



DeepSeek-R1 and the Rise of Open Large Reasoning Models

Lewis Tunstall

Research Scientist at Hugging Face

I

Why train
large reasoning
models?

II

Can we train
fully open
reasoning models?

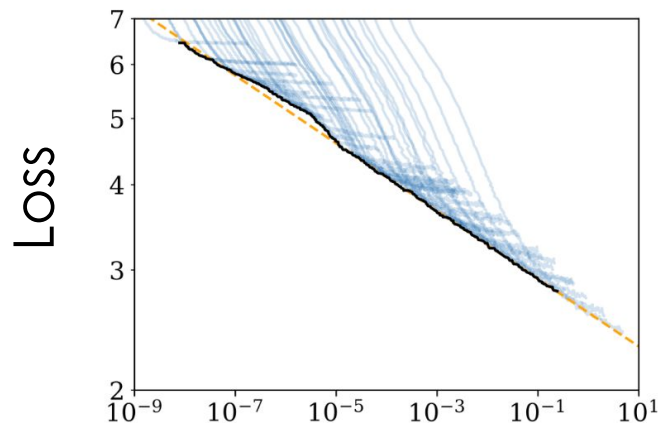
III

What **impact** has
DeepSeek-R1 had
on **open AI**?

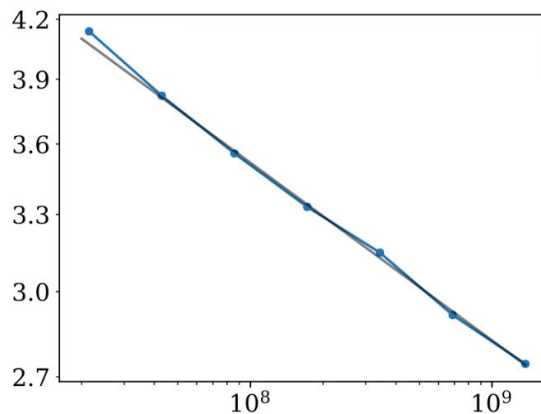
Why train large reasoning models?

scaling, code, tool-use

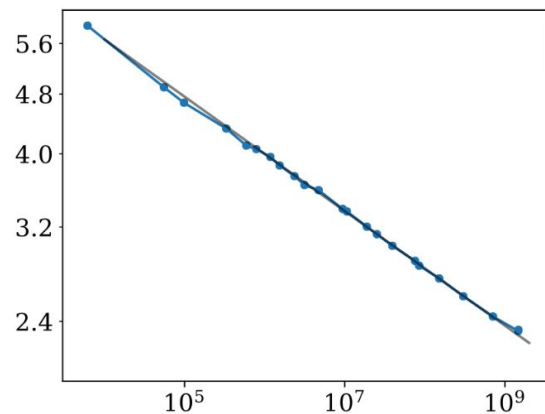
Scaling laws: **predictable** returns



Compute



Data

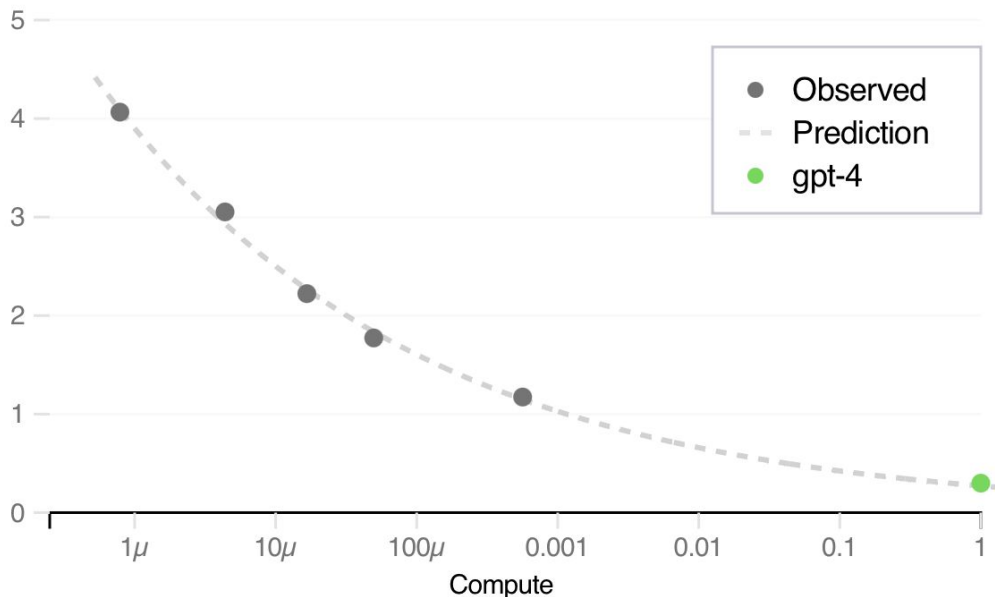


Model size

Scaling laws: **predictable** returns

Capability prediction on 23 coding problems

– Mean Log Pass Rate



<https://openai.com/index/gpt-4-research/>

Limits to scaling?

Constraints to scaling training runs by 2030

EPOCH AI



By 2030 it will be very likely *possible* to train models that exceed GPT-4 in scale to the same degree that GPT-4 exceeds GPT-2 in scale ...

<https://epoch.ai/blog/can-ai-scaling-continue-through-2030>

Limits to scaling?

Constraints to scaling training runs by 2030

EPOCH AI



By 2030 it will be very likely *possible* to train models that exceed GPT-4 in scale to the same degree that GPT-4 exceeds GPT-2 in scale ...

... whether AI developers will actually pursue this level of scaling depends on their willingness to **invest hundreds of billions of dollars**

<https://epoch.ai/blog/can-ai-scaling-continue-through-2030>

Limits to scaling?

NewScientist

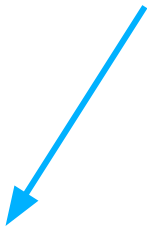
Is OpenAI hitting a wall with huge and expensive GPT-4.5 model?

Some researchers think OpenAI's giant and expensive latest model is a sign that tech companies cannot keep making progress by continually scaling up

By [Matthew Sparkes](#)

📅 28 February 2025

deprecated less than
2 months after release!



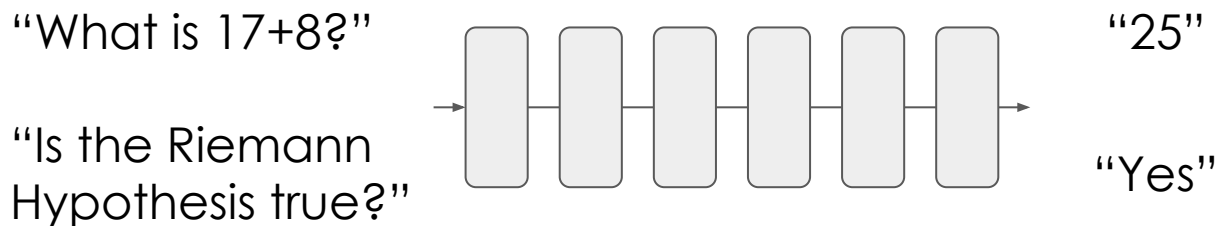
2025-04-14: GPT-4.5-preview

On April 14th, 2025, we notified developers that the `gpt-4.5-preview` model is deprecated and will be removed from the API in the coming months.

SHUTDOWN DATE	MODEL / SYSTEM	RECOMMENDED REPLACEMENT
2025-07-14	gpt-4.5-preview	gpt-4.1

<https://www.newscientist.com/article/2470327-is-openai-hitting-a-wall-with-huge-and-expensive-gpt-4-5-model/>

New frontier: **scaling test-time compute**



A transformer model uses **fixed amount of compute** per token irrespective of difficulty.

So how can we scale compute at **inference-time**?

New frontier: **scaling test-time compute**

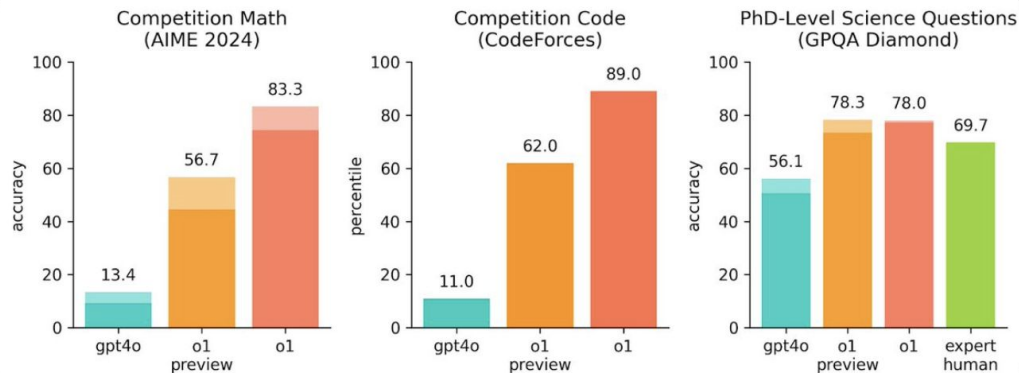


Noam Brown

@polynoamial



Today, I'm excited to share with you all the **fruit** of our effort at @OpenAI to create AI models capable of truly general reasoning: OpenAI's new o1 model series! (aka 🍓) Let me explain 🧵 1/



7:17 PM · Sep 12, 2024 2.5M Views

o1 - the first large reasoning model

New frontier: **scaling test-time compute**

GPT-4o

User

oyfjdnisdr rtqwainr acxz mynzbhbx -> Think step by step

Use the example above to decode:

oyekaijzdf aaptcg suaokybhai ouow aqht
mynznvaatzacdfoulxxz

Let's break this down step by step based on the example:

