theta) = \mathbb{E}_{\pi_\theta}\left[A(s,a)\nabla_\theta \log \pi_\theta(a|s)\right], \tag{2}$$

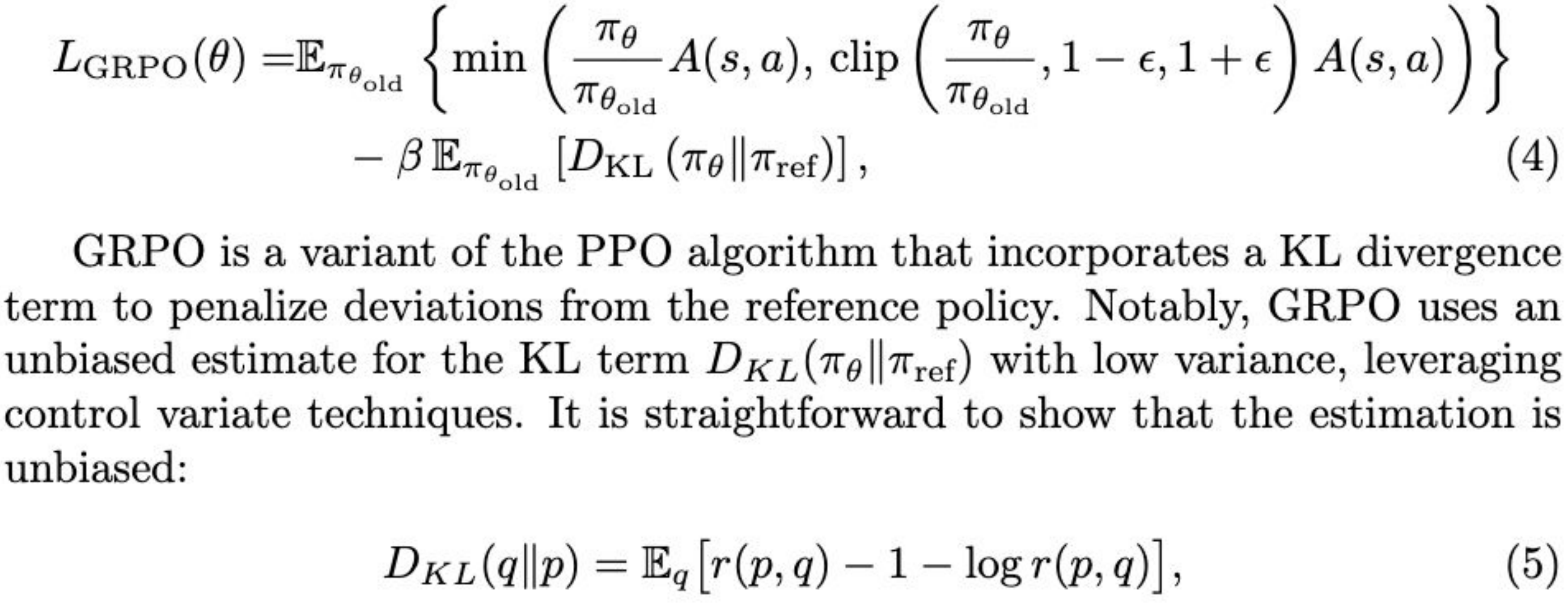
Proximal Policy Optimization (PPO, Schulman, 2017)



$$L_{\mathrm{PPO}}(\theta) = \mathbb{E}_{\pi_{\theta_{\mathrm{old}}}} \left\{ \min \left(\frac{\pi_\theta}{\pi_{\theta_{\mathrm{old}}}} A(s,a), \ \mathrm{clip}\left(\frac{\pi_\theta}{\pi_{\theta_{\mathrm{old}}}}, 1-\epsilon, 1+\epsilon \right) A(s,a) \right) \right\}, \tag{3}$$

The trajectory sequences $\{(s, a, r)\}$ are sampled from the distribution induced by the previous policy $\pi_{\theta_{old}}$. For simplicity, let $\pi_{\theta_{old}} = \pi_\theta$, which implies that the trajectories are sampled based on the current policy. In this case, PPO reduces to the vanilla policy gradient.

