



Teaching a Language Model to Reason

Supervised fine tuning and Reinforcement Learning on
Qwen3-1.7B

Marcus Isenborg

Antti Huovila

David Hellström

Dennis Johansson

The Rise of Reasoning Models

Models that think step-by-step before answering

Sep 2024

OpenAI o1

First major reasoning model. Demonstrated that chain-of-thought reasoning dramatically improves problem solving on math and coding tasks.

Jan 2025

DeepSeek R1

Open-source breakthrough. Published the full algorithm behind training reasoning models, making the technique accessible to researchers worldwide.

2025+

Open Replication

Researchers begin replicating the DeepSeek approach on smaller models, proving RL-based reasoning can work without massive compute budgets.

Our Research Question

Can post-training improve a small (Qwen3 1.7B) base model's mathematical reasoning ability?



Approach 1: RLVR (R1-Zero)

Apply reinforcement learning directly on the base model using GRPO on verifiable math tasks. The model learns to reason through trial and error - no reasoning examples provided.



Approach 2: Supervised Fine Tuning

Supervised fine-tuning on reasoning trajectories distilled from a larger reasoning model. The model learns by imitating expert chain-of-thought outputs.

The Dataset: GSM8K¹



Grade School Math with 8,000+ problems with numerical answers

Example Problem

Janet's ducks lay 16 eggs per day. She eats 3 for breakfast every morning and bakes muffins for her friends every day with 4. She sells every remaining egg for \$2. How much does she make per day?

Answer: \$18

Why GSM8K?

- ✓ Verifiable answers, we can automatically check if the model is correct
- ✓ Simple enough for a small model to make progress on
- ✓ Clear numeric output makes reward design straightforward

¹ Cobbe, K., Kosaraju, V., Bavarian, M., et al. "Training Verifiers to Solve Math Word Problems." arXiv:2110.14168, 2021.

How RLVR Training Works

Reinforcement Learning from Verifiable Rewards – learning by doing

Pretraining

- Reading a textbook
- Memorizing patterns from data
- No feedback on performance
- Passive absorption of knowledge

VS

RL Training (Ours)

- **Practicing problems yourself**
- **Trying, failing, and learning why**
- **Direct reward signal: right or wrong**
- **No reward model needed**

Reward signal: Correct answer \rightarrow reward = 1 | Incorrect answer \rightarrow reward = 0.

In practice, the reward also includes a formatting component, but for simplicity we focus on the correct answer.

The GRPO Algorithm²

Group Relative Policy Optimization

1

Sample

For each problem, generate a group of responses from the model.



2

Score

Check each responses answer against the known correct answer (reward 0 or 1)



3

Compare

Calculate advantage: how much better/worse each response is vs the group average



4

Update

Increase probability of better-than-average answers, decrease worse ones

Supervised Fine-Tuning

Our second approach - learning from a teacher model's reasoning traces

Supervised Fine-Tuning (SFT)

How It Works

- A larger reasoning model generates step-by-step solutions for GSM8K problems ¹
- Qwen3-1.7B trained to reproduce these reasoning trajectories
- Model learns to imitate expert reasoning

Key Idea

- All model parameters are updated during training
- The model directly learns "how to think" from expert demonstrations
- Fast convergence but requires high-quality training data

¹ https://huggingface.co/datasets/multi-domain-reasoning/gsm8k_reasoning_nemotron-70B



