

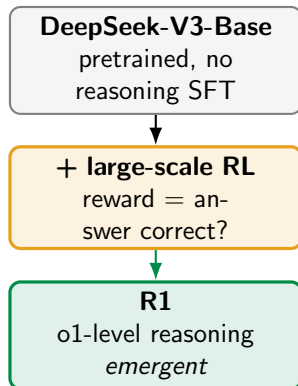
An open model reaches o1-level – and nobody taught it to reason

DeepSeek-R1 matches OpenAI's **o1** on competition math (AIME 79.8 vs 79.2) and coding (Codeforces 96th percentile) – open weights, published recipe.

The striking part is *how* it got there. The reasoning was **not demonstrated in training data**. Starting from a base model and a simple “is the answer correct?” reward, the model **taught itself** to think longer, check its work, and backtrack.

Advanced reasoning **emerged** from reinforcement learning – no human reasoning traces required.

Today: *how pure RL incentivizes reasoning, the full R1 pipeline, and why distilling R1 beats running RL on a small model.*



Outline

The setup: base model, correctness reward, GRPO

R1-Zero: reasoning from pure RL

The full R1 pipeline

Distillation into small models

Results and caveats

Recap

The setup

A pretrained base, a reward that only checks the answer, and one RL algorithm.

The starting point and the target

Start from DeepSeek-V3-Base – a strong pretrained model, but *not* fine-tuned to reason.

Target: strong reasoning on *verifiable* tasks – math, code, STEM – where an answer can be checked automatically.

The bet: do **not** start with supervised reasoning traces. The authors hypothesize that imitating human-written chains *caps* the model at human-like reasoning and blocks it from discovering its own.

Instead of teaching *how* to reason (SFT on chains), give the model a goal (*get the answer right*) and let RL find the reasoning that achieves it.

First experiment – **R1-Zero** – is the extreme version: **pure RL, zero SFT**.

The reward is rule-based, not a neural model

For verifiable tasks, the reward is computed by **rules**, not a learned network:

$$R_{\text{rule}} = R_{\text{accuracy}} + R_{\text{format}}$$

- ▶ **Accuracy:** is the final answer correct? Math answers go in a box and are checked; code is run against test cases.
- ▶ **Format:** did the model wrap its thinking in `<think>...</think>` then give `<answer>...</answer>`? A tiny template – the *only* structural constraint imposed.

No neural reward model for reasoning. A learned RM invites *reward hacking* at scale – the policy games the scorer instead of solving the problem. A rule you cannot fool is safer.

The RL algorithm: GRPO (recall from [LLM-1])

R1 optimizes the reward with **GRPO** – the critic-free algorithm from the GRPO deck. We just *recall* it here:

- ▶ Sample a **group** of G answers to each question.
- ▶ Score each with the rule-based reward.
- ▶ Advantage = how far each answer's reward is from the **group mean**, in std units:

$$\hat{A}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

- ▶ Push up better-than-average answers, push down the rest. **No value network.**

The group mean *is* the baseline – so a reward that only fires on correct answers still gives a usable learning signal: "better than the other $G - 1$ tries."

Config: $G = 16$ samples/question, lr 3×10^{-6} , KL coef 0.001, up to 32k–64k tokens/answer, ~ 10.4 k steps.

R1-Zero: pure RL, no SFT

The paper's most striking result – watch reasoning appear on its own.

Why would that even work? Selection, not teaching

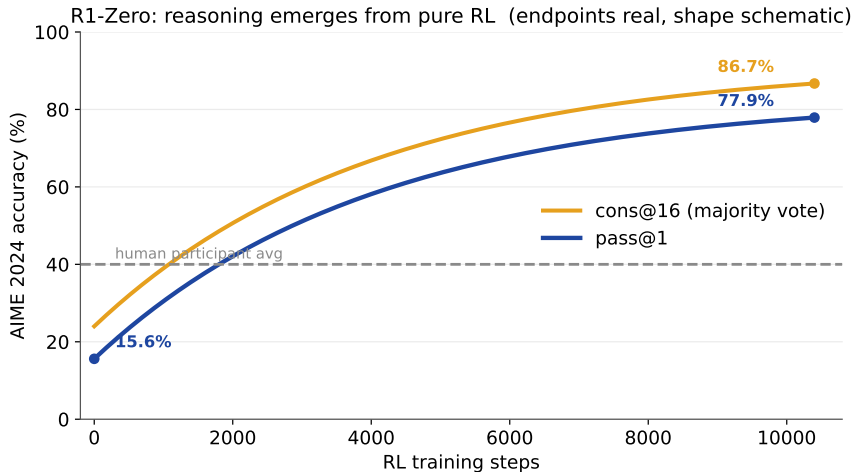
Rewarding only the final answer sounds too weak to *produce* reasoning. The trick is that RL does not **teach** – it **selects**.