1. Example given:

- Input: oyfjdnisdr rtqwainr acxz mynzbhbx
- Output: **Think step by step**

OpenAI o1-preview

User

oyfjdnisdr rtqwainr acxz mynzbhbx -> Think step by step

Use the example above to decode:

oyekaijzdf aaptcg suaokybhai ouow aqht
mynznvaatzacdfoulxxz

Thought for 5 seconds ^

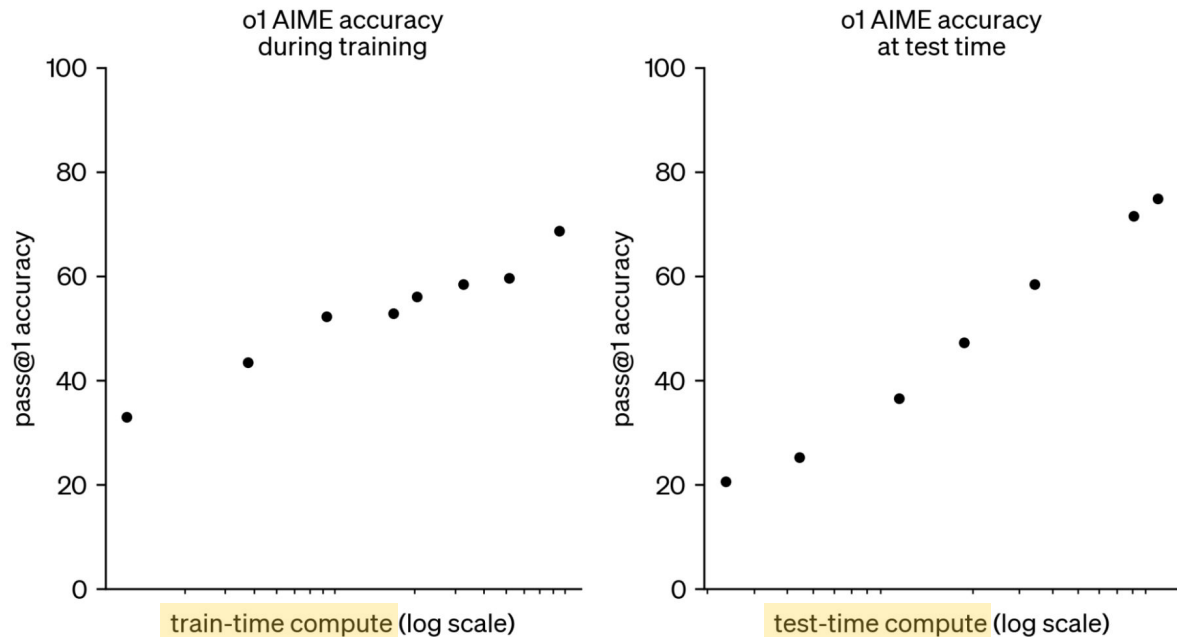
First, what is going on here?

We are given:

First, an example:

“think” first, then answer
(System 1 vs 2)

New frontier: **scaling test-time compute**

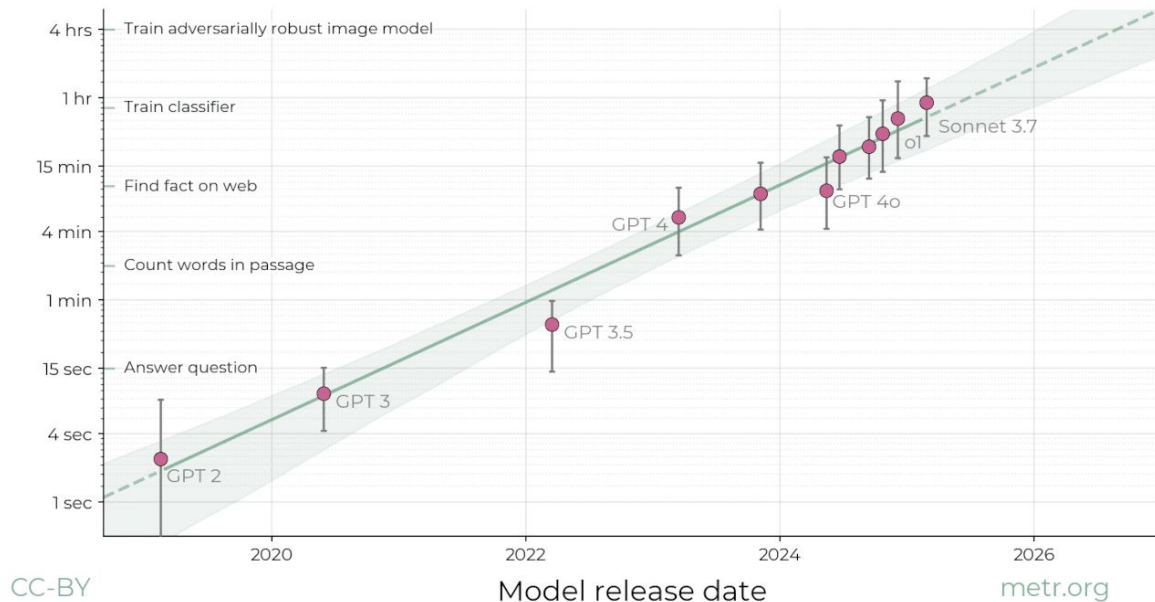


New frontier: **scaling test-time compute**

The length of tasks AI can do is doubling every 7 months



Task length (at 50% success rate)



<https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks/>

o1: what we knew

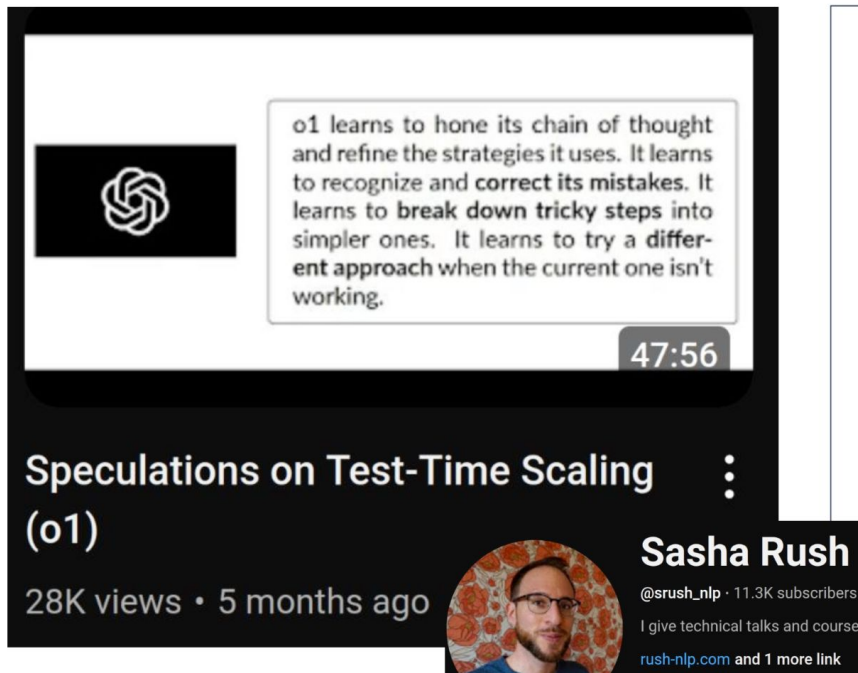
model learns “by itself”
search

“Think step by step”

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.

- OpenAI, 2024

o1: how did it actually work?



o1 learns to hone its chain of thought and refine the strategies it uses. It learns to recognize and correct its mistakes. It learns to break down tricky steps into simpler ones. It learns to try a different approach when the current one isn't working.

47:56

Speculations on Test-Time Scaling (o1)

28K views • 5 months ago

Sasha Rush
@srush_nlp · 11.3K subscribers
I give technical talks and courses
rush-nlp.com and 1 more link

The Suspects

- Guess + Check
- Process Rewards
- Search / AlphaZero
- Learning to Correct

<https://youtu.be/6PEJ96k1kiw?feature=shared>

DeepSeek-R1: first open-weights reasoner

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

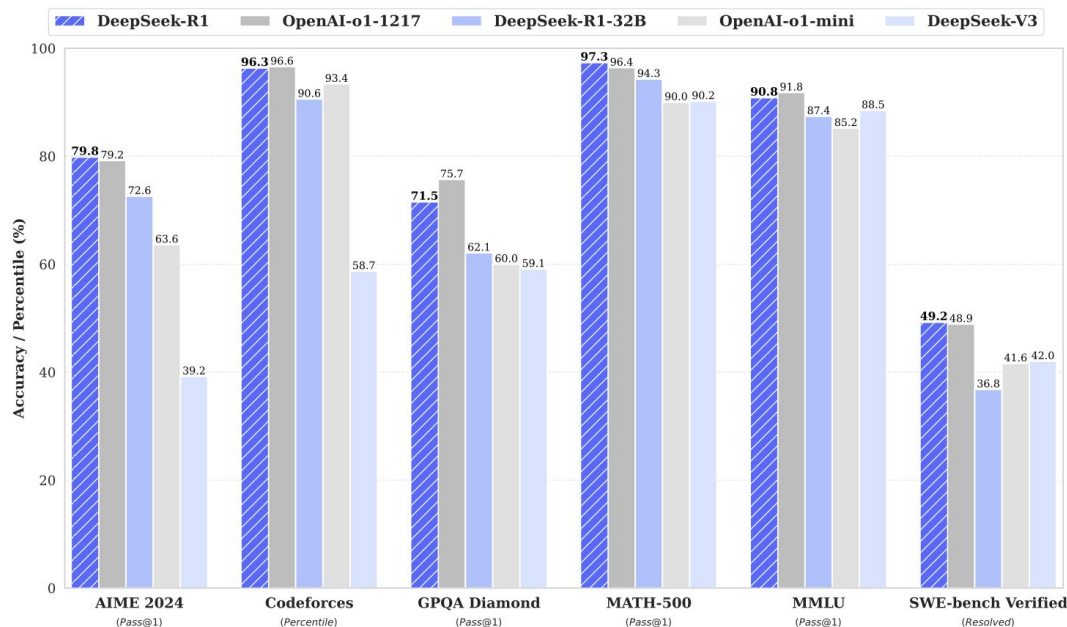


Figure 1 | Benchmark performance of DeepSeek-R1.

