DeepSeek-R1:

Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

Hantian (Welk) Lin

AI struggles with general reasoning

Most modern AI are good at prediction, but struggle with multi-step thinking, long-chain logic, and consistency.

Reasoning requires something different:

- Planning multiple steps
- Keeping track of intermediate results
- Checking correctness

Reasoning is a cognitive capability, not pure knowledge, so increasing data alone does not reliably improve reasoning performance.

Proposed solutions and their limitations

Scale up the size of LLMs

- Very expensive on computation power and data size.
- Not efficient or targeted

Use chain-of-thought prompting (asking the model to explain its response step-by-step)

- Requires careful prompt design
- Relies on human-written examples -> expensive

Train specifically for multi-step reasoning

- Requires human annotation → expensive and slow
- Introduces human bias
- Forces models to mimic human thinking patterns -> non-human-like pathways might be better sometime in some cases

Solution that DeepSeek picks: reinforcement learning (RL)

Reinforcement learning is a training method where a model learns by interacting with a task and receiving rewards based on its outputs, instead of learning from labeled examples:

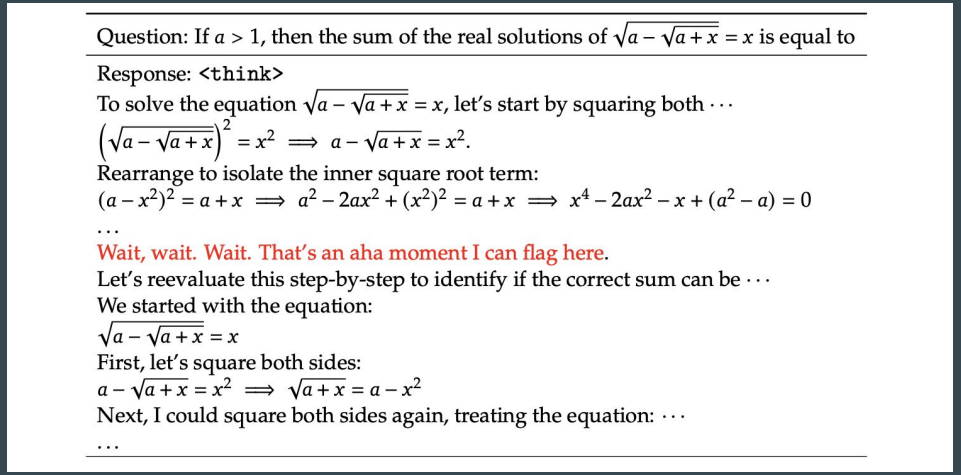
1. Model is given a problem
2. It generates an answer
3. A reward function evaluates the answer
    a. Correct → high reward
    b. Incorrect → low reward
4. Model updates itself to favor better outputs

The model is optimized to maximize rewards.

Supervised learning vs Reinforcement learning (RL)

Supervised learning → learns from examples

Reinforcement learning → learns from feedback



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:

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

$\cdots$

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: $\cdots$

$\cdots$

In the paper, the researchers mentioned "Although we do not explicitly teach the model how to reason, it successfully learns improved reasoning strategies through reinforcement learning".

"Sophisticated reasoning behaviors, such as self-verification and reflection, appeared to emerge organically during the reinforcement learning process".

How DeepSeek used RL

“The reward signal is solely based on the correctness of final predictions against ground-truth answers, without imposing constraints on the reasoning process itself”.

- Basically, the researchers don’t care how the model got the correct answer, as long as the answer was correct.

“Notably, we bypass the conventional supervised fine-tuning (SFT) phase before RL training. This design choice stems from our hypothesis that human-defined reasoning patterns may limit model exploration, whereas unrestricted RL training can better incentivize the emergence of novel reasoning capabilities in LLMs”.

- This new pipeline removed human constraints and allowed the model to develop the non-human-like pathways mentioned before.