- ▶ Each step, the model samples **many** attempts at a question – some short, some that happen to check their work or try another route.
- ▶ The attempts that *happen* to reason land the correct answer **more often** $\Rightarrow$ higher reward $\Rightarrow$ get reinforced.
- ▶ Nobody named “verify” or “backtrack.” They survive because they *pay off* – so they grow more frequent over training.

Like **evolution**: no designer drew the eye – organisms that saw better survived. Here, rollouts that reason better get rewarded. Correctness is the *selection pressure*; RL is the amplifier.

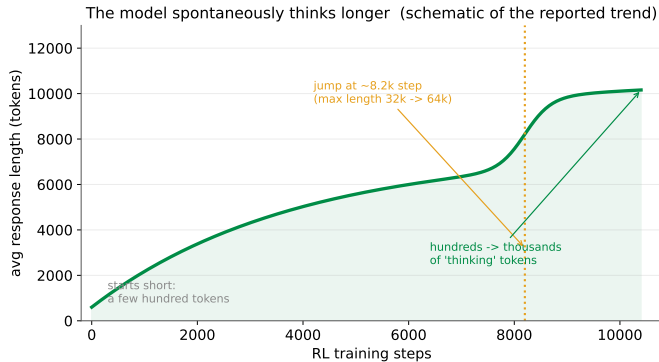
So expect reasoning to *appear* and *deepen* as RL runs – exactly what the next slides show.

Accuracy climbs from base toward o1 over RL steps



Pure RL on V3-Base lifts AIME pass@1 from **15.6%** to **77.9%**; majority vote over 16 samples reaches **86.7%**, past the human-participant average – **with no supervised reasoning data at all.**

The model learns to think longer – on its own



Nobody set a target length. As RL proceeds, the model *chooses* to generate more – from a few hundred to thousands of tokens per answer.

Extra tokens are spent **exploring alternatives, re-deriving, and verifying** – the model buys accuracy with test-time thinking.

Curve schematic; the trend and the ~8.2k-step jump are from the paper's Figure 1.

Predict-first: what behaviors emerge from just a correctness reward?

The reward says *nothing* about *how* to reason – only whether the final answer is right.
So what does the model develop?

Predict-first: what behaviors emerge from just a correctness reward?

The reward says *nothing* about *how* to reason – only whether the final answer is right. So what does the model develop?

It spontaneously learns to **self-verify** ("let me check"), **explore alternative approaches**, and **backtrack** when a path fails – behaviors never demonstrated in any training example.

The "aha moment": mid-training, an intermediate R1-Zero interrupts its own derivation – *"Wait, wait. Wait. That's an aha moment I can flag here. Let's reevaluate this step-by-step..."* – and restarts with a better method. Reflection, learned from reward alone.

But R1-Zero is rough to read

Pure RL bought reasoning power at the cost of **presentation**:

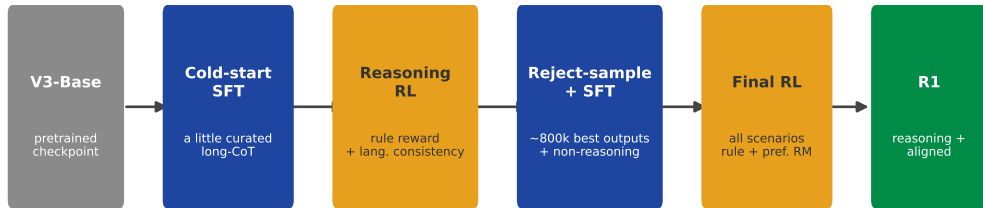
- ▶ **Poor readability:** chains are optimized to reach the answer, not to be read – messy, unformatted, hard to follow.
- ▶ **Language mixing:** a single chain-of-thought can switch between English and Chinese mid-thought (the base model is bilingual, and nothing penalized the mixing).
- ▶ **Narrow:** rule rewards only cover verifiable domains – weak at general writing and open-domain Q&A.