~4 months after o1

Models: **8**

Downloads: **9.1M**

Derivatives: **2.5k**

Downloads: **4.4M**

last 30 days

<https://arxiv.org/abs/2501.12948>

22 Jan 2025

[cs.CL]

DeepSeek-R1: similar to o1



Mark Chen ✓

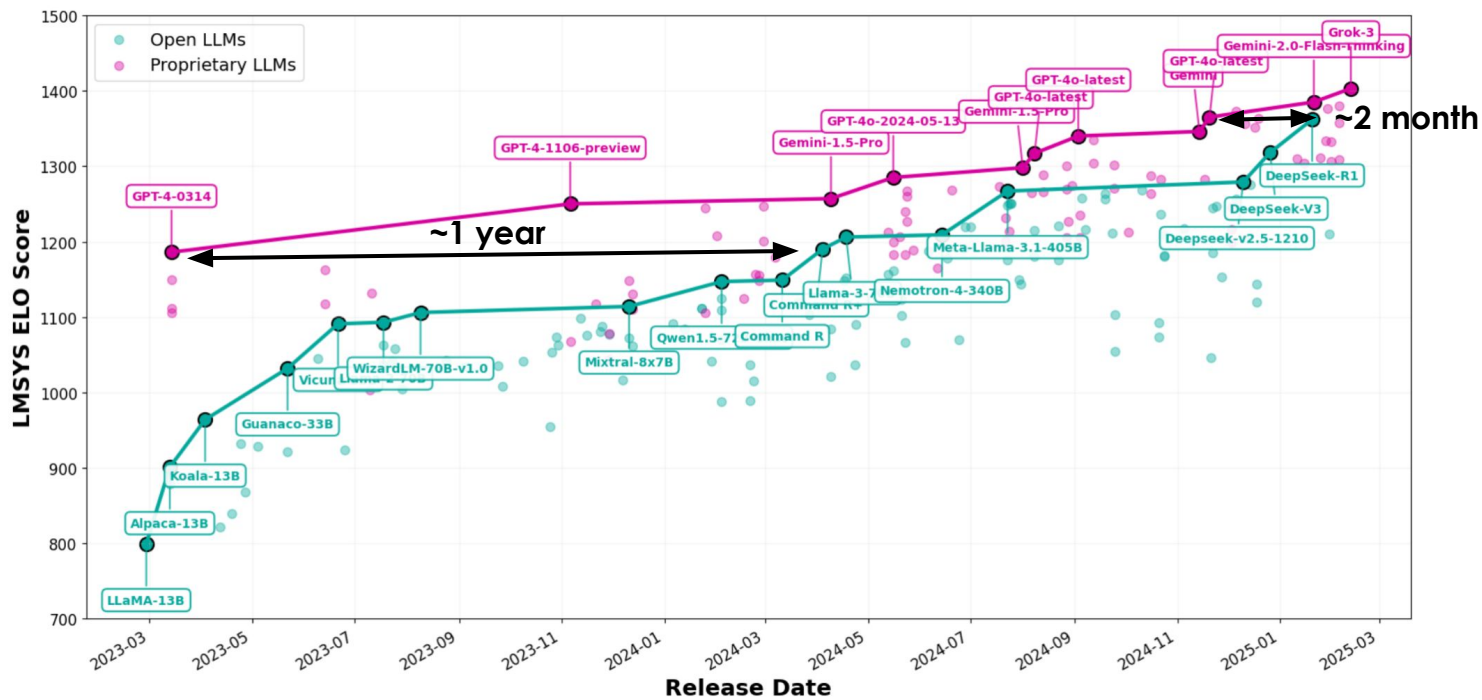
@markchen90



Congrats to DeepSeek on producing an o1-level reasoning model! Their research paper demonstrates that they've independently found some of the core ideas that we did on our way to o1.

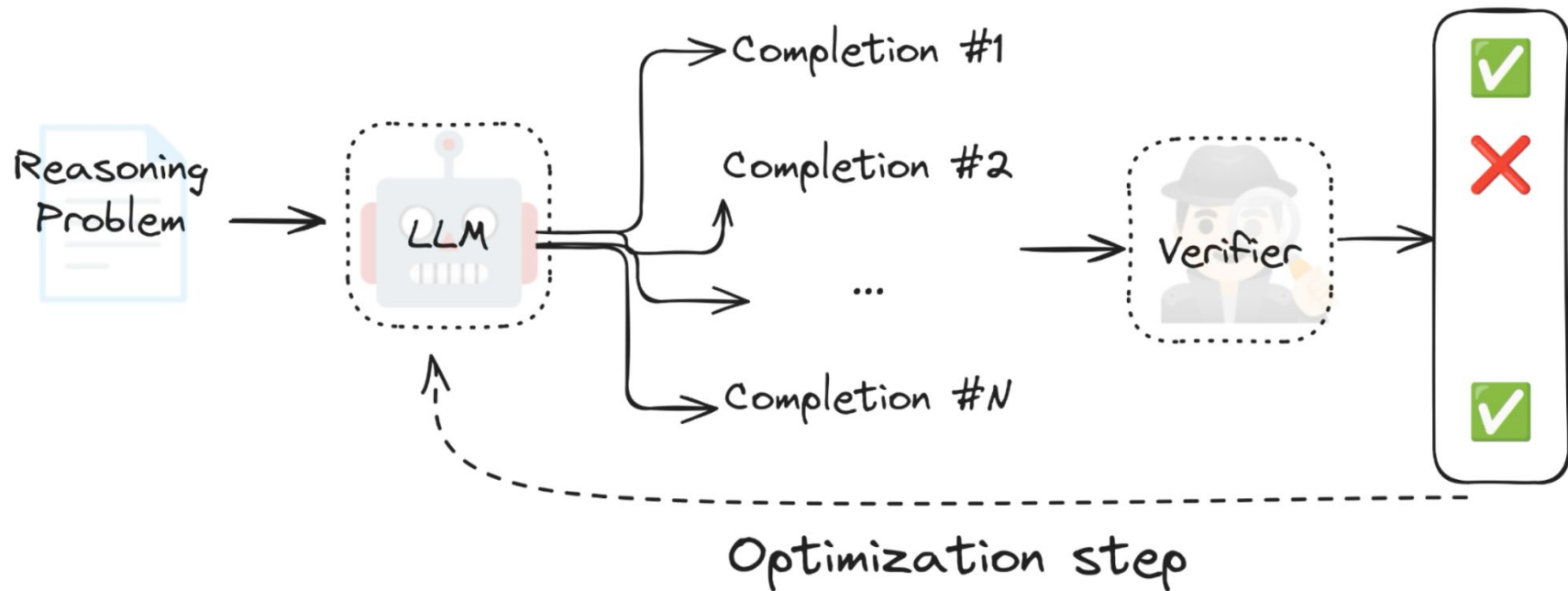
7:11 PM · Jan 28, 2025 · **8.2M** Views

DeepSeek-R1: closing the gap



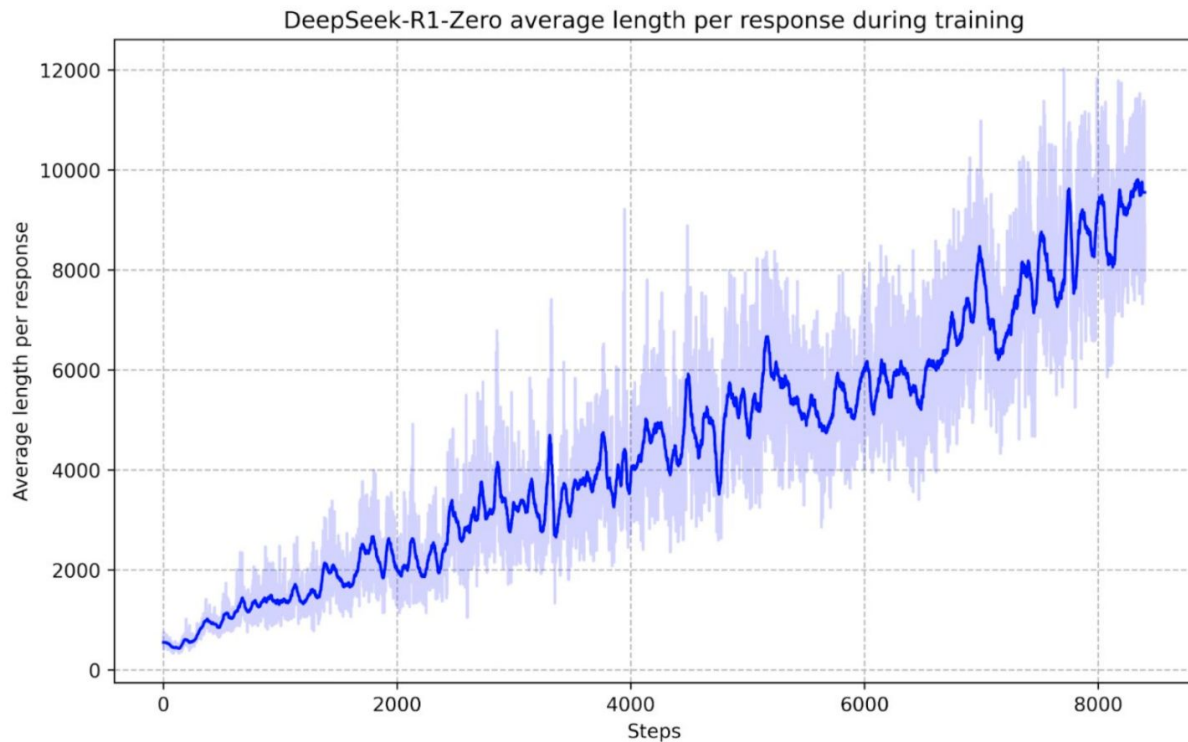
<https://huggingface.co/spaces/andrewreed/closed-vs-open-arena-elo>

DeepSeek-R1: simpler than expected!



