Reasoning LLMs

With a deep dive into DeepSeek-R1

Dr. Nimrita Koul

Agenda

1. Introduction to Reasoning in LLMs
2. Techniques for improving reasoning in LLMs
3. DeepSeek R1: Architecture, Features, Performance
4. Applications and Future Directions
5. Hands-on DeepSeek R1

Reasoning

- Reasoning is the ability to draw conclusions based on facts, rules or evidence.

- Traditional reasoning types humans use:
  - Deductive – "All humans are mortals, I am a human, so I am mortal. "
  - Inductive – Inferring general patterns from specific examples. " The sun has risen in the east every day I have observed. The sun will rise in the east tomorrow as well. "
  - Abductive- Making educated guesses e.g. diagnosis from symptoms.
  - Probabilistic – Bayesian Inference.

Source Credits: https://www.funwithpuzzles.com

Logical Reasoning

- A farmer is going to the market with a wolf, a goat, and a cabbage. He comes to a river with a small boat that can carry only him and one of the three items at a time. If left alone together, the wolf will eat the goat, and the goat will eat the cabbage. How can the farmer get everything safely across the river?

Mathematical Reasoning

- A shopkeeper sells an item at a 20% profit. If the cost price of the item is increased by ₹50 and the selling price remains the same, the profit percentage drops to 10%. What is the original cost price of the item?

AI Reasoning

- AI Algorithms can reason using **strict logic rules** (rule-based systems)or using **flexible pattern recognition** from input data (neural networks).

- Reasoning models can accurately solve verifiable tasks—*such as math, logic puzzles and coding tasks*.

- Models like Anthropic's Claude 3.7 Sonnet, OpenAI's o1, o3, and DeepSeek's DeepSeek- R1 are reasoning LLMs.

- Such reasoning LLMs are built by finetuning large base models using Reinforcement Learning.

Training Standard LLMs using Supervised Fine Tuning

- Pretraining
- Supervised Finetuning (SFT)
- Reinforcement Learning through Human Feedback (RLHF)

Inference Time Strategies to build Reasoning Capabilities

Inference Time Strategies to build Reasoning Capabilities –

1. Chain of Thought Prompting and Decoding

Long CoT allows LLMs to generate more tokens and multiple outputs thus spending more computation power on a problem during inference.

Reasoning in AI Models with Long Chain of Thought Prompting

- With Long Chain of Thought (LongCoT) prompting LLMs is made to generate a series of intermediate reasoning steps before giving a final answer.

- The prompt “**Let’s think this through step by step**” enables reasoning.

