Title: Reasoning language model

URL: https://en.wikipedia.org/wiki/Reasoning_language_model

PageID: 79013463

Categories: Category:Artificial intelligence, Category:Automated reasoning, Category:Large language models, Category:Machine learning, Category:Natural language processing

-----

Reasoning language models (RLMs) are large language models that are trained further to solve tasks that take several steps of reasoning . [1] They tend to do better on logic, math, and programming tasks than standard LLMs, can revisit and revise earlier steps, and make use of extra computation while answering as another way to scale performance , alongside the number of training examples, parameters, and training compute. [2]

History

2024

In September 2024, OpenAI released o1-preview , an LLM with enhanced reasoning. [3] The full version, o1 , followed in December 2024. OpenAI also began sharing results on its successor, o3 . [4] [5] [6]

The development of reasoning LLMs has illustrated what Rich Sutton called the "bitter lesson": that scaling compute often outperforms methods that rely on specific human insights. [7] For example, the Generative AI Research Lab (GAIR) explored complex methods such as tree search and reinforcement learning to replicate o1's capabilities. In their "o1 Replication Journey" papers they reported that knowledge distillation (training a smaller model to imitate o1's outputs) worked surprisingly well. This highlighted the effectiveness of distillation in this context. [8] [9]

Alibaba released reasoning versions of its Qwen LLMs in November 2024. [10] In December 2024, the team introduced QvQ-72B-Preview, an experimental visual reasoning model. [11]

In December 2024, Google introduced Deep Research in Gemini , [12] a feature that runs multi-step research tasks. [13]

On December 16, 2024, an experiment with a Llama 3B model showed that by scaling test-time compute, a relatively small model could outperform a much larger Llama 70B model on challenging reasoning tasks. This suggested that better inference strategies can unlock useful reasoning capabilities even in small models. [14] [15]

2025

In January 2025, DeepSeek released R1 , a model with comparable performance to o1 at lower cost. The release demonstrated the effectiveness of Group Relative Policy Optimization (GRPO). [16] [17] On January 25, 2025, DeepSeek added a feature to DeepSeek R1 that lets the model search the web while it reasons, making it easier to combine retrieval with reasoning. [18] The effectiveness of distillation for reasoning models was shown in works such as s1-32B, which achieved strong performance through budget forcing and scaling methods. [19] [9]

On February 2, 2025, OpenAI released Deep Research based on their o3 model, [20] allowing users to initiate complex research tasks and generate comprehensive reports which incorporate various sources from the web. [20]

Supervised finetuning

A large language model (LLM) can be fine-tuned on a dataset of reasoning tasks paired with example solutions and step-by-step (reasoning) traces. The fine-tuned model can then produce its own reasoning traces for new problems. [21] [22]

Because human-written traces are costly to collect, researchers have proposed ways to build such datasets automatically. In rejection sampling finetuning (RFT), new reasoning traces are gathered in a loop: [23]

Sample a task prompt.

Generate many reasoning traces for the prompt.

Use a verifier to remove reasoning traces with a wrong final answer, and optionally remove duplicates

Reinforcement learning

A pretrained language model can be further trained with RL. In the RL formalism, a generative language model is a policy π {\displaystyle \pi } . A task prompt is an environmental state x {\displaystyle x} , and the model's response is an action y {\displaystyle y} . The probability that the model responds x {\displaystyle x} with y {\displaystyle y} is π (y | x) {\displaystyle \pi (y|x)} .

Training a reasoning language model with RL means constructing a reward model r (x , y) {\displaystyle r(x,y)} to guide the RL process. Intuitively, the reward says how good a response is for a prompt. For a reasoning task, the reward is high if the response solves the task and low if it does not.

A response y {\displaystyle y} may be broken-down into multiple steps, written y 1 , y 2 , … , y n {\displaystyle y_{1},y_{2},\dots ,y_{n}} .