Reinforcement Learning with Verifiable Rewards (RLVR)

DeepSeek-R1: increased thinking time



DeepSeek-R1: “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 \dots

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

Rearrange to isolate the inner square root term:

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

\dots

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 \dots

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: \dots

\dots

DeepSeek-R1: distillation into smol models

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	86.7	94.5	65.2	57.5	1633

Can we train **fully open**
reasoning models?

datasets, tooling, knowledge

The **spectrum** of open

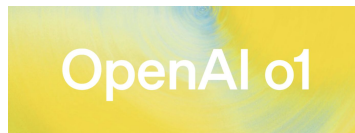
closed model APIs

open model weights

fully open model

 Claude


 Gemini

 OpenAI o1



LLaMA
by  Meta

 Mistral AI

 deepseek

Granite 

 BLOOM 

 StarCoder2
built by 

closed

open

Open **weights** and open **source**

open **weights**:

- can be fine-tuned and aligned
- run locally or on provider
- data contamination
- no control for content creators
- reproducibility

open **source**:

- data is available
- training code is available
- fosters innovation
- builds trust in the systems
- anybody can built on top

Open R1: **fully reproduce DeepSeek-R1**

Many details missing from DeepSeek-R1 tech report:

- Training code?
- Scaling and trade-offs??
- Verifiers???
- Data????

<https://huggingface.co/open-r1>



Hugging Face

Search models, datasets, users...



Models



Datasets

Log in using Single Sign-On to view activity within the huggingface



Open R1

Enterprise

Community

<https://github.com/huggingface/o...>



AI & ML interests

None defined yet.



Recent Activity



lewtun updated a dataset 2 minutes ago

[open-r1/details-open-r1_R1-Distill-Qwen-Math-...](#)

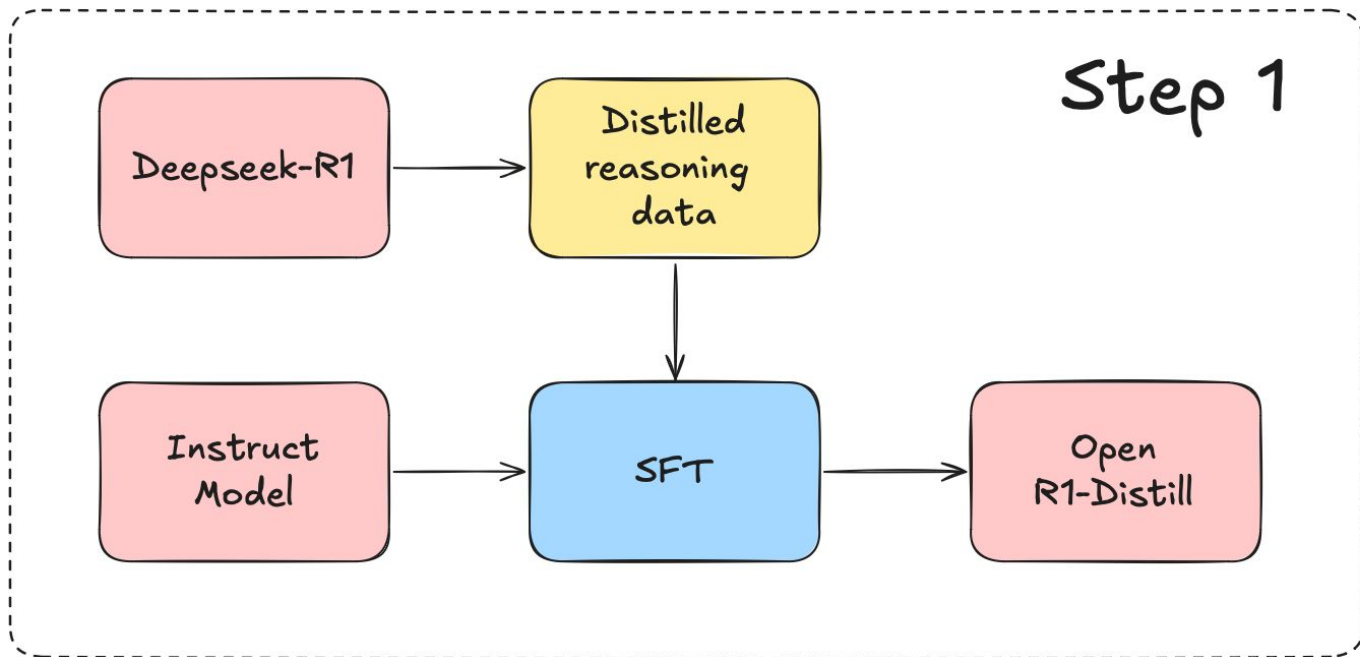


Organization Card

Welcome to Open-R1 🐳 🤗

Open-R1 is an open initiative to replicate and extend DeepSeek-R1 model, in a fully transparent and collaborative way.

Open R1: **step 1 - distilled models**



Distillation goes brr: **math**

Reasoning datasets

updated 21 days ago

Datasets with reasoning traces for math and code released by the community 

bespokelabs/Bespoke-Stratos-17k

 Viewer • Updated Jan 31 •  16.7k •  21.1k •  308

Note 17k reasoning traces for coding and math distilled from  DeepSeek-R1

open-thoughts/OpenThoughts-114k

 Viewer • Updated 30 days ago •  228k •  26k •  701

Note 114k reasoning traces covering math, science, code, and puzzles distilled from  DeepSeek-R1

open-r1/OpenThoughts-114k-math

 Viewer • Updated Jan 30 •  89.1k •  1.02k •  81

Note Filtered version of OpenThoughts-114k based on correct answers

PrimeIntellect/NuminaMath-QwQ-CoT-5M

 Viewer • Updated Jan 22 •  5.14M •  2.35k •  48

Note 5M reasoning traces for math problems distilled from QwQ-32B

ServiceNow-AI/R1-Distill-SFT

 Viewer • Updated Feb 8 •  1.85M •  1.91k •  295

Note 1.7M reasoning traces for math problems distilled from DeepSeek-R1-Distill-Qwen-32B

Datasets 1,582

 reasoning 

Full-text search

↑↓ Sort: Trending

FreedomIntelligence/medical-o1-reasoning-...

 Viewer • Updated 14 days ago •  90.1k •  11.9k •  678

ZennyKenny/synthetic_vc_financial_decisio...

 Viewer • Updated about 17 hours ago •  200 •  553 •  9

ZennyKenny/tactical-military-reasoning-v...

 Viewer • Updated 11 days ago •  150 •  560 •  6

twinkle-ai/tw-function-call-reasoning-10k

 Viewer • Updated 6 days ago •  10k •  188 •  5

nlpllmeval/NLP-Course-LLM-Reasoning-Eval-...

 Viewer • Updated 3 days ago •  918 •  114 •  5

twinkle-ai/tw-reasoning-instruct-50k

 Viewer • Updated 7 days ago •  50k •  179 •  4

AndriyBilinskiy/logical-reasoning-ua

 Viewer • Updated Feb 6 •  20k •  220 •  3

glaiveai/reasoning-v1-20m

 Viewer • Updated Mar 19 •  22.2M •  7.67k •  206

reedmayhew/claude-3.7-sonnet-reasoning

 Viewer • Updated Mar 24 •  179 •  396 •  7

twinkle-ai/tw-math-reasoning-2k

 Viewer • Updated 7 days ago •  2k •  100 •  2

moremilk/CoT_Reasoning_The_Ancient_Past

 Viewer • Updated 6 days ago •  6k •  42 •  2

albertvillanova/visual-spatial-reasoning

Updated Dec 14, 2022 •  38 •  5

Math: finding hard problems



English (eng), day 1

Tuesday, 16. July 2024

Problem 1. Determine all real numbers α such that, for every positive integer n , the integer

$$\lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is a multiple of n . (Note that $\lfloor z \rfloor$ denotes the greatest integer less than or equal to z . For example, $\lfloor -\pi \rfloor = -4$ and $\lfloor 2 \rfloor = \lfloor 2.9 \rfloor = 2$.)