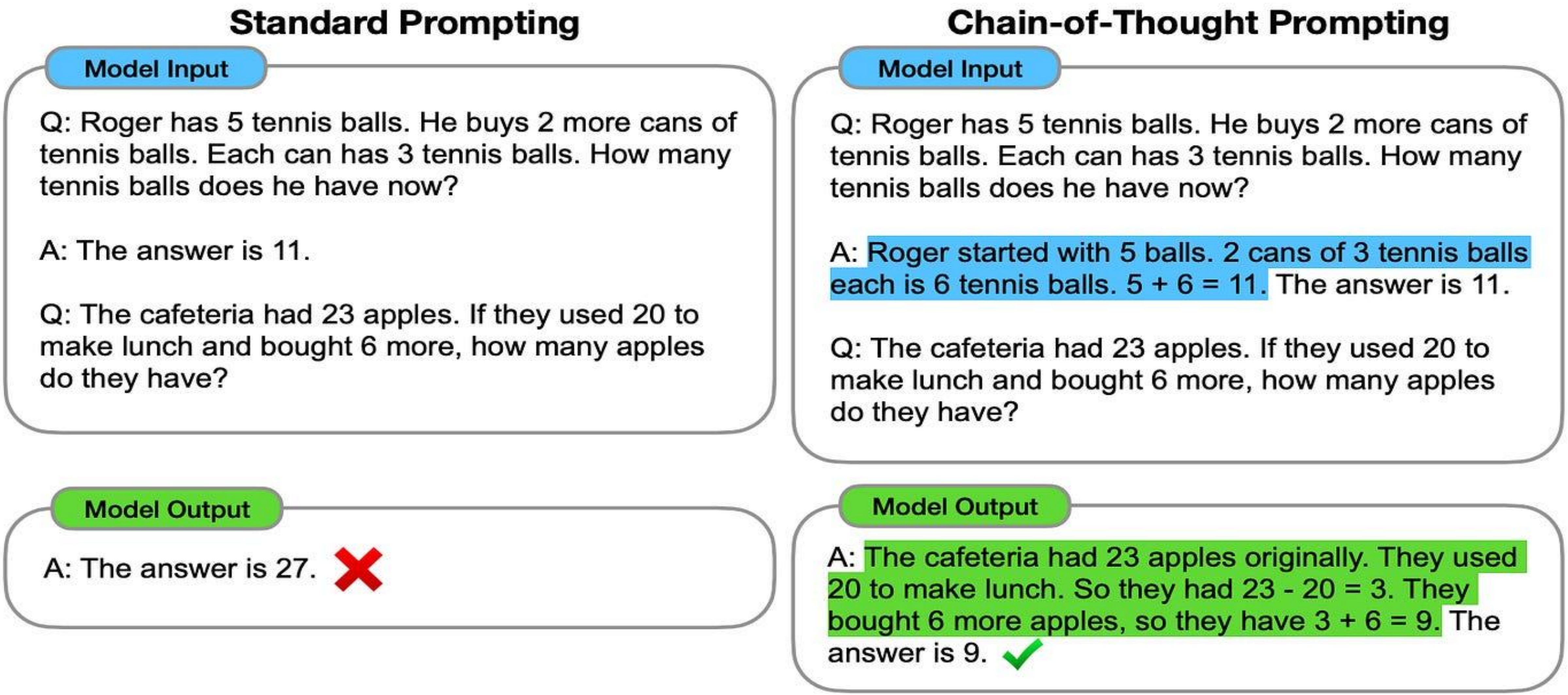


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Source Credits: Chain-of-Thought Prompting Elicits Reasoning in Large Language Modelshttps://arxiv.org/pdf/2201.11903

Long COT is not same as the standard CoT

- While a standard CoT is short and human-readable, a long CoT is several **thousand tokens** long and not very human readable.

- It is a long and complex reasoning trace behaviours

Farmer Solves River Crossing Puzzle Safely

A farmer is going to the market with a wolf, a goat, and a cabbage. He comes to a river with a small boat that can carry only him and one of the three items at a time. If left alone together, the wolf will eat the goat, and the goat will eat the cabbage. How can the farmer get everything safely across the river?

Thought for 275 seconds

Okay, so there's this classic puzzle about a farmer who needs to get a wolf, a goat, and a cabbage across a river. The boat can only carry the farmer and one item at a time. The challenge is that if the wolf and goat are left alone, the wolf will eat the goat, and if the goat and cabbage are left alone, the goat will eat the cabbage. The goal is to figure out a sequence of crossings that gets all three items across safely without any of them being eaten. Hmm, let me think through this step by step.

First, let me try to visualize the problem. The farmer starts on the left bank of the river with the wolf, goat, and cabbage. The boat is small, so each trip can only carry the farmer and one item. So, the farmer can take one item across, leave it on the right bank, and return alone. But he has to be careful about which items are left alone on either bank.

The main constraints are:

1. Wolf and goat can't be left alone on the same bank.

2. Goat and cabbage can't be left alone on the same bank.

So, the farmer needs to manage these constraints while moving everything across. Let's break it down.

First, if the farmer takes the wolf across first. So, he leaves the goat and cabbage on the left bank. But wait, if he takes

Farmer Solves River Crossing Puzzle Safely

- Left: Goat, Farmer
- Right: Wolf, Cabbage

7. **Take the goat to the right bank.**
   - All items and farmer are now safely on the right bank.

Final Answer:
The farmer should first take the goat across, return alone, take the wolf across, bring the goat back, take the cabbage across, return alone, and finally take the goat across again. This sequence ensures nothing is eaten. The steps are:

1. Farmer and Goat → Right
2. Farmer ← Left
3. Farmer and Wolf → Right
4. Farmer and Goat ← Left
5. Farmer and Cabbage → Right
6. Farmer ← Left
7. Farmer and Goat → Right

\boxed{
\text{Farmer takes goat over, returns, takes wolf over, brings goat back, takes cabbage over, returns, and takes goat over.}
}

New chat

Message DeepSeek

DeepThink (R1) Search

Dr. Nimrita Koul

2. Parallel Decoding

- Instead of generating a single output with LLM, it **generates multiple outputs and aggregates** these outputs to form a single, final answer.