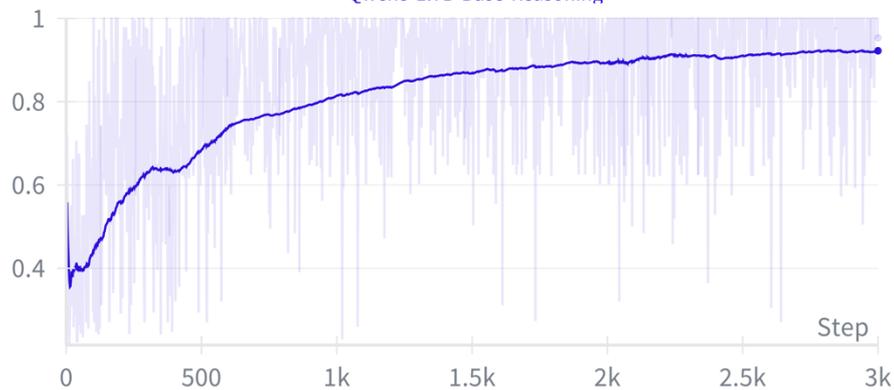
Training Progress

Observing how the model learn to reason

Training Curves

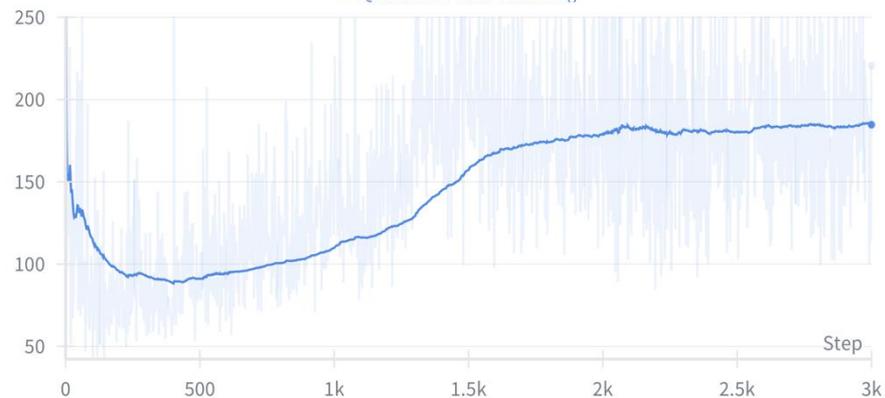
train/mean_reward

— Qwen3-1.7B-Base-Reasoning



train/mean_response_length

— Qwen3-1.7B-Base-Reasoning



The model organically learns to produce longer, more detailed reasoning chains as accuracy improves

Training Challenges



Dataset Saturation

Later in training, many problems were being solved 100% of the time across all samples in a group. When every answer is correct, the advantage is zero and the model cannot learn from those examples.

Diminishing Returns

As the easy problems got mastered, a large fraction of training batches had no useful gradient signal. The model plateaued because it had effectively "aced" the training set.

Memory constraints

Training on a single GPU forced us to compromise with the number of batches and lower number of responses per question.

Implication: The training dataset needs to be harder and more diverse to sustain learning



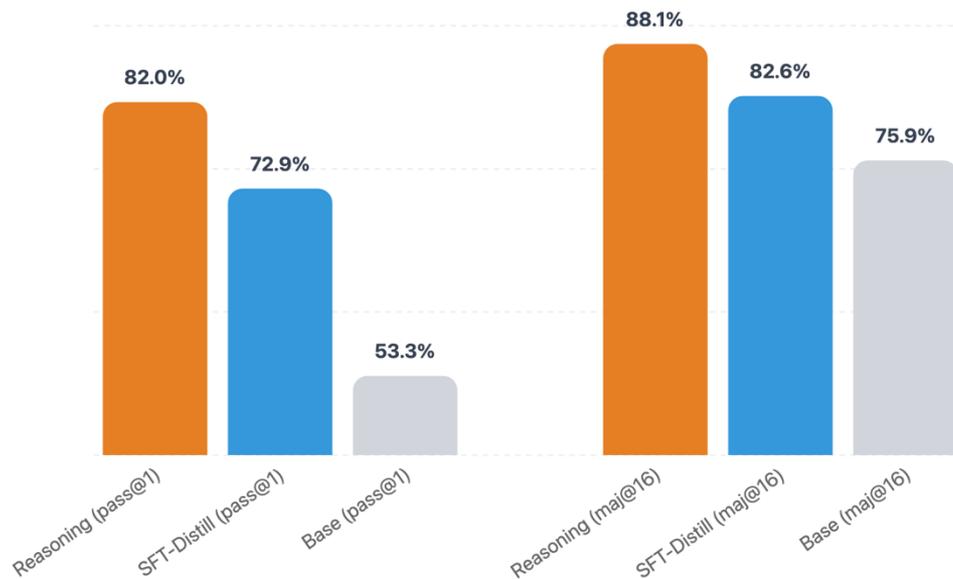
Results

How well did our model perform?