Most recent systems use policy-gradient methods such as Proximal Policy Optimization (PPO) because PPO constrains each policy update with a clipped objective, which stabilises training for very large policies. [24]

Outcome reward model

An outcome reward model, or outcome-supervised RM (ORM), [21] gives the reward for a step r (x , y 1 , … , y i) {\displaystyle r(x,y_{1},\dots ,y_{i})} based on the final answer: r (x , y 1 , … , y i) = r (x , y n) {\displaystyle r(x,y_{1},\dots ,y_{i})=r(x,y_{n})} . Such models are often called "verifiers".

For tasks with answers that are easy to verify, such as math word problems , the outcome reward can be binary: 1 if the final answer is correct, 0 otherwise. [21] If automatic verification is hard, humans can label answers as correct or not, and those labels can be used to finetune a base model that predicts the human label. [22] For tasks like creative writing, where quality is not simply true or false, one can train a reward model on human ranked preference data, as in reinforcement learning from human feedback . [25] A base model can also be fine-tuned to predict, from a partial thinking trace x , y 1 , … , y m {\displaystyle x,y_{1},\dots ,y_{m}} , whether the final answer will be correct, and this prediction can serve as a binary reward. [21]

The ORM is usually trained with logistic regression , i.e. by minimizing cross-entropy loss . [26]

Given a PRM, an ORM can be constructed by multiplying the total process reward during the reasoning trace, [25] by taking the minimum, [26] or by other ways of aggregating process rewards. DeepSeek used a simple ORM to train the R1 model . [17]

Process reward model

A process reward model, or process-supervised RM (PRM), [21] gives the reward for a step r (x , y 1 , … , y i) {\displaystyle r(x,y_{1},\dots ,y_{i})} based only on the steps so far: (x , y 1 , … , y i) {\displaystyle (x,y_{1},\dots ,y_{i})} .

Given a partial thinking trace x , y 1 , … , y m {\displaystyle x,y_{1},\dots ,y_{m}} , a human can judge whether the steps so far are correct, without looking at the final answer. This yields a binary reward. Because human labels are costly, a base model can be fine-tuned to predict them. [21] The PRM is usually trained with logistic regression on the human labels, i.e. by minimizing the cross-entropy loss between true and predicted labels. [26]

As an example, a 2023 OpenAI paper collected 800K process labels for 75K thinking traces. A labeler saw a trace and marked each step as "positive" if it moved toward a solution, "neutral" if it was not wrong but did not help, and "negative" if it was a mistake. After the first "negative" label, the labeler stopped on that trace and moved to another. The authors argued that labeling up to the first error was enough to train a capable PRM, even though labeling later steps could give richer signals. [25] [27]

To avoid human labels, researchers have proposed methods to create PRM without human labels on the processes. Inspired by Monte Carlo tree search (MCTS), the Math-Shepherd method samples multiple continuations until the end, starting at each reasoning step y i {\displaystyle y_{i}} , and set the reward at that step to be either # (correct answers) # (total answers) {\displaystyle {\frac {\#{\text{(correct answers)}}}{\#{\text{(total answers)}}}}} in the case of "soft estimation", or { 1 if one of the answers is correct 0 else {\displaystyle {\begin{cases}1&{\text{if one of the answers is correct}}\\0&{\text{else}}\end{cases}}} in the case of "hard estimation". This creates process rewards from an ORM, which is often easier or cheaper to construct. A PRM can then be trained on these labels. [26] Some work has tried a fully MCTS approach. [28]

One can also use an ORM to implicitly construct a PRM, similar to direct preference optimization . [29]

Guided sampling

A trained ORM can be used to pick the best response. The policy generates several responses, and the ORM selects the best one. This implements a simple form of test-time compute scaling ("best-of-N"). [22] [30]

A trained PRM can guide reasoning by a greedy tree search : the policy proposes several next steps, the PRM picks one, and the process repeats. This mirrors using an ORM to pick a whole response. [31] Beam search performs better than greedy search.