Good Reasoning Depends on Good Planning

- Effective reasoning requires a structured plan.

- AI planning is the search for a sequence of good actions to take to achieve a goal (maximize a reward).

Main Planning Techniques : Classical AI Planning

1. Classical AI Planning

Classical planning solves problems by **searching for an action sequence that transforms an initial state into a goal state**.

- However, **classical AI planning depends on a correct action models,** it may not be appropriate for LLMs

Main Planning Techniques: Reinforcement Learning

2. **Reinforcement Learning for Planning in LLMs**:

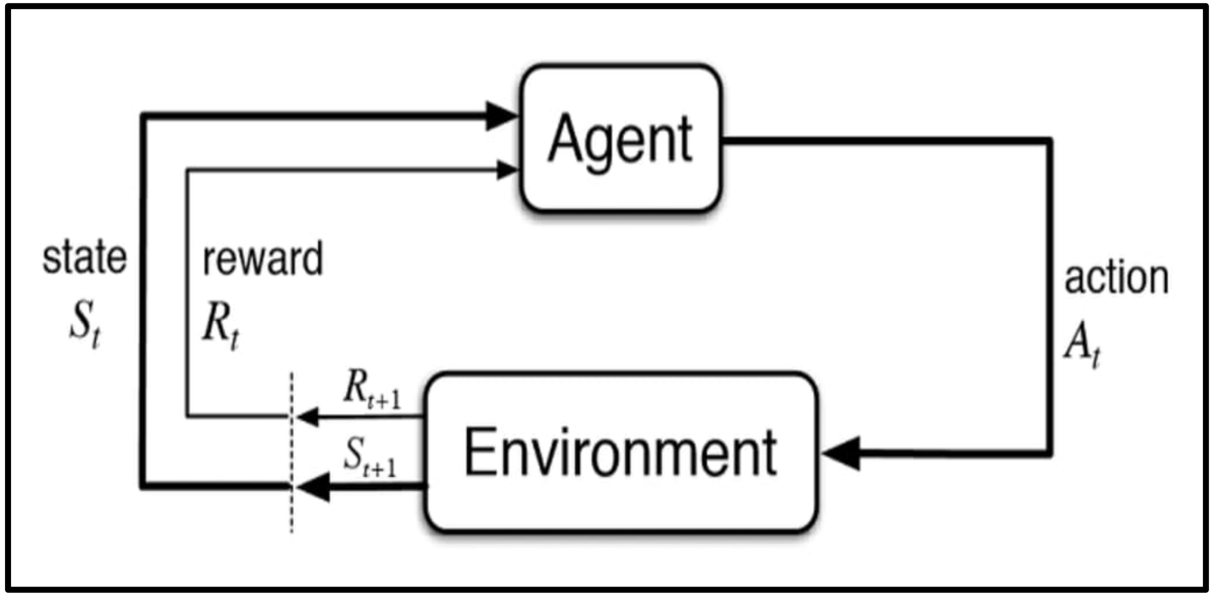
- An AI Agent can use RL to **learn a correct sequence of decisions** to take by interacting with an environment and receiving feedback in the form of rewards.

- Over time, the agent learns a **policy that maximizes** cumulative reward (generates best LLM output).

- DeepSeek demonstrated that reinforcement learning can drive complex reasoning improvements in AI without requiring vast supervised datasets.