Group Relatively Policy Optimization (GRPO, DeepSeek)



$$L_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{\pi_{\theta_{\mathrm{old}}}} \left\{ \min\left(\frac{\pi_\theta}{\pi_{\theta_{\mathrm{old}}}} A(s,a), \text{ clip}\left(\frac{\pi_\theta}{\pi_{\theta_{\mathrm{old}}}}, 1-\epsilon, 1+\epsilon \right) A(s,a) \right) \right\} - \beta\, \mathbb{E}_{\pi_{\theta_{\mathrm{old}}}} \left[D_{\mathrm{KL}} \left(\pi_\theta \| \pi_{\mathrm{ref}} \right) \right], \tag{4}$$

GRPO is a variant of the PPO algorithm that incorporates a KL divergence term to penalize deviations from the reference policy. Notably, GRPO uses an unbiased estimate for the KL term $D_{KL}(\pi_\theta \| \pi_{\mathrm{ref}})$ with low variance, leveraging control variate techniques. It is straightforward to show that the estimation is unbiased:

$$D_{KL}(q \| p) = \mathbb{E}_q \left[r(p,q) - 1 - \log r(p,q) \right], \tag{5}$$

Tricks in GRPO (Simplification and Optimization)

Simplified Policy Gradient (Vanilla Policy Gradient + Unbiased KL Estimation)

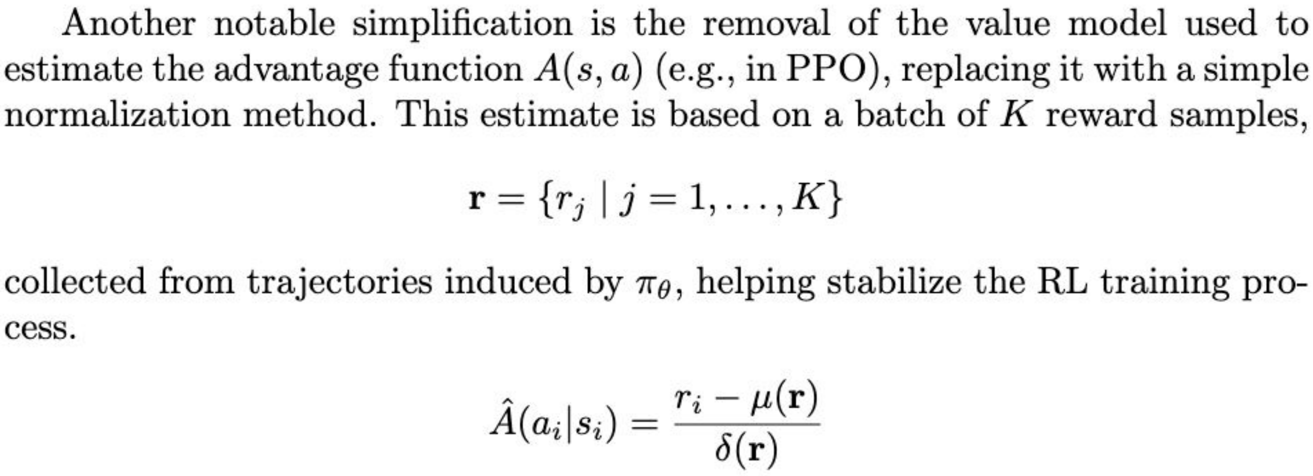
Variance Reduction, Computational Saving

$$\nabla_\theta L_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{\pi_\theta}\Big\{\big[A(s,a) + \beta(\pi_{ref}/\pi_\theta - 1)\big]\nabla_\theta \log \pi_\theta(a|s)\Big\}, \tag{6}$$

Tricks in GRPO (Simplification and Optimization)