Problem 2. Determine all pairs (a, b) of positive integers for which there exist positive integers g and N such that

$$\gcd(a^n + b, b^n + a) = g$$

holds for all integers $n \geq N$. (Note that $\gcd(x, y)$ denotes the greatest common divisor of integers x and y .)

Problem 3. Let a_1, a_2, a_3, \dots be an infinite sequence of positive integers, and let N be a positive integer. Suppose that, for each $n > N$, a_n is equal to the number of times a_{n-1} appears in the list a_1, a_2, \dots, a_{n-1} .

Prove that at least one of the sequences a_1, a_3, a_5, \dots and a_2, a_4, a_6, \dots is eventually periodic.

(An infinite sequence b_1, b_2, b_3, \dots is *eventually periodic* if there exist positive integers p and M such that $b_{m+p} = b_m$ for all $m \geq M$.)

Solution 1

To solve the problem, we need to find all real numbers α such that, for every positive integer n , the integer

$$S_n(\alpha) = \lfloor \alpha \rfloor + \lfloor 2\alpha \rfloor + \cdots + \lfloor n\alpha \rfloor$$

is divisible by n , i.e., $S_n(\alpha) \equiv 0 \pmod n$.

Step 1: Break Down α into Integer and Fractional Parts

Let $\alpha = m + f$, where $m = \lfloor \alpha \rfloor \in \mathbb{Z}$ and $f = \{\alpha\} \in [0, 1)$ is the fractional part of α .

Step 2: Express the Sum in Terms of m and f

Using this, we have:

$$\lfloor k\alpha \rfloor = \lfloor k(m + f) \rfloor = km + \lfloor kf \rfloor.$$

So, the sum becomes:

$$S_n(\alpha) = m \sum_{k=1}^n k + \sum_{k=1}^n \lfloor kf \rfloor = m \frac{n(n+1)}{2} + \sum_{k=1}^n \lfloor kf \rfloor.$$

Step 3: Modulo n Simplification

We are interested in $S_n(\alpha) \pmod n$:

$$S_n(\alpha) \equiv \left(m \frac{n(n+1)}{2} + \sum_{k=1}^n \lfloor kf \rfloor \right) \pmod n.$$

Since $m \frac{n(n+1)}{2}$ is divisible by n , the expression simplifies to:

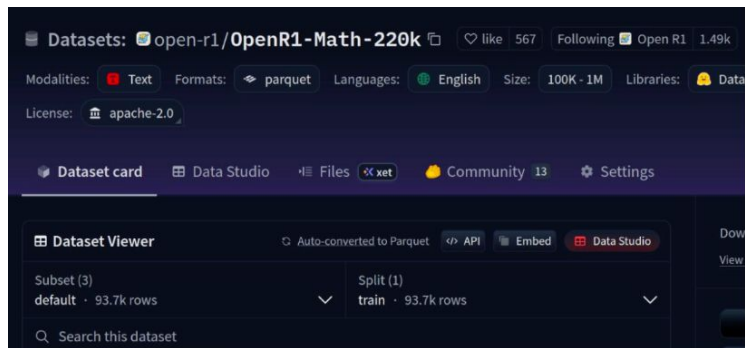
$$S_n(\alpha) \equiv \sum_{k=1}^n \lfloor kf \rfloor \pmod n.$$

Step 4: Analyze the Fractional Part f

Our goal is to find all $f \in [0, 1)$ such that:

$$\sum_{k=1}^n \lfloor kf \rfloor \equiv 0 \pmod n \quad \text{for all } n \in \mathbb{N}.$$

Math: **verifying** model answers



```
from math_verify import parse, verify

# Parse the gold and answer
# If you know that gold will only contain LaTeX
# parse(gold, extraction_config=[LatexE

gold = parse("${1,3} \\cup {2,4}$")
answer = parse("${1,2,3,4}$")

# Order here is important!
verify(gold, answer)

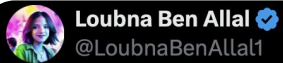
# >>> True
```

- Complete set theory support (Intervals, FiniteSets, set operations)
- Unicode symbol substitution support (e.g. $\beta \rightarrow \text{beta}$)
- Applies Latex fixes for common malformations (e.g. $\text{frac}13 \rightarrow 1/3$)
- Equation and inequality parsing, with symbol assignment resolution (e.g. $x = 1 \rightarrow 1$)
- Percentage best effort conversion (e.g. $10\% \rightarrow 0.1$)
- Units in text handling (e.g. $10 \text{ cm} \rightarrow 10$)
- Exact representation of the input expressions (e.g. $0.333 \rightarrow \text{Float}(333, 3)$)

<https://github.com/huggingface/Math-Verify>



Math: open data for model training



We just published the second OpenR1 update with OpenR1-220k-Math, our new large-scale dataset for mathematical reasoning generated by DeepSeek R1.

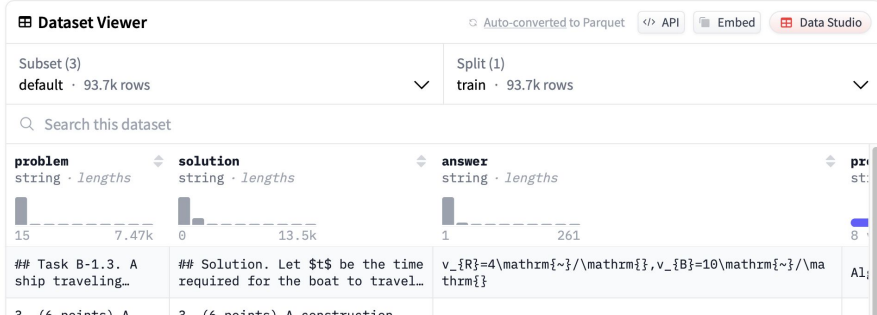
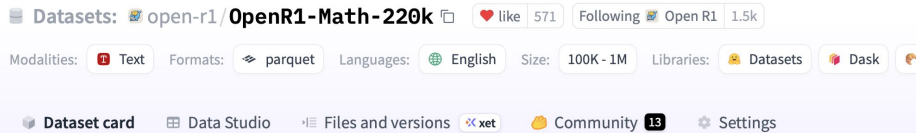
We generated 800k+ reasoning traces on 512 H100s in 3 days 🚀



Community Article Published February 10, 2025

[Edit article](#)

We are now two weeks into the [Open R1 project](#) which aims to reconstruct the missing pieces of DeepSeek R1—specifically, the training pipeline and synthetic data.



[hf.co/datasets/open-r1/OpenR1-Math-220k](https://huggingface.co/datasets/open-r1/OpenR1-Math-220k)

Distillation goes brr: code



HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS STANDINGS CUSTOM INVOCATION

D. Array and GCD

time limit per test: 2 seconds

memory limit per test: 512 megabytes

You are given an integer array a of size n .

You can perform the following operations any number of times (possibly, zero):

- pay one coin and increase any element of the array by 1 (you must have at least 1 coin to perform this operation);
- gain one coin and decrease any element of the array by 1.

Let's say that an array is *ideal* if both of the following conditions hold:

- each element of the array is at least 2;
- for each pair of indices i and j ($1 \leq i, j \leq n; i \neq j$) the greatest common divisor (GCD) of a_i and a_j is equal to 1. If the array has less than 2 elements, this condition is automatically satisfied.

Let's say that an array is *beautiful* if it can be transformed into an ideal array using the aforementioned operations, provided that you initially have no coins. If the array is already ideal, then it is also beautiful.

The given array is not necessarily beautiful or ideal. You can remove any elements from it (including removing the entire array or not removing anything at all). Your task is to calculate the minimum number of elements you have to remove (possibly, zero) from the array a to make it **beautiful**.

Input

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 4 \cdot 10^5$).

The second line contains n integers a_1, a_2, \dots, a_n ($2 \leq a_i \leq 10^9$).

Additional constraint on the input: the sum of n over all test cases doesn't exceed $4 \cdot 10^5$.

Output

For each test case, print a single integer — the minimum number of elements you have to remove (possibly, zero) from the array a to make it **beautiful**.

Example

input

```
5
3
5 5 5
4
2 3 2 4
1
3
3
2 100 2
5
2 4 2 11 2
```

output

```
0
2
0
0
1
```

Note

In the first example, you don't need to delete any elements, because the array is already beautiful. It can be transformed into an ideal array as follows: $[5, 5, 5] \rightarrow [4, 5, 5] \rightarrow [4, 4, 5] \rightarrow [4, 3, 5]$ (you end up with 3 coins).

In the second example, you need to remove 2 elements so that the array becomes beautiful. If you leave the elements $[2, 3]$ and delete the other elements, then the given array is already ideal (and therefore, beautiful).

In the third example, you don't need to delete any elements because the array is already ideal (and thus, beautiful).

In the fourth example, the array is beautiful. It can be transformed into an ideal array as follows:

$[2, 100, 2] \rightarrow [2, 99, 2] \rightarrow [2, 99, 3] \rightarrow [2, 98, 3] \rightarrow [2, 97, 3]$ (you end up with 2 coins).