Reinforcement Learning

Learning From Feedback

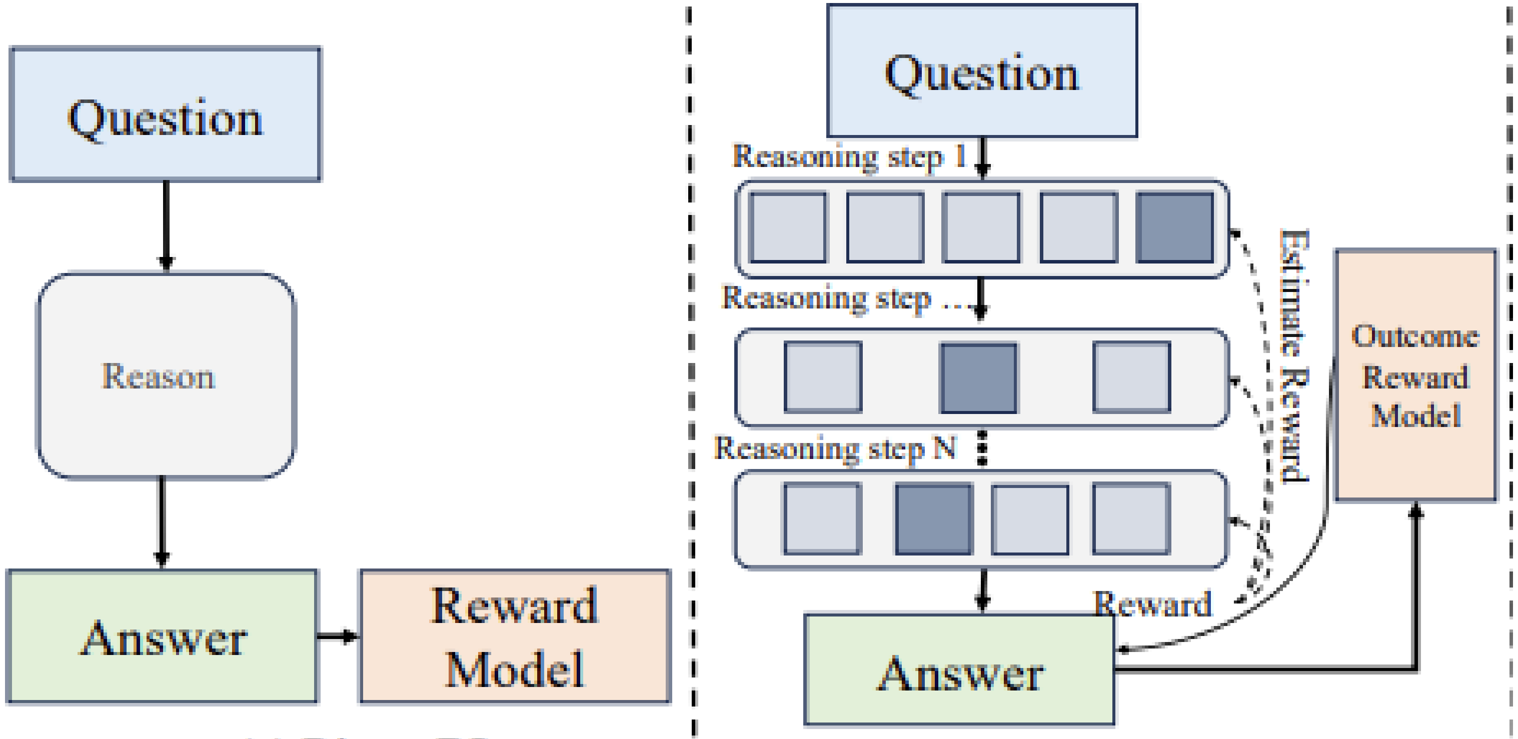


Source Credits: Sutton, R. S. and Barto, A. G. Introduction to Reinforcement Learning

In the context of LLMs, RL is used in many forms –

- **RLHF** to fine tune a model
- RL to train **an AI agent that uses an LLM** as part of its decision process.
- An **LLM agent acting in a simulated environment** (say, a text-based game or a web navigation task) can be improved via RL by trying actions, seeing outcomes, and learning a policy.

However, pure **RL can be sample-inefficient** (requiring many trials) and lacks guarantees of optimality. Modern techniques like model-based RL explicitly learn a model of the environment and plan within it, **blending classical planning ideas with reinforcement learning**.



(a) Direct RL
(b) Multi-Step RL With ORM

Source Credits: https://arxiv.org/pdf/2501.09686

Summary: Techniques to Enhance Reasoning Capabilities of Models

- Chain-of-Thought (CoT) Prompting and Integration
- Enhanced RLHF Integration
- Deliberative Alignment
- Train on Math and Code Combined to Enhance Model Reasoning
- Reinforcement Learning-Based Reasoning
- Distillation of Large Model Reasoning into Smaller Models

DeepSeek R1 Model Architecture

DeepSeek R1 is based on DeepSeek-V3

- DeepSeek –R1-Zero and DeepSeek-R1 reasoning models are based on a powerful, open-source base model called DeepSeek-v3

- DeepSeek-v3 is a 671 Billion parameter Mixture-of-Experts (MoE) open-source model .

- The total size of DeepSeek-V3 models on Hugging Face is 685B, which includes 671B of the Main Model weights and 14B of the Multi-Token Prediction (MTP) Module weights.