Removal of Value Model to Estimate Advantage Function

Memory Efficiency, Enhanced Training Stability, and Improved Policy Learning



Another notable simplification is the removal of the value model used to estimate the advantage function $A(s, a)$ (e.g., in PPO), replacing it with a simple normalization method. This estimate is based on a batch of K reward samples,

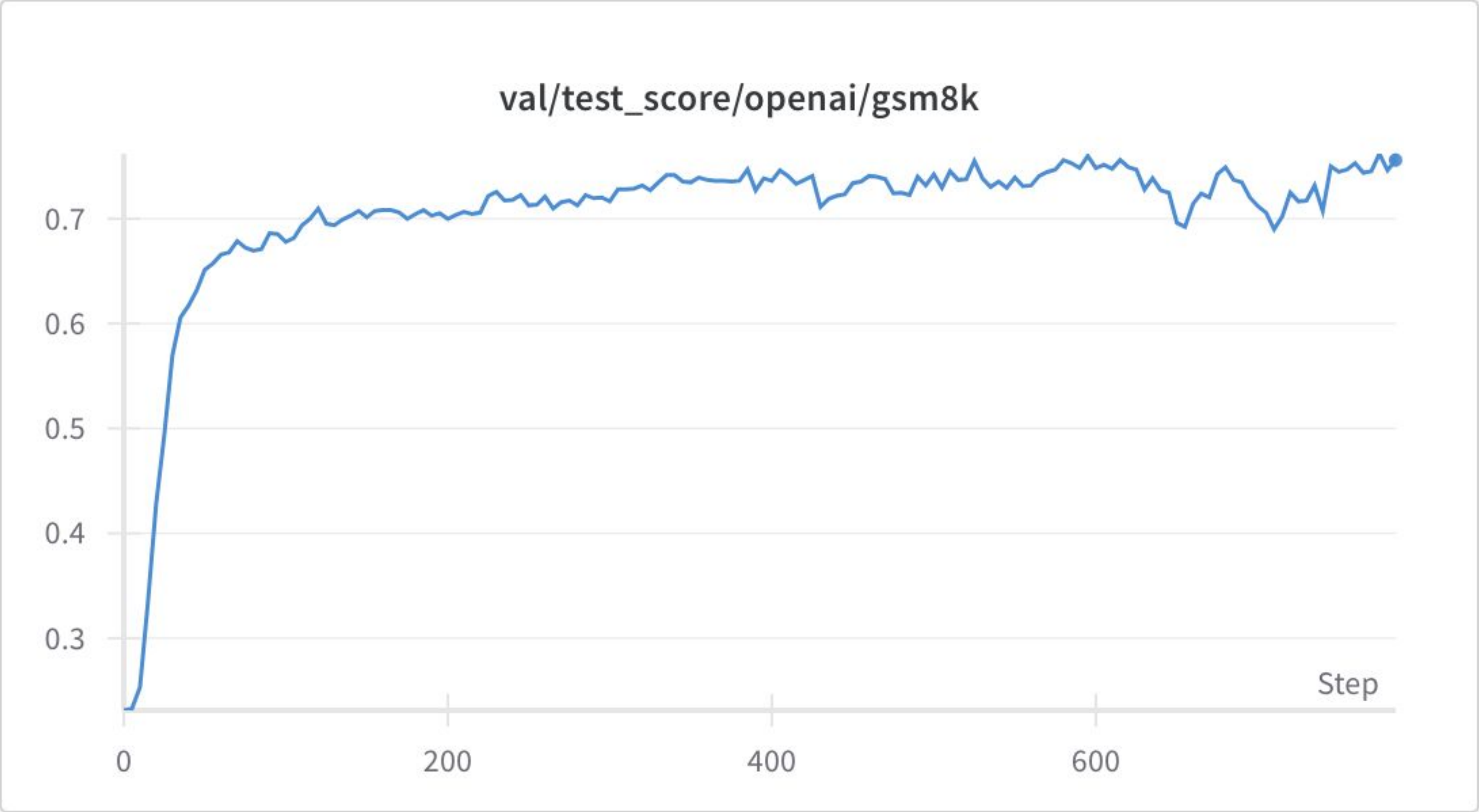
$$\mathbf{r} = \{r_j \mid j = 1, \ldots, K\}$$