Contest status

#	When	Who	Problem	Lang	Verdict	Time	Memory
318205236	May/03/2025 12:07 UTC+2	23CS02002	D - Array and GCD	C++23 (GCC 14-64, msys2)	Running on test 15	0 ms	0 KB
318205233	May/03/2025 12:07 UTC+2	SajidZakaria0057	D - Array and GCD	C++20 (GCC 13-64)	Wrong answer on test 4	453 ms	134100 KB
318205003	May/03/2025 12:04 UTC+2	23CS02002	D - Array and GCD	C++23 (GCC 14-64, msys2)	Runtime error on test 4	406 ms	74800 KB
318205011	May/03/2025 12:04 UTC+2	Krityapriya	D - Array and GCD	C++23 (GCC 14-64, msys2)	Wrong answer on test 5	93 ms	7800 KB
318204841	May/03/2025 12:03 UTC+2	23CS02002	D - Array and GCD	C++23 (GCC 14-64, msys2)	Runtime error on test 4	202 ms	43500 KB
318204372	May/03/2025 11:57 UTC+2	RaduM0037	D - Array and GCD	C++17 (GCC 7-32)	Accepted	187 ms	5500 KB

Code: problem dataset & distillation

Datasets: open-r1/codeforces

Modalities: Tabular Text Formats: parquet

Dataset card Data Studio Files and versions

Dataset Viewer (First 5GB)

Split (2)
train · ~9.55k rows (showing the first 9.56k)

Search this dataset

id
string · lengths

aliases
sequence · lengths

contest_id
string · length

3	8	1	19	0	1
852/A			null	852	
852/B			null	852	
852/C			null	852	
852/D			null	852	
852/E			null	852	
852/F			null	852	

< Previous

Datasets: open-r1/codeforces-cots

like 152 Follow Open R1 1.49k

Dataset card Data Studio

Subset (11)
solutions · 47.8k rows

Split (1)
train · 47.8k rows

Search this dataset

prompt
string · lengths

generation
string · lengths

finish_reason
string · classes

api_metadata
dict

interaction_format
string · classes

messages
list · lengths

1.73k~2.9k 51.9%

1.28k~14.2k 12%

stop 95.7%

null 98%

2 100%

You will be given a competitive programming problem. Please reason step by step about the solution, then provide a complete implementation in C++17.

Your solution must read input from standard input (cin), write output to standard output (cout). Do not include any debug prints or additional output.

Put your final solution within a single code block:

```
```cpp
<your code here>
```
```

Problem

<think>

Okay, let's try to figure out how to solve this problem. The problem is that we have an array of odd length, and we can repeatedly remove pairs of adjacent elements until there's one left. We need to find the maximum possible remaining element after all possible operations.

Hmm. Let's think about the process. Each operation removes two adjacent elements. Since the array length is odd, after each operation the length decreases by 1 (because 2 are removed, so the new length is $n-2 + \dots$ wait, no. Wait, when you remove two adjacent elements, the array's length reduces by 2 each time. Wait, the original array length is n , which is odd. Then after each operation, the length becomes $n-2$, which is odd minus 2, which is even? Wait no, wait. Wait, n is odd. For example, if n is 3, after removing two elements, the length is 1. Which is odd. Wait, so each operation reduces the length by 2. So the parity of the length remains odd. Because subtracting 2 from an odd number gives another odd number. Wait, no. Wait $3-2=1$ (odd), $1-2=-1$,

stop

{
 "completion_tokens": 2504,
 "prompt_tokens": 878,
 "prompt_tokens_details": null,
 "total_tokens": 3382
}

null

[{ "content": "Yo
\n\nYour solution i
your final solutio
odd.\n\nIn one ope
array \$\$\$[4,7,4,2,
\n\underline{4,2},9]
\n\underline{4},7,\nvalue of the remai
The first line con
each test case con
contains \$\$\$n\$\$\$ i
\$\$\$n\$\$\$ over all t
Examples\n```\ninput
[6]\$\$\$
Since the
array \$\$\$a\$\$\$ is \$
2}] \\\nto [1]\$\$\$
T
One way to maximiz
remaining element
array of odd length
possible operation
length decreases b
each time. Wait, t
Wait, n is odd. Fo

Code: OlympicCoder



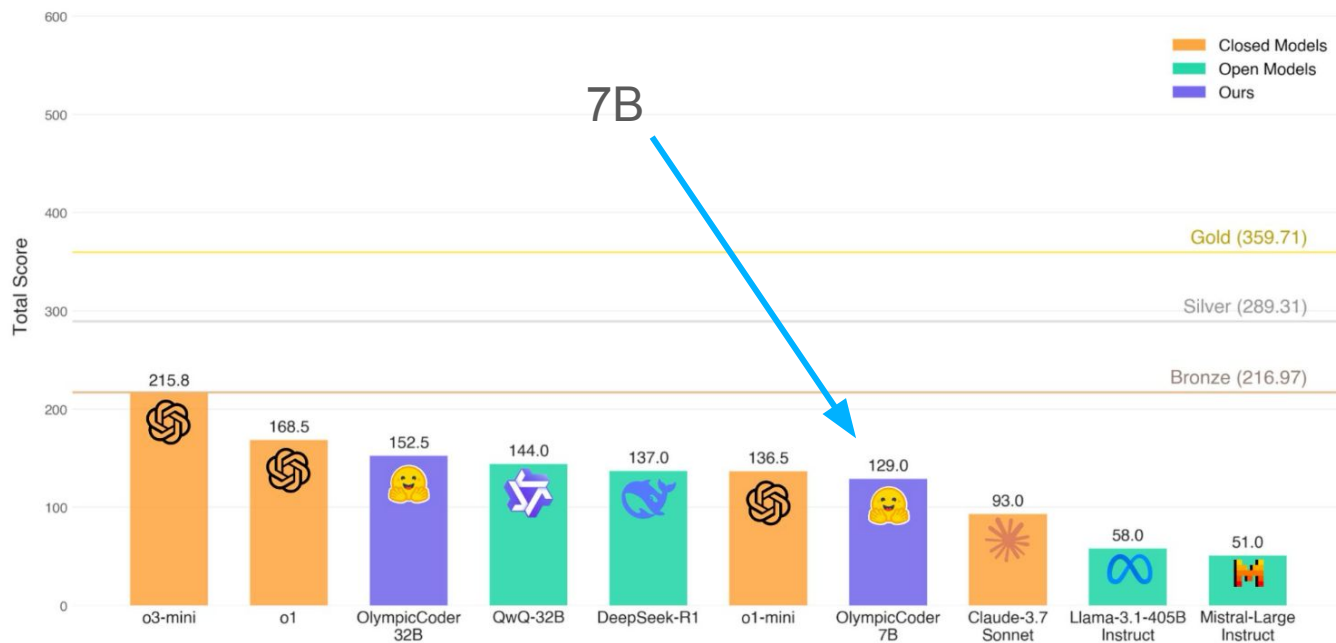
IOI 2024 Eg

Practice Tasks

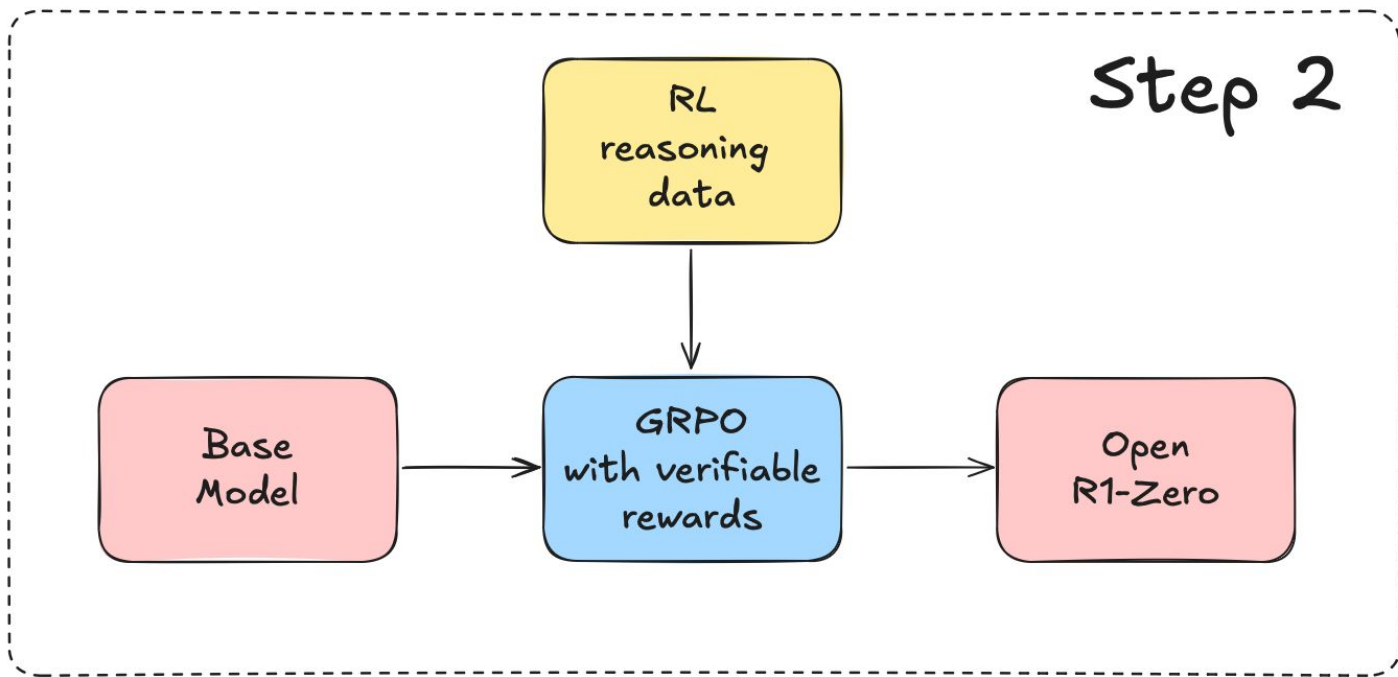
- **aplusb**
 - Statement: [Official English](#)
 - [Task attachments](#)
 - [Test data](#)
- **machine**
 - Statement: [Official English](#)
 - [Task attachments](#)
 - [Test data](#)
- **pyramids**
 - Statement: [Official English](#)
 - [Task attachments](#)
 - [Test data](#)
- **treasure**
 - Statement: [Official English](#)
 - [Task attachments](#)

International Olympiad in Informatics (2024)

Performance of 10 selected models across 50 submissions



Open R1: **step 2 - RL with verifiable rewards**



TRL: reinforcement learning for LLMs

 **Quentin Gallouédec** ✓
@QGallouedec

 TRL 0.14 – Featuring GRPO! 