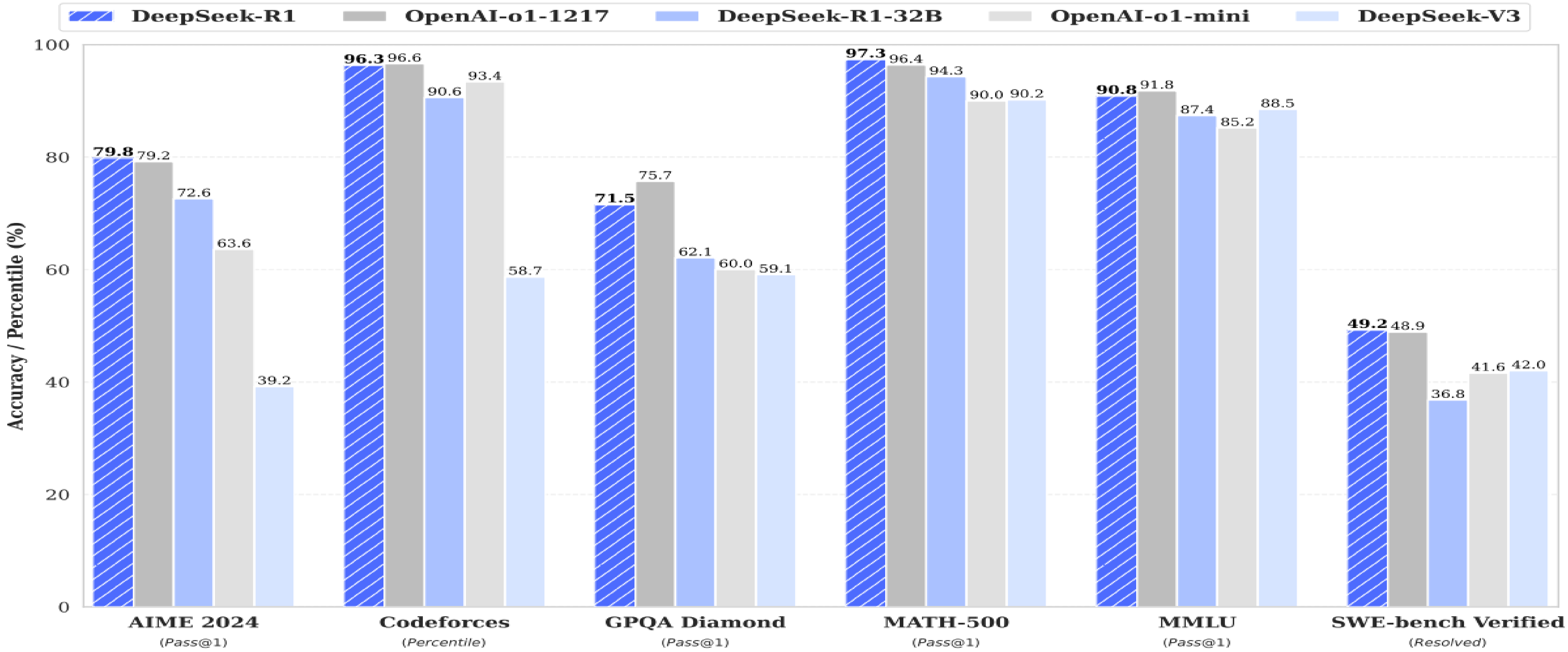
- DeepSeek R1 matches the performance of OpenAI’s o1 model.

DeepSeek-R1-Zero

- DeepSeek-R1-Zero is the first reasoning model from DeepSeek. DeepSeek-R1-Zero's training strategy uses pure large-scale RL without Supervised Finetuning.

- The model self-evolves and learns to use long CoT to solve complex reasoning problems through RL

DeepSeek R1 Performance Comparison



Source Credits: https://arxiv.org/abs/2501.12948

DeepSeek-R1

- Though DeepSeek-R1-Zero developed great reasoning capabilities, its language capabilities were not that great.

- To solve these problems, DeepSeek proposed a new multi-stage training process which resulted in DeepSeek-R1.

DeepSeek R1-Zero Training

RL with Group Relative Policy Optimization (GRPO)

- To train DeepSeek-R1-Zero, the DeepSeek-v3 base model is finetuned directly using RL algorithm called **G**roup **R**elative **P**olicy **O**ptimization (GRPO). GRPO does not need a neural reward model (value function).