Performance Results

GSM8K

Grade school level math questions



Emergent Reasoning Traces

The model learned to "think out loud", showing its work before answering

Example: Model's Reasoning Trace

GSM8K Question:

At 8:00, 5000 people lined up at a polling station to cast their vote on election day. By midday $\frac{2}{5}$ of the people had voted and by 16:00 $\frac{2}{3}$ of the remaining people had voted. What's the number of those who had not voted by 16:00?

Assistant:

🌀 Thinking...

First, we determine the number of people who had voted by midday. At 8:00, 5000 people lined up. By midday, $\frac{2}{5}$ of them had voted:

$$\left[\frac{2}{5} \times 5000 = 2000 \right]$$

So, 2000 people had voted by midday. The number of people who had not voted by midday is:

$$\left[5000 - 2000 = 3000 \right]$$

By 16:00, $\frac{2}{3}$ of the remaining 3000 people had voted. The number of people who had voted by 16:00 is:

$$\left[\frac{2}{3} \times 3000 = 2000 \right]$$

So, 2000 people had voted by 16:00. The number of people who had not voted by 16:00 is:

$$\left[3000 - 2000 = 1000 \right]$$

Answer: 1000



Concerns & Discussion

Reward Hacking Example

Correct answer, broken reasoning - the model backtracks when it hits a non-integer

User: Drew has caught 16 fish. He must give his father twice as many as his mother. Drew has two brothers and one sister. Each brother receives 2 fish, his sister gets 1. If all 16 fish are distributed, how many fish will the father receive?

Assistant: <think>

First, let's determine how many fish the siblings receive.

- Each brother receives 2 fish, so the brothers receive $2 \times 2 = 4$ fish.
- The sister receives 1 fish.

[...] $3x = 11$

Since 11 is not divisible by 3, it seems there might be an error in the problem statement or interpretation. However, assuming the problem is correctly stated and we need to find the closest whole number distribution, we can proceed as follows:

$11 = 3 \times 3 + 2$ [...] $3 \times 3 = 9$ fish (3 to mother, 6 to father)

$2 \times 3 = 6 + 2 = 8.5 + 2.5 = 10.5 + 1.5 = 11.5$

</think>

Answer: 6

The model learned GSM8K only has integer answers. When $3x = 11$ yields no integer, it backtracks and forces a whole-number output producing incoherent math (" $8.5 + 2.5 = 10.5 + 1.5 = 11.5$ ") but landing on the correct answer 6 by chance.

Everything Is a Math Problem

The model was trained exclusively on math . It now sees everything as a math problem, even simple greetings

User

Hello!

Model

`<think>`

To solve this problem, it is known for a triangle that the sum of its internal angles...

1. First we know the measure of angle A: ...

...

5. Solving for x: ...

Therefore, the measure of angle C is 30 degrees.

`</think>`

`<answer>30</answer>`

Future Improvements



Diverse Training Data

Use multiple benchmarks beyond GSM8K, include harder math (MATH, AMC), code, and logic problems to prevent narrow specialization.

Curriculum Learning

Phase in harder problems as the model masters easier ones. Start with GSM8K, then introduce competition-level math to sustain learning signal.

Non-Integer Answers

Include problems with decimal, fractional, or symbolic answers to prevent the integer-rounding reward hack we observed.

Cleaner Base Model

Start from a model with verified training data, or test on a model known to not have seen reasoning traces in pretraining.

Key Takeaways



RL improved pass@1 from ~50% to ~80%; SFT achieved ~73% correct on pass@1



The model organically learns to produce longer, structured reasoning chains



Data contamination concerns make it hard to claim a pure "from scratch" result



Reward hacking is a real risk, simple reward functions can be exploited



Narrow training produces narrow models, ours hallucinates math everywhere



Future: combine SFT first, then RL on top for even stronger reasoning

Questions?

Thank you for listening!