TRL 0.14 brings *GRPO*, the RL algorithm behind  DeekSeek-R1.

⚡ Blazing fast generation with vLLM integration.
 Optimized training with DeepSpeed ZeRO 1/2/3.


GRPO x  deepspeed
vLLM 

```
○○○  
!pip install trl  
from trl import GRPOTrainer  
trainer = GRPOTrainer("Qwen/Qwen2.5-7B", reward_funcs, train_dataset=train_dataset)
```

3:53 PM · Jan 30, 2025 · 12.7K Views

GRPOTrainer

`GRPOTrainer` implements the [Group Relative Policy Optimization \(GRPO\) algorithm](#) that is more memory-efficient than PPO and was used to train [Deepseek AI's R1](#).

```
from datasets import load_dataset  
from trl import GRPOTrainer  
  
dataset = load_dataset("trl-lib/tldr", split="tr.  
  
# Dummy reward function: count the number of uni  
def reward_num_unique_chars(completions, **kwargs  
    return [len(set(c)) for c in completions]  
  
trainer = GRPOTrainer(  
    model="Qwen/Qwen2-0.5B-Instruct",  
    reward_funcs=reward_num_unique_chars,  
    train_dataset=dataset,  
)  
trainer.train()
```

Code verifiability crisis

“verifiable” problems with weak tests only

| Datasets: deepmind/code_contests like 166 Follow Deepmind 275 Dataset c | |
|---|---|
| Split (3)
train · 3.76k rows | |
| Filtered by source with 2 | |
| public_tests
sequence | private_tests
sequence |
| <pre>{ "input": ["5 5\n1 2 3 4 5\n", "3 1\n1 1 2\n", "5 2\n1 1 2 1 3\n", "output": ["16", "3", "14"] }</pre> | <pre>{ "input": ["50 1\n50 8 46 9 12 38 41 18 49 10 23 15 16 3 13 17 48 8 31 32 6 31 31 49 9
58 31 61 4 59 5 55 33 41 81 55 58 23 95 98 60 62 54 94 47 33 20 67 31 67 34 26 47 96 96]</pre> |
| <pre>{ "input": ["6 4\n2 5 5 6 6\n1 3\n2 4\n3 5\n4 6\n", "3 3\n314 159 265\n1 2\n1 3\n3 2\n", "4 2\n2 1 1 5\n1 2\n1 3 4\n", "4
2\n1 5 1 5\n1 2\n1 3 4\n", "output": ["LOSE\n", "WIN\n0 0 0\n", "WIN\n1 5 1 5\n", "LOSE\n"] }</pre> | <pre>{ "input": ["2 1\n1000000000 1000000000\n2 1\n", "3 2\n123 345 567\n1 2\n3 2\n", "3 0\n\n", "WIN\n0 0\n", "WIN\n678678678 395063145 1073741823 2 3 1\n", "WIN\n0 4\n", "LO</pre> |
| <pre>{ "input": ["3 10\n2 3 2\n6\n1 5\n2 1 10\n2 2 9\n2 5 6\n3 1 7 8\n4 1 2 3 4\n", "output": [
"YES\nYES\nNO\nNO\nYES\nNO\n"] }</pre> | <pre>{ "input": [], "output": [] }</pre> |
| <pre>{ "input": ["2\n1 2\n", "3\n1 2\n2 3\n", "5\n1 2\n1 3\n1 4\n2 5\n", "6\n1 2\n1 3\n1 4\n2 5\n2 6\n", "output": ["YES",
"NO", "NO", "YES"] }</pre> | <pre>{ "input": ["50\n16 4\n17 9\n31 19\n22 10\n8 1\n40 30\n3 31\n20 29\n47 27\n22 25\n32 34
35\n24 41\n30 15\n31 27\n8 44\n26 5\n26 48\n8 32\n53 17\n3 34\n3 51\n30 28\n47 10\n53 60</pre> |
| <pre>{ "input": ["3\n0 0 0\n", "5\n0 1 1 1 10\n", "2\n0 1\n", "output": ["3 2 1", "1 4 3 2 5", "1 2"] }</pre> | <pre>{ "input": ["100\n0 57 121 57 0 19 251 19 361 19 160 57 578 664 57 19 50 0 621 91 5 2
89 42 31 36 5 68 37 10 49 24 7 32 65 59 14 12 11 6 27 34 91 72 21 87 98 3 97 25 100 46 8</pre> |
| <pre>{ "input": ["3\n10 20 10\n6\n1 1\n2 1\n2 2\n3 1\n3 2\n3 3\n", "7\n1 2 1 3 1 2 1\n9\n2 1\n2 2\n3 1\n3 2\n3 3\n1 1\n7 1\n7
7\n7 4\n", "output": ["20\n10 20\n10\n20\n10\n", "2\n3\n2\n3\n2\n3\n2\n3\n1\n1\n3\n"] }</pre> | <pre>{ "input": ["2\n1 10\n3\n2 2\n2 1\n1 1\n", "2\n3922 3922\n3\n2 2\n2 1\n1 1\n", "1\n1000</pre> |
| <pre>{ "input": ["6 3\n3 1 2 3 2 3\n", "10 1\n1 1 1 1 1 1 1 1\n", "12 4\n1 2 3 1 2 3 4 1 2 1 2 1\n", "output": [
"6\n", "1\n10\n", "12\n"] }</pre> | <pre>{ "input": [], "output": [] }</pre> |
| <pre>{ "input": ["3\n1 2 3\n", "5\n5 4 3 2 1\n", "output": ["0 0 0\n", "0 1 3 6 10\n"] }</pre> | <pre>{ "input": ["1\n1\n", "100\n98 52 63 2 18 96 31 58 84 40 41 45 66 100 46 71 26 48 81 20
1364 1377 1409 1445 1455 1461 1534 1553 1565 1572 1581 1664 1706 1715 1779 1787 1837 184</pre> |
| <pre>{ "input": ["5 3\n00011\n3\n1 2 3\n1\n4\n3\n3 4 5\n", "8 6\n00110011\n3\n1 3 8\n5\n1 2 5 6 7\n2\n6 8\n2\n3 5\n2\n4
7\n1\n2\n", "19 5\n1001001100000110\n2\n2 3\n2\n5 6\n2\n8 9\n5\n12 13 14 15 16\n1\n19\n", "7 3\n0011100\n3\n1 4 6\n3\n</pre> | <pre>{ "input": ["1 1\n1\n1\n1\n", "output": ["0\n"] }</pre> |
| <pre>{ "input": ["3\n1 3 2\n-100 2 3\n", "2\n2 1\n-3 0\n", "5\n2 1 4 3 5\n2 2 2 3 4\n", "output": ["3\n", "0\n", "19\n"] }</pre> | <pre>{ "input": [], "output": [] }</pre> |

Code verifiability crisis

| Problem | Lang | Verdict | Time | Memory |
|--|-----------|--------------------------------|---------|-----------|
| 1246E - To Make 1 | PyPy 3-64 | Wrong answer on test 4 | 93 ms | 0 KB |
| 1148G - Gold Experience | PyPy 3-64 | Wrong answer on test 14 | 687 ms | 124500 KB |
| 1148H - Holy Diver | PyPy 3-64 | Time limit exceeded on test 26 | 3000 ms | 72600 KB |
| 1515H - Phoenix and Bits | PyPy 3-64 | Time limit exceeded on test 7 | 4000 ms | 65400 KB |
| 1056H - Detect Robots | PyPy 3-64 | Wrong answer on test 3 | 77 ms | 0 KB |
| 1515H - Phoenix and Bits | Python 3 | Time limit exceeded on test 7 | 4000 ms | 59600 KB |
| 1491I - Ruler Of The Zoo | Python 3 | Wrong answer on test 4 | 406 ms | 900 KB |

Code verifiability crisis



2025-01-06

CODEELO: Benchmarking Competition-level Code Generation of LLMs with Human-comparable Elo Ratings

B. Replace Character

time limit per test: 1 second

memory limit per test: 256 megabytes

You're given a string s of length n , consisting of only lowercase English letters.

You must do the following operation exactly once:

- Choose any two indices i and j ($1 \leq i, j \leq n$). You can choose $i = j$.
- Set $s_i := s_j$.

You need to minimize the number of distinct permutations[†] of s . Output any string with the smallest number of distinct permutations after performing **exactly one** operation.

[†] A permutation of the string is an arrangement of its characters into any order. For example, "bac" is a permutation of "abc" but "bcc" is not.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 500$). The description of the test cases follows.

The first line of each test case contains n ($1 \leq n \leq 10$) — the length of string s .

The second line of each test case contains s of length n . The string contains only lowercase English letters.

Output

For each test case, output the required s after applying exactly one operation. If there are multiple solutions, print any of them.