- DeepSeek-R1-Zero uses **a rules-based reward system**

- Training uses two types of rewards:
  - **Accuracy Reward** evaluates whether the model's response is correct.
  - **Format Reward** enforces a desired format on the model's output.

Group Relative Policy Optimization (GRPO).

- GRPO is an RL algo used to train reasoning LLMs.

- While PPO uses a neural reward model (value function) to estimate rewards, GPRO generates multiple responses for each input prompt and uses the **average reward of these groups of responses as a baseline**.

- Uses KL divergence between current policy and a reference policy (a SFT model) into the loss function to ensure stable updates to policy.

- GRPO needs less memory, compute, is stable and scalable.

Reward Computation in GRPO

1. Given a single input, model generates a group/batch of outputs

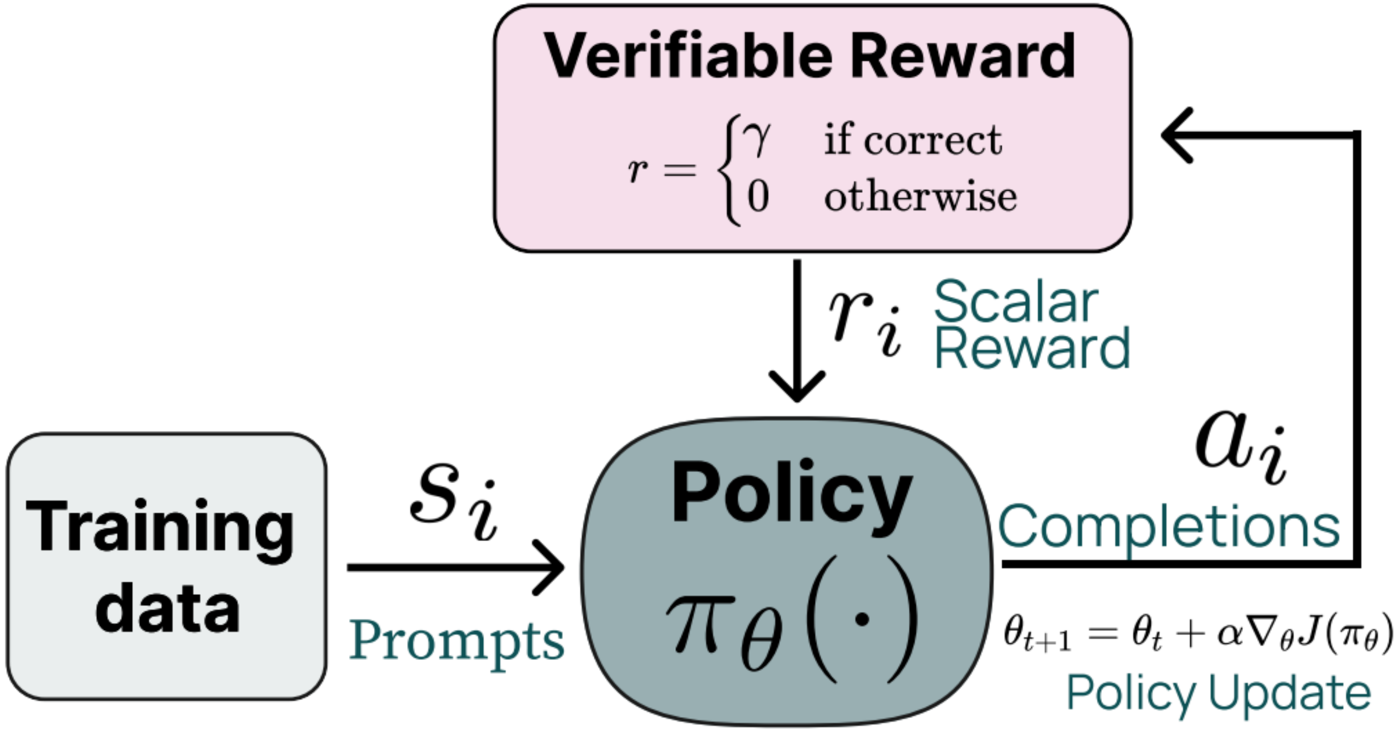
2. A **pretrained reward model trained on preference data** is used to assign a reward to each output

3. Unlike PPO which computes an absolute reward, GRPO computes **advantage of each response relative to other responses** in the same batch. The average reward for the batch acts as the baseline for advantage estimation.