Lookahead search is another tree search method. The policy proposes several next steps, then makes a short rollout for each. If a solution is found during rollout, the search stops early. Otherwise, the PRM scores each rollout, and the step with the highest score is chosen. [15]

Self-consistency can be combined with an ORM. The model generates multiple answers, and the answers are clustered so that each cluster has the same final answer. The ORM scores each answer, scores in each cluster are summed, and the answer from the highest-scoring cluster is returned. [26]

Benchmarks

Reasoning models generally score higher than non-reasoning models on many benchmarks, especially on tasks requiring multi-step reasoning. [32] [33] [34] [35] [36] [37] [38]

Some benchmarks exclude reasoning models because their responses take longer and cost more. [39] [40] [41] [42]

Humanity's Last Exam

The HLE benchmark tests expert-level reasoning across mathematics, humanities, and the natural sciences, and shows large performance gaps between models. State-of-the-art reasoning models score low on HLE, leaving room to improve. For example, the full reasoning model o3 reached 26.6%, [20] while the lighter o3-mini-high (on text-only questions) reached 13%. [43]

AIME

On the American Invitational Mathematics Examination (AIME), a difficult math competition, non-reasoning models usually solve under 30% of problems. Models that use reasoning methods score between 50% and 80%. [2] [17] [19] While OpenAI's o1 maintained or slightly improved its accuracy from reported 2024 results to 2025 AIME results, o3-mini (high) reached a higher accuracy (80%) at a much lower cost (about 12 times cheaper). [44]

o3-mini performance

According to OpenAI's January 2025 report on o3-mini, adjusting "reasoning effort" significantly affects performance, especially for STEM tasks. Moving from low to high reasoning effort raises accuracy on AIME 2024, GPQA Diamond, and Codeforces , typically by 10–30%. With high effort, o3-mini (high) achieved 87.3% on AIME (different from the MathArena AIME benchmark), 79.7% on GPQA Diamond, 2130 Elo on Codeforces, and 49.3 on SWE-bench Verified. [44]

Drawbacks

Computational cost

Reasoning models often need far more compute while answering than non-reasoning models. On AIME, they were 10 to 74 times more expensive [25] than non-reasoning counterparts.

Generation time

Due to the tendency of reasoning language models to produce verbose outputs, the time it takes to generate an output increases greatly when compared to a standard large language model .

Models

OpenAI

GPT-5

o4-mini

o3 and o3-mini

o1 and o1-preview

Gemini

2.5 Pro and Flash

2.0 Flash Thinking

DeepSeek

R1 (based on V3)

R1-Lite-Preview (test version based on V2.5)

Qwen

QvQ-72B-Preview — an experimental visual reasoning model launched on December 24, 2024, which integrates image understanding with verbal chain-of-thought reasoning.

QwQ-32B-Preview — an experimental text-based reasoning model released in late November 2024 that emphasizes complex, step-by-step analysis.

Anthropic

Claude Sonnet 3.7 has an adjustable amount of 'thinking' tokens.

Mistral AI

Magistral (medium & small)

xAI

Grok 3

Grok 4

Hugging Face

OlympicCoder-7B & 32B, as part of reproducing the R1 training openly (Open R1 project). [45] [46]

See also

Automated reasoning

Reflection (artificial intelligence)

Large language model

References

External links

Fortes, Armando (2025-01-27). "atfortes/Awesome-LLM-Reasoning" . GitHub . Retrieved 2025-01-27 .

Huang, Jie; Chang, Kevin Chen-Chuan (2023-05-26). "Towards Reasoning in Large Language Models: A Survey". arXiv : 2212.10403 [cs.CL].

Besta, Maciej; Barth, Julia; Schreiber, Eric; Kubicek, Ales; Catarino, Afonso; Gerstenberger, Robert; Nyczyk, Piotr; Iff, Patrick; Li, Yueling (2025-01-23). "Reasoning Language Models: A Blueprint". arXiv : 2501.11223 [cs.AI].