A clear indication that needs a special judge

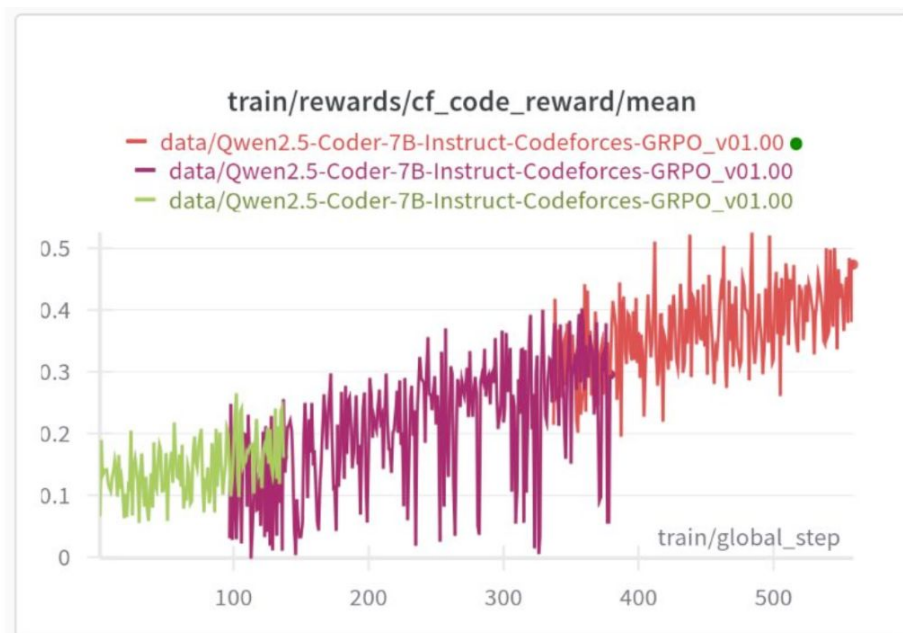
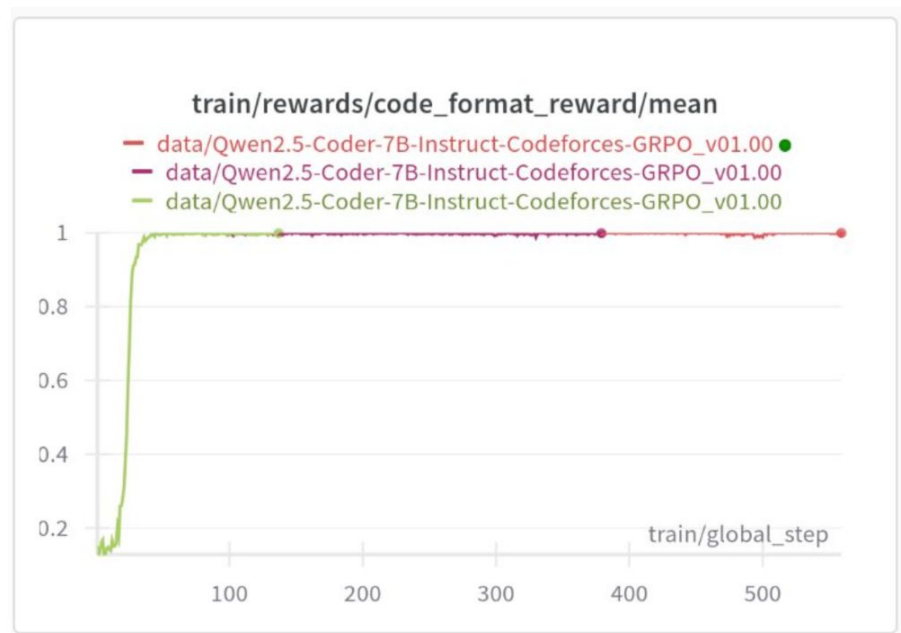
~30% require a special checker

R1 generated checkers working for 60%+ of them

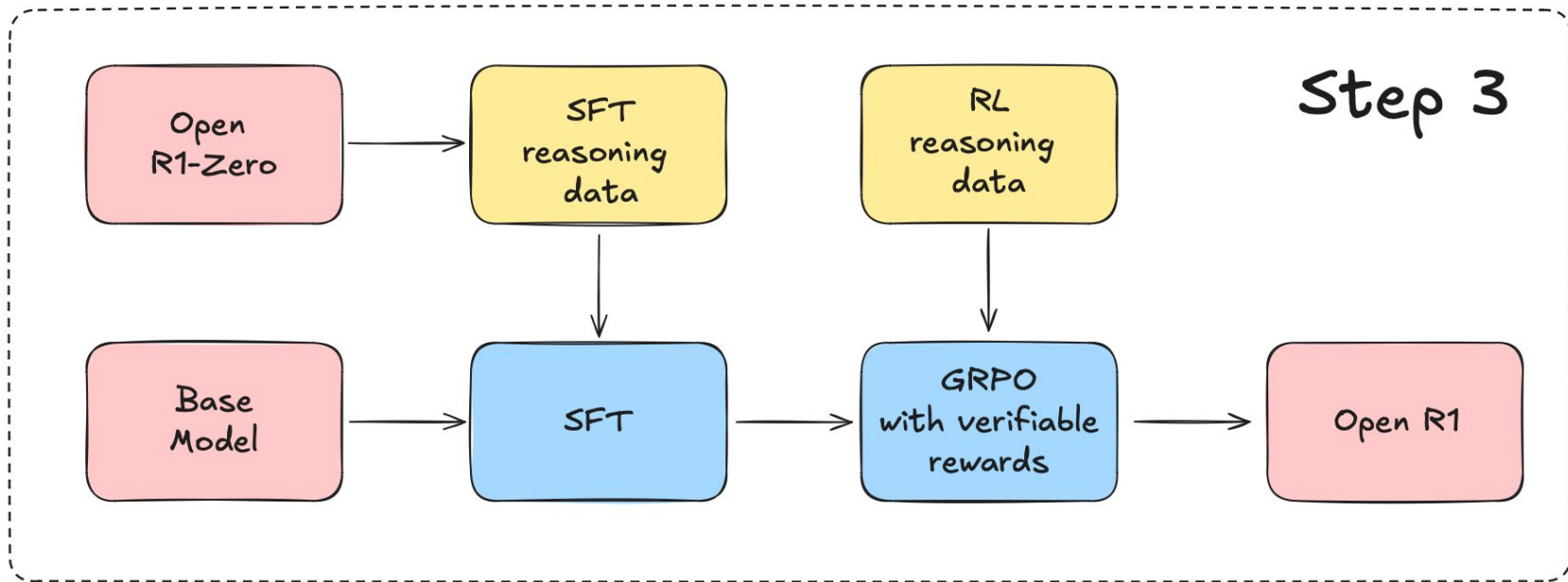
Validated with correct and incorrect solutions

-> Apply same idea to test cases

Code reward curves



Open R1: **step 3** - bring it altogether (WIP)



What **impact** has DeepSeek-R1 had on **open AI**?

DeepSeek-R1: huge impact on open AI!



We now have much better
tools for reinforcement
learning 💪



DeepSeek-R1: huge impact on open AI!

Datasets 1,582

reasoning

Full-text search

Sort: Trending

FreedomIntelligence/medical-o1-reasoning-...

Viewer • Updated 14 days ago • 90.1k • 11.9k • 678

ZennyKenny/synthetic_vc_financial_decisio...

Viewer • Updated about 17 hours ago • 200 • 553 • 9

ZennyKenny/tactical-military-reasoning-v...

Viewer • Updated 11 days ago • 150 • 560 • 6

twinkle-ai/tw-function-call-reasoning-10k

Viewer • Updated 6 days ago • 10k • 188 • 5

nlp11meval/NLP-Course-LLM-Reasoning-Eval-...

Viewer • Updated 3 days ago • 918 • 114 • 5

twinkle-ai/tw-reasoning-instruct-50k

Viewer • Updated 7 days ago • 50k • 179 • 4

AndriyBilinskiy/logical-reasoning-ua

Viewer • Updated Feb 6 • 20k • 220 • 3

glaiveai/reasoning-v1-20m

Viewer • Updated Mar 19 • 22.2M • 7.67k • 206

reedmayhew/claude-3.7-sonnet-reasoning

Viewer • Updated Mar 24 • 179 • 396 • 7

twinkle-ai/tw-math-reasoning-2k

Viewer • Updated 7 days ago • 2k • 100 • 2

moremilk/CoT_Reasoning_The_Ancient_Past

Viewer • Updated 6 days ago • 6k • 42 • 2

albertvillanova/visual-spatial-reasoning

Updated Dec 14, 2022 • 38 • 5

**1000+ reasoning datasets
now available, with large
orgs like NVIDIA also
contributing**

DeepSeek-R1: huge impact on open AI!

Models 225,078

Filter by name

Full-text search

Sort: Trending

deepseek-ai/DeepSeek-Prover-V2-671B

Text Generation • Updated 6 days ago • 3.47k • 696

Qwen/Qwen3-235B-A22B

Text Generation • Updated 6 days ago • 42.8k • 702

JetBrains/Mellum-4b-base

Text Generation • Updated 7 days ago • 1.2k • 243

XiaomiMiMo/MiMo-7B-RL

Text Generation • Updated 1 day ago • 3.01k • 225

microsoft/Phi-4-reasoning-plus

Text Generation • Updated about 5 hours ago • 5.02k • 206

Qwen/Qwen3-30B-A3B

Text Generation • Updated 7 days ago • 92k • 462

microsoft/Phi-4-reasoning

Text Generation • Updated about 5 hours ago • 3.78k • 147

Reasoning models now
available in many shapes
and sizes

Thank you! 🙌



@lewtun



@_lewtun