The capability is there; the *package* is not. That gap motivates the full **R1** pipeline – keep the reasoning, fix the delivery.

The full R1 pipeline

Four stages that keep R1-Zero's reasoning but make it readable and general.

Four stages: SFT and RL, alternating



SFT stages (blue) teach readable CoT; RL stages (orange) reward correctness, then broad alignment

A small **cold-start SFT** makes chains readable; **reasoning RL** re-grows the skill (now with a language-consistency reward); **reject-sample + SFT** folds the RL model's own best outputs plus non-reasoning data back in; a **final RL** stage aligns all scenarios.

What each stage does

1. **Cold-start SFT.** Fine-tune V3-Base on a *few thousand* curated long-CoT examples in a clean, conversational format. Just enough to fix readability before RL – *not* enough to teach reasoning.
2. **Reasoning-oriented RL.** GRPO with rule rewards (as R1-Zero), plus a **language-consistency** reward = fraction of the CoT in the target language. Slightly lowers scores but removes the mixing.
3. **Rejection sampling + SFT.** Sample the RL model, *keep the good answers* ($\sim 600k$ reasoning), add $\sim 200k$ non-reasoning (writing, QA, code) $\rightarrow \sim 800k$ examples, and SFT on them.
4. **Final RL for all scenarios.** A last RL pass: rule rewards for reasoning *and* a preference reward model for helpfulness/harmlessness.

The two ideas: **SFT for style**, **RL for capability** – and **bootstrap** off the model's own best generations.

Distillation

Move R1's reasoning into small dense models – with plain SFT.

Distill R1's reasoning into small dense models

Generate $\sim$ **800k** reasoning examples with R1, then **SFT** open base models on them – *no RL stage for the students*.

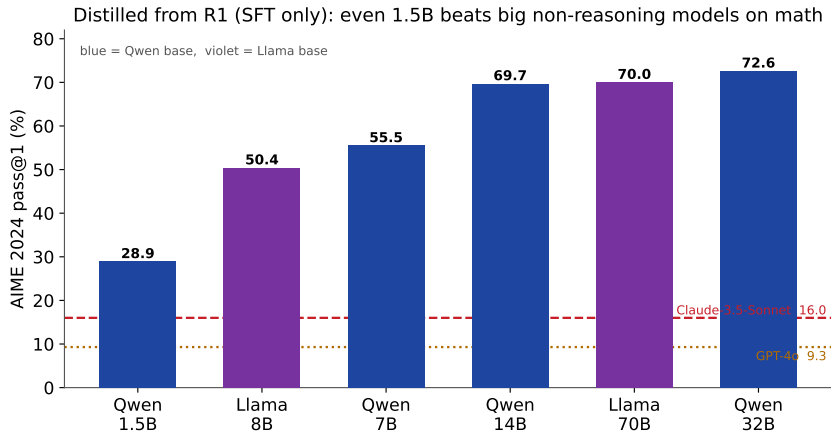
Students: **Qwen** and **Llama** bases, **1.5B** $\rightarrow$ **70B**.

A pure knowledge-transfer move: the small model learns to imitate R1's long chains-of-thought.

The teacher's reasoning is written down as text. Any model that can be fine-tuned on text can *inherit* it – cheaply, with standard SFT.

2–3 epochs of SFT on the 800k set; the same data used in R1's own stage-3 SFT.