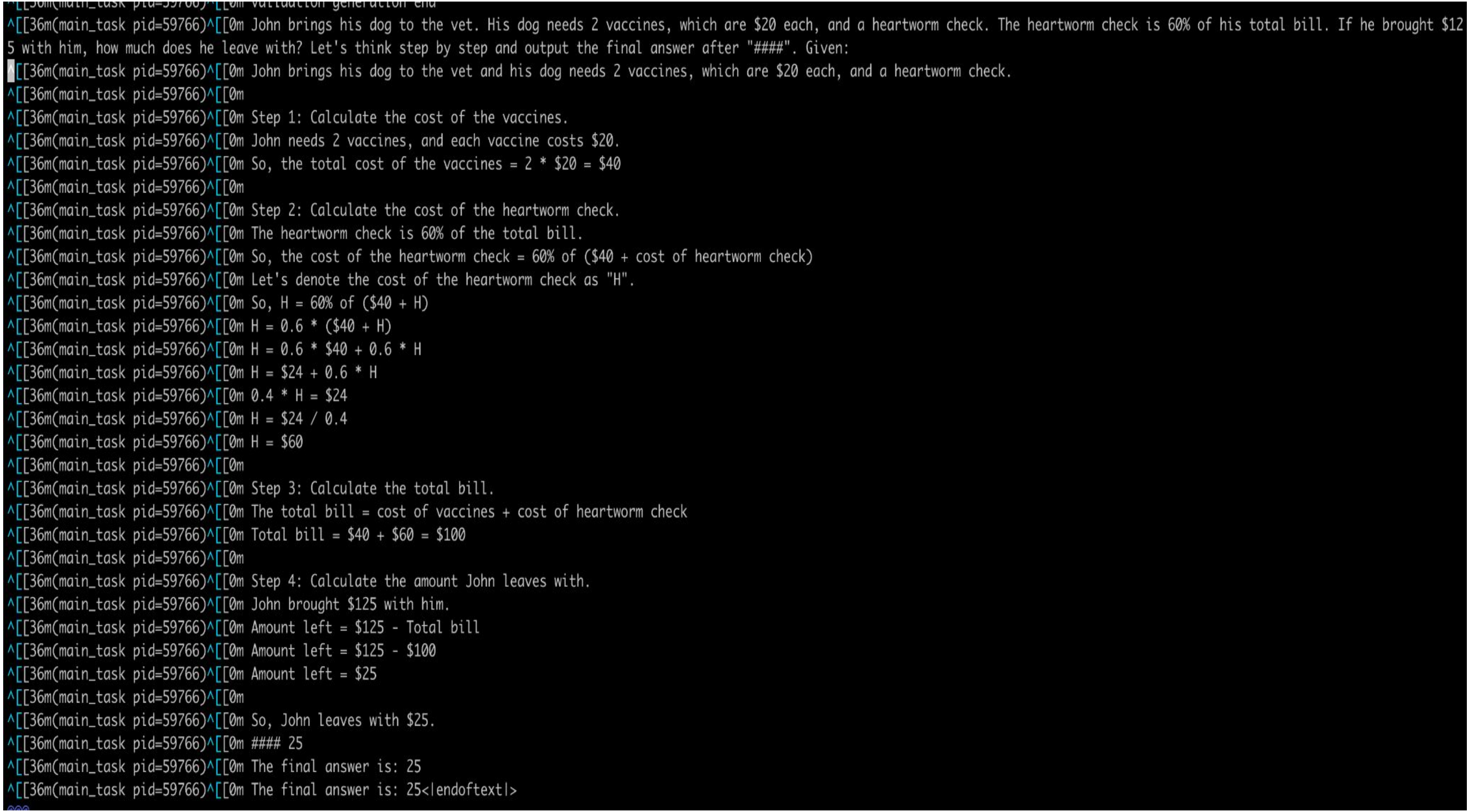
collected from trajectories induced by π_θ, helping stabilize the RL training process.