RL implementation details

Proximal Policy Optimization (PPO): rewards + value model

“The training objective of the value model is to predict the expected cumulative reward from the current position onward, based on the tokens generated from the beginning up to the current position.”

Group Relative Policy Optimization (GRPO): rewards + compare all outputs

For each question q, GRPO samples a group of outputs $\{o_1, o_2, \cdots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_θ by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \quad (1)$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (2)$$

where π_{ref} is a reference policy, ε and β are hyper-parameters, and A_i is the advantage, computed using a group of rewards $\{r_1, r_2, \ldots, r_G\}$ corresponding to the outputs within each group:

$$A_i = \frac{r_i - mean(\{r_1, r_2, \cdots, r_G\})}{std(\{r_1, r_2, \cdots, r_G\})}. \quad (3)$$

Rule-based rewards in DeepSeek

DeepSeek uses two types of rule-based rewards, both equally weighted.

- Accuracy Reward: Measures whether the final answer is correct
- Format Reward: Encourages structured outputs. Requires reasoning in specific format.

“The self-evolution of DeepSeek-R1-Zero underscores the power and beauty of RL: rather than explicitly teaching the model how to solve a problem, we simply provide it with the right incentives, and it autonomously develops advanced problem-solving strategies.”

Model-based rewards in DeepSeek

DeepSeek uses two types of model-based rewards.

- Helpful Reward Model: Takes a response and evaluate how preferable it is.
- Safety Reward Model : Takes a response and evaluates how safe it is.

Compared to rule-based rewards which have hard-coded correctness (tasks involving math, code, logic), model-based rewards evaluate responses more softly and subjectively for general tasks.

Yet, RL is not the answer to every problem

"Exclusive reliance on RL can lead to reward hacking and suboptimal behavior in ill-posed tasks, while depending solely on SFT may prevent the model from optimizing its reasoning capabilities through exploration."

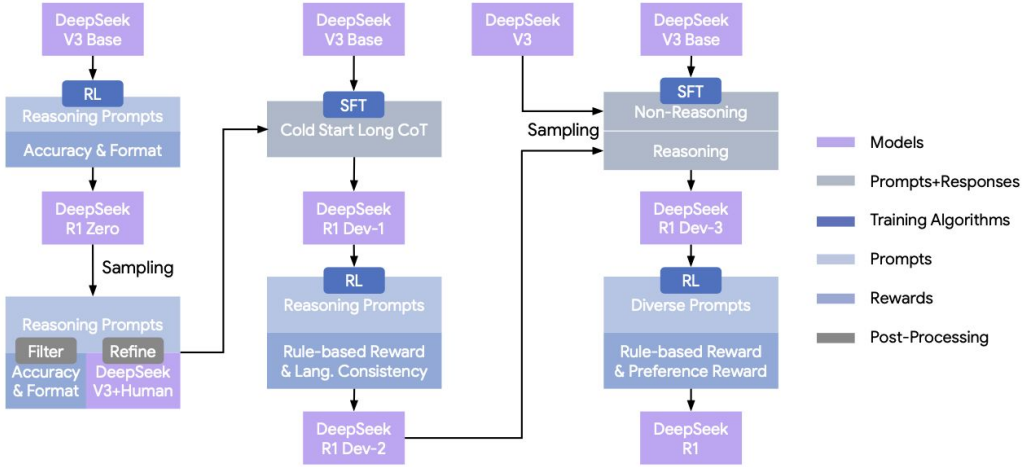


Figure 2 | The multi-stage pipeline of DeepSeek-R1. A detailed background on DeepSeek-V3 Base and DeepSeek-V3 is provided in Supplementary A.1. The models DeepSeek-R1 Dev1, Dev2, and Dev3 represent intermediate checkpoints within this pipeline.

What if it adds SFT first and then RL?

Can you now tell why DeepSeek prefers GRPO than PPO?