Distilled small models beat big non-reasoning models on math



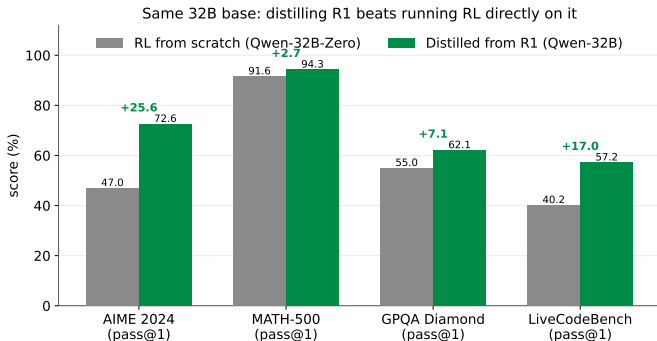
Even **Distill-Qwen-1.5B** (28.9 on AIME) clears GPT-4o (9.3) and Claude-3.5 (16.0); quality scales with student size up to **Distill-Qwen-32B** at 72.6.

Predict-first: distill R1, or run RL on the small model directly?

You have a 32B base. To make it reason, is it better to **distill from R1**, or to run the same **large-scale RL** on the 32B itself?

Predict-first: distill R1, or run RL on the small model directly?

You have a 32B base. To make it reason, is it better to **distill from R1**, or to run the same **large-scale RL** on the 32B itself?

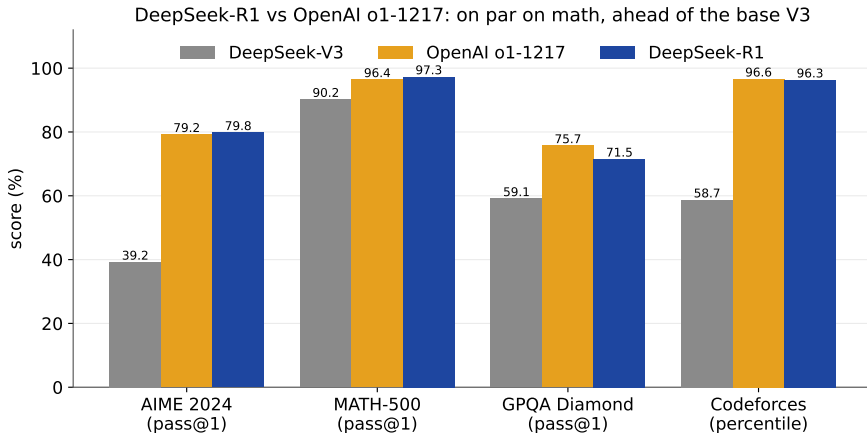


Distillation wins, and not by a little (AIME 72.6 vs 47.0). RL from scratch on a small model needs enormous compute and still trails imitating a stronger teacher's reasoning.

Results and caveats

Where R1 lands against o1 – and where it still falls short.

R1 vs o1-1217, head to head



On math, R1 is **on par** with o1-1217 (AIME 79.8 vs 79.2, MATH-500 97.3 vs 96.4) and far above the base V3. o1 still leads on GPQA; the two trade blows on code.

Caveats and limitations

- ▶ **Only where a verifier exists.** Pure RL needs a reliable reward. For open-ended tasks (writing) there is no rule to check, so a learned RM is used – and it can be **reward-hacked**.
- ▶ **Language mixing** persists for non-English/Chinese queries.
- ▶ **Prompt-sensitive:** few-shot prompting *hurts* R1 – zero-shot, just-state-the-problem works best.
- ▶ **No tools** (no calculator/search) and weaker structured output; overthinks easy questions.
- ▶ **One report, one lab.** An 86-page technical report; this deck abstracts its headline story, not every ablation.

Still: pure RL from a correctness reward produced frontier reasoning – capability that was *latent in the base model* and **drawn out**, not injected.

Recap

- ▶ **R1-Zero**: pure RL (GRPO) on V3-Base with a **rule-based** reward – reasoning **emerges**: longer thinking, self-verification, backtracking, the "aha moment." No SFT, no reasoning traces.
- ▶ R1-Zero is capable but **unreadable** and mixes languages.
- ▶ **R1 pipeline**: cold-start SFT → reasoning RL → reject-sample + SFT → final RL. *SFT for style, RL for capability.*
- ▶ **R1** $\approx$ **o1** on math/code; well ahead of the base V3.
- ▶ **Distillation** of R1's chains into 1.5B–70B models **beats** running RL on those small models directly.
- ▶ Reward = correctness; the RL **draws out** latent ability rather than injecting new skills.

Next: [LLM-8] Qwen3 – the same reasoning-RL family, with a single model that can *switch* between slow thinking and fast answering.