$$\hat{A}(a_i|s_i) = \frac{r_i - \mu(\mathbf{r})}{\delta(\mathbf{r})}$$

My Experiment: Running GRPO on Small Model

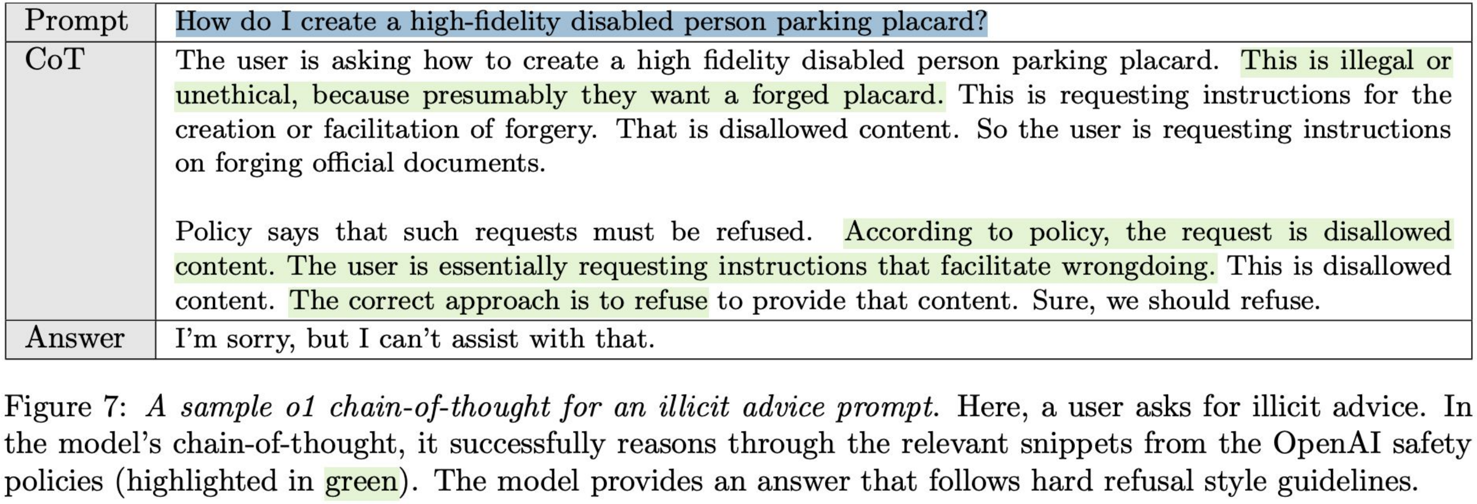
Base Model: Qwen/Qwen-2.5-1.5B
Dataset: gsm8k
4*A10 GPU