4. Then GRPO updates the policy using these relative advantages ensuring the **responses with higher rewards are reinforced**. KL divergence is the part of objective function to keep the policy close to a reference SFT model.

Using RL to learn from Verifiable Rewards

- We can **directly use the verification result (result of string matching or code passing test cases) as a reward signal** for training with RL.
- This is the fundamental concept upon which all modern reasoning models are based.
- This can be implemented as process rewards or pure RL.



Source Credits: https://arxiv.org/pdf/2411.15124

Self-Evolution in DeepSeek-R1-Zero

- Model was not explicitly taught to decompose problems, search for a solution, perform backtracking or evaluate its own line of thought.

- The LLM autonomously learned necessary behaviours for solving problems via an RL-based "self-evolution" process.

From DeepSeek-R1-Zero to DeepSeek-R1: DeepSeek-R1's Multistage Training Pipeline

DeepSeek-R1 vs DeepSeek-R1-Zero

- DeepSeek-R1 differs from DeepSeek-R1-Zero in the addition of a **language consistency reward** which is calculated as the portion of the models output written in the desired target language.

- i.e. While DeepSeek-R1-Zero is good at reasoning, DeepSeek-R1 is good at reasoning as well as at generating natural language.

DeepSeek-R1's Multistage Training Pipeline

- Like DeepSeek-R1-Zero, DeepSeek-R1 begins with DeepSeek-v3 as a base model.

- Then, DeepSeek-R1 undergoes four stages of training – 2 steps of SFT and 2 steps of RL

Stage One: SFT on Cold Start Data

- During this stage, R1 is trained on a small data set of pairs of prompts and the summaries of their corresponding long CoT reasoning (Supervised Cold Start Data)

Stage Two: Reasoning Oriented RL

- In stage 2, large scale RL is used with additional reward for language consistency.

- This language consistency reward improves the **overall alignment of the resulting model with human preferences** – making it more fluent and readable.

Stage Three: SFT on Diverse Data Examples

- In stage 3, another level of SFT is carried out on more diverse dataset.

- The SFT dataset for the stage 3contained 600K prompt-reasoning trails plus 200K non-reasoning prompt-response pairs.

Stage Four: General-purpose RLHF

- In stage 4, RLHF is performed to align the model with human preferences.

Architectural Innovations in DeepSeek Models

Innovations in DeepSeek V3 that make it very economical

- Use of **multi-headed latent attention for better memory footprint**

- Use of **fine-grained and shared experts** in an optimized Mixture of Experts structure.

- Use **of multi-token prediction objective**

- **Innovative Load Balancing Strategy and Training Objective**

- Uses a **reduced FP8 precision** throughout training by using a new quantized training strategy.

Distilled Models from DeepSeek-R1

- To create distilled models – the authors of DeepSeek-R1 began with two base models ***Qwen-2.5*** *and* ***LLaMA-3***. Then the base models were trained via SFT over 800K supervised training examples curated in 3rd stage of training pipeline for DeepSeek-R1.

- 6 models were distilled from DeepSeek-R1 based on Qwen-2.5 and LLaMA-3 - (DeepSeek-R1 1.5B, DeepSeek-R1 7B, DeepSeek-R1 8B, DeepSeek-R1 14B, DeepSeek-R1 32B, DeepSeek-R1 70B)

Real-World Applications of Reasoning Models

Scientific Research & Problem Solving

- Assists in mathematical proofs, theorem validation.
- Generates hypotheses in physics and chemistry.

Automated Reasoning & Decision Support

- Legal analysis (contract review, legal argument structuring).
- Medical diagnosis assistance (differential diagnosis, symptom analysis).

Coding & Debugging

- Code generation with logical step validation.
- Error detection and reasoning-based fixes.

AI-Augmented Education

- Interactive tutors for logical reasoning and critical thinking.
- Step-by-step guidance in STEM education.