v

t

e

History timeline

timeline

Companies

Projects

Parameter Hyperparameter

Hyperparameter

Loss functions

Regression Bias–variance tradeoff Double descent Overfitting

Bias–variance tradeoff

Double descent

Overfitting

Clustering

Gradient descent SGD Quasi-Newton method Conjugate gradient method

SGD

Quasi-Newton method

Conjugate gradient method

Backpropagation

Attention

Convolution

Normalization Batchnorm

Batchnorm

Activation Softmax Sigmoid Rectifier

Softmax

Sigmoid

Rectifier

Gating

Weight initialization

Regularization

Datasets Augmentation

Augmentation

Prompt engineering

Reinforcement learning Q-learning SARSA Imitation Policy gradient

Q-learning

SARSA

Imitation

Policy gradient

Diffusion

Latent diffusion model

Autoregression

Adversary

RAG

Uncanny valley

RLHF

Self-supervised learning

Reflection

Recursive self-improvement

Hallucination

Word embedding

Vibe coding

Machine learning In-context learning

In-context learning

Artificial neural network Deep learning

Deep learning

Language model Large language model NMT

Large language model

NMT

Reasoning language model

Model Context Protocol

Intelligent agent

Artificial human companion

Humanity's Last Exam

Artificial general intelligence (AGI)

AlexNet

WaveNet

Human image synthesis

HWR

OCR

Computer vision

Speech synthesis 15.ai ElevenLabs

15.ai

ElevenLabs

Speech recognition Whisper

Whisper

Facial recognition

AlphaFold

Text-to-image models Aurora DALL-E Firefly Flux Ideogram Imagen Midjourney Recraft Stable Diffusion

Aurora

DALL-E

Firefly

Flux

Ideogram

Imagen

Midjourney

Recraft

Stable Diffusion

Text-to-video models Dream Machine Runway Gen Hailuo AI Kling Sora Veo

Dream Machine

Runway Gen

Hailuo AI

Kling

Sora

Veo

Music generation Riffusion Suno AI Udio

Riffusion

Suno AI

Udio

Word2vec

Seq2seq

GloVe

BERT

T5

Llama

Chinchilla AI

PaLM

GPT 1 2 3 J ChatGPT 4 4o o1 o3 4.5 4.1 o4-mini 5

1

2

3

J

ChatGPT

4

4o

o1

o3

4.5

4.1

o4-mini

5

Claude

Gemini Gemini (language model) Gemma

Gemini (language model)

Gemma

Grok

LaMDA

BLOOM

DBRX

Project Debater

IBM Watson

IBM Watsonx

Granite

PanGu-Σ

DeepSeek

Qwen

AlphaGo

AlphaZero

OpenAI Five

Self-driving car

MuZero

Action selection AutoGPT

AutoGPT

Robot control

Alan Turing

Warren Sturgis McCulloch

Walter Pitts

John von Neumann

Claude Shannon

Shun'ichi Amari

Kunihiko Fukushima

Takeo Kanade

Marvin Minsky

John McCarthy

Nathaniel Rochester

Allen Newell

Cliff Shaw

Herbert A. Simon

Oliver Selfridge

Frank Rosenblatt

Bernard Widrow

Joseph Weizenbaum

Seymour Papert

Seppo Linnainmaa

Paul Werbos

Geoffrey Hinton

John Hopfield

Jürgen Schmidhuber

Yann LeCun

Yoshua Bengio

Lotfi A. Zadeh

Stephen Grossberg

Alex Graves

James Goodnight

Andrew Ng

Fei-Fei Li

Alex Krizhevsky

Ilya Sutskever

Oriol Vinyals

Quoc V. Le

Ian Goodfellow

Demis Hassabis

David Silver

Andrej Karpathy

Ashish Vaswani

Noam Shazeer

Aidan Gomez

John Schulman

Mustafa Suleyman

Jan Leike

Daniel Kokotajlo

François Chollet

Neural Turing machine

Differentiable neural computer

Transformer Vision transformer (ViT)

Vision transformer (ViT)

Recurrent neural network (RNN)

Long short-term memory (LSTM)

Gated recurrent unit (GRU)

Echo state network

Multilayer perceptron (MLP)

Convolutional neural network (CNN)

Residual neural network (RNN)

Highway network

Mamba

Autoencoder

Variational autoencoder (VAE)

Generative adversarial network (GAN)

Graph neural network (GNN)

Category