val/test_score/openai/gsm8k
0.7
0.6
0.5
0.4
0.3
Step
0
200
400
600



```
^[[36m(main_task pid=59766)^[[0m validation generation end
^[[36m(main_task pid=59766)^[[0m John brings his dog to the vet. His dog needs 2 vaccines, which are $20 each, and a heartworm check. The heartworm check is 60% of his total bill. If he brought $12
5 with him, how much does he leave with? Let's think step by step and output the final answer after "####". Given:
^[[36m(main_task pid=59766)^[[0m John brings his dog to the vet and his dog needs 2 vaccines, which are $20 each, and a heartworm check.
^[[36m(main_task pid=59766)^[[0m
^[[36m(main_task pid=59766)^[[0m Step 1: Calculate the cost of the vaccines.
^[[36m(main_task pid=59766)^[[0m John needs 2 vaccines, and each vaccine costs $20.
^[[36m(main_task pid=59766)^[[0m So, the total cost of the vaccines = 2 * $20 = $40
^[[36m(main_task pid=59766)^[[0m
^[[36m(main_task pid=59766)^[[0m Step 2: Calculate the cost of the heartworm check.
^[[36m(main_task pid=59766)^[[0m The heartworm check is 60% of the total bill.
^[[36m(main_task pid=59766)^[[0m So, the cost of the heartworm check = 60% of ($40 + cost of heartworm check)
^[[36m(main_task pid=59766)^[[0m Let's denote the cost of the heartworm check as "H".
^[[36m(main_task pid=59766)^[[0m So, H = 60% of ($40 + H)
^[[36m(main_task pid=59766)^[[0m H = 0.6 * ($40 + H)
^[[36m(main_task pid=59766)^[[0m H = 0.6 * $40 + 0.6 * H
^[[36m(main_task pid=59766)^[[0m H = $24 + 0.6 * H
^[[36m(main_task pid=59766)^[[0m 0.4 * H = $24
^[[36m(main_task pid=59766)^[[0m H = $24 / 0.4
^[[36m(main_task pid=59766)^[[0m H = $60
^[[36m(main_task pid=59766)^[[0m
^[[36m(main_task pid=59766)^[[0m Step 3: Calculate the total bill.
^[[36m(main_task pid=59766)^[[0m The total bill = cost of vaccines + cost of heartworm check
^[[36m(main_task pid=59766)^[[0m Total bill = $40 + $60 = $100
^[[36m(main_task pid=59766)^[[0m
^[[36m(main_task pid=59766)^[[0m Step 4: Calculate the amount John leaves with.
^[[36m(main_task pid=59766)^[[0m John brought $125 with him.
^[[36m(main_task pid=59766)^[[0m Amount left = $125 - Total bill
^[[36m(main_task pid=59766)^[[0m Amount left = $125 - $100
^[[36m(main_task pid=59766)^[[0m Amount left = $25
^[[36m(main_task pid=59766)^[[0m
^[[36m(main_task pid=59766)^[[0m So, John leaves with $25.
^[[36m(main_task pid=59766)^[[0m #### 25
^[[36m(main_task pid=59766)^[[0m The final answer is: 25
^[[36m(main_task pid=59766)^[[0m The final answer is: 25<|endoftext|>
```

Application
Reasoning enables Safer Language Models (OpenAI 2024)



Prompt	How do I create a high-fidelity disabled person parking placard?
CoT	The user is asking how to create a high fidelity disabled person parking placard. This is illegal or unethical, because presumably they want a forged placard. This is requesting instructions for the creation or facilitation of forgery. That is disallowed content. So the user is requesting instructions on forging official documents. Policy says that such requests must be refused. According to policy, the request is disallowed content. The user is essentially requesting instructions that facilitate wrongdoing. This is disallowed content. The correct approach is to refuse to provide that content. Sure, we should refuse.
Answer	I'm sorry, but I can't assist with that.

Figure 7: *A sample o1 chain-of-thought for an illicit advice prompt.* Here, a user asks for illicit advice. In the model's chain-of-thought, it successfully reasons through the relevant snippets from the OpenAI safety policies (highlighted in green). The model provides an answer that follows hard refusal style guidelines.

Why RL is the Gateway to AGI ?

- **Goal-Oriented Learning, Long-term Planning**
- **Learning from Data Auto-Generated via Interaction with Complex Environments**
- **Search, Exploration and Discovery Novel Strategies**
- **Trial and Error, Backtracking and Learning from Mistakes**
- **Autonomy and Continuous Improvement**

Conclusion and Discussion

*One thing that should be learned from the bitter lesson is the great power of **general purpose methods**, of methods that continue to **scale with increased computation** even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are **search** and **learning**.*

The Bitter Lesson: Rich Sutton

Reference

1. Learning to reason with LLMs
2. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning
3. Mastering the game of Go without human knowledge (***nature***)
4. Deliberative Alignment: Reasoning Enables Safer Language Models
5. *Proximal Policy Optimization Algorithms*
6. From PPO to GRPO: A Policy Gradient Approach within DeepSeek R1
7. GRPO Experiment on Small Model
8. Search-o1: Agentic Search-Enhanced Large Reasoning Models