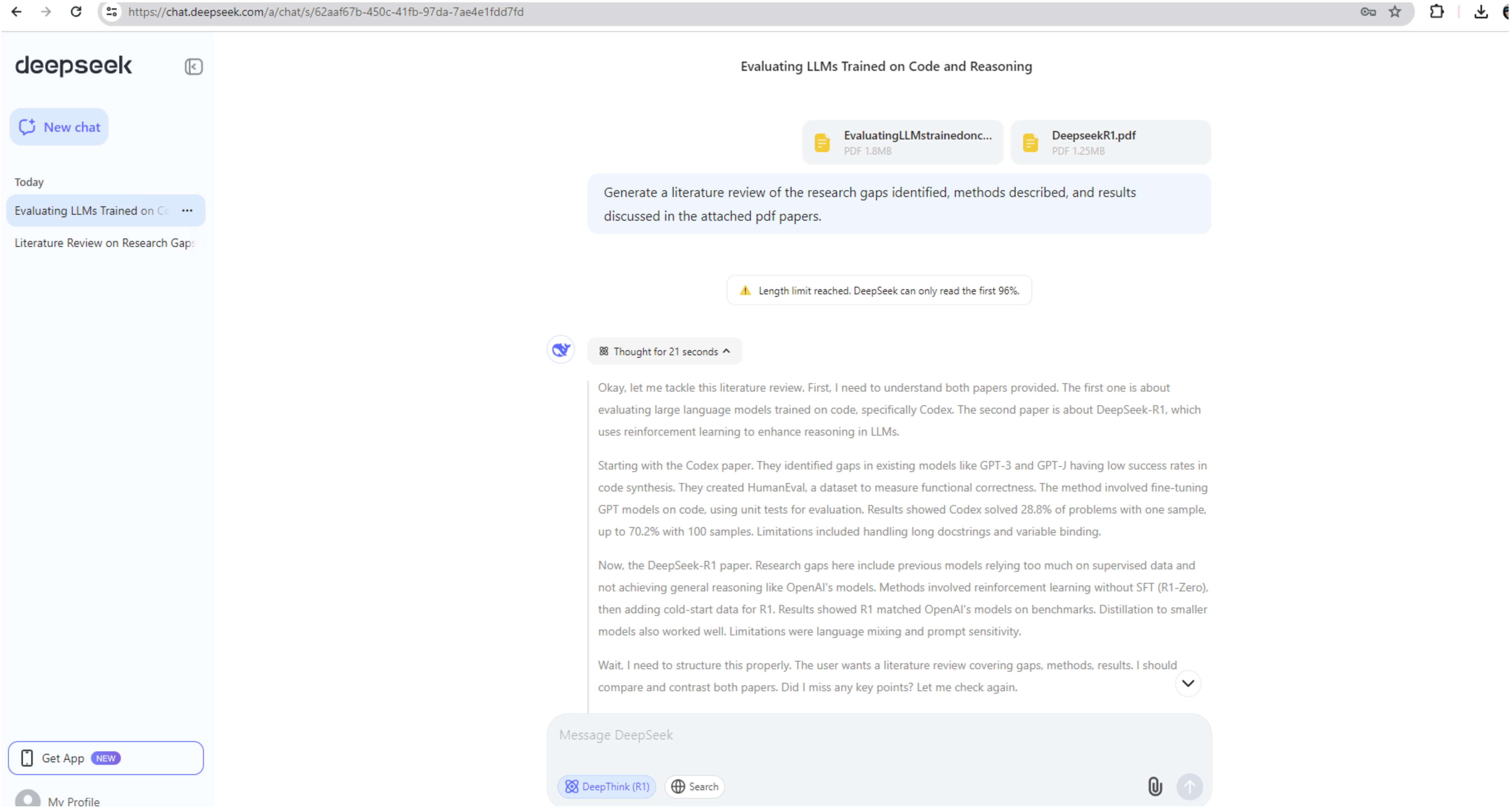
Emerging Trends in Reasoning LLMs

- **Fine-tuning, reinforcement learning, and test-time scaling to optimize LLM performance.**
- **Long CoT (and inference-time scaling):** Using more compute at inference time—*by generating a longer CoT*— can allow users to dynamically improve a model's reasoning capabilities.
- **Self-evolution through RL: Using** large-scale RL training, the LLMs can execute complex reasoning strategies within their long CoT without explicitly being trained on this objective.
- **Less human supervision:** Reasoning models rely less on human supervision as compared to standard LLMs. Rewards during RL training are derived primarily from rules-based systems.
- **Distillation:** With large and powerful reasoning models, we can distil the capabilities of these models into smaller, dense models using simple strategies.

Open Challenges in Reasoning Models

- Safety training for long CoT
- Balance between general / reasoning capabilities
- Optimal role of SFT in training reasoning models
- Minimizing "overthinking" in long CoT
- Efficient hosting of reasoning models

Hands On DeepSeek-1 using chat.deepseek.com