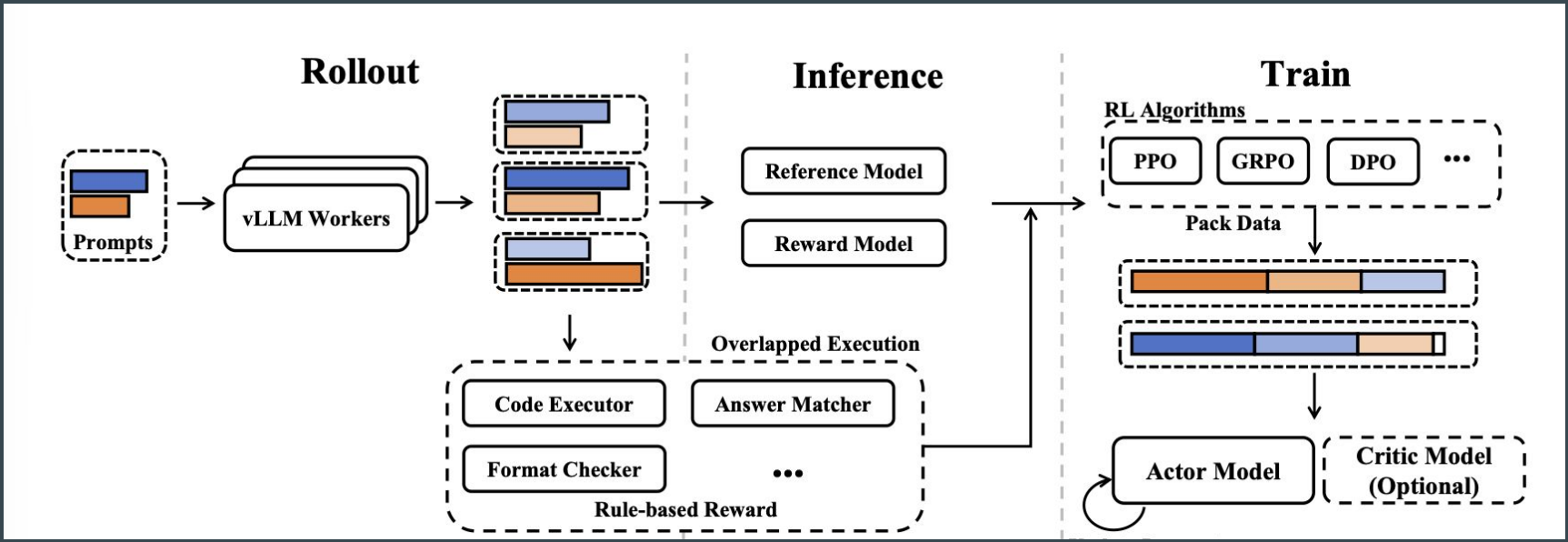
GRPO vs PPO

“This is inherently difficult, especially when only the final outcome reward is available. The challenge becomes even more pronounced when training long chain-of-thought reasoning models.”

“As the output length increases, the model is more likely to engage in behaviors such as reflection and revision during generation, meaning that the content initially generated may later be revised or contradicted, which makes it even less feasible to predict the final reward based on a partial response.”

It is hard to score a process.

Putting everything together



Some limitations of RL in DeepSeek

Long evaluation time

- Not great for software engineering task

Very sensitive to prompt

- Relies heavily on prompt design. Few-shot prompting can even degrade performance.

Reward signals are hard to construct for complex tasks

- Poor reward design can lead to reward hacking.

Some limitations of RL in DeepSeek

Small models (7B, 16B):

- Fail to improve with RL
- Struggle with long reasoning
- Show repetition

Large models (32B, 230B, 671B):

- Successfully benefit from RL
- Enable long chain-of-thought reasoning

RL effectiveness depends on model size

References

DeepSeek-AI. (2025). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948. https://arxiv.org/abs/2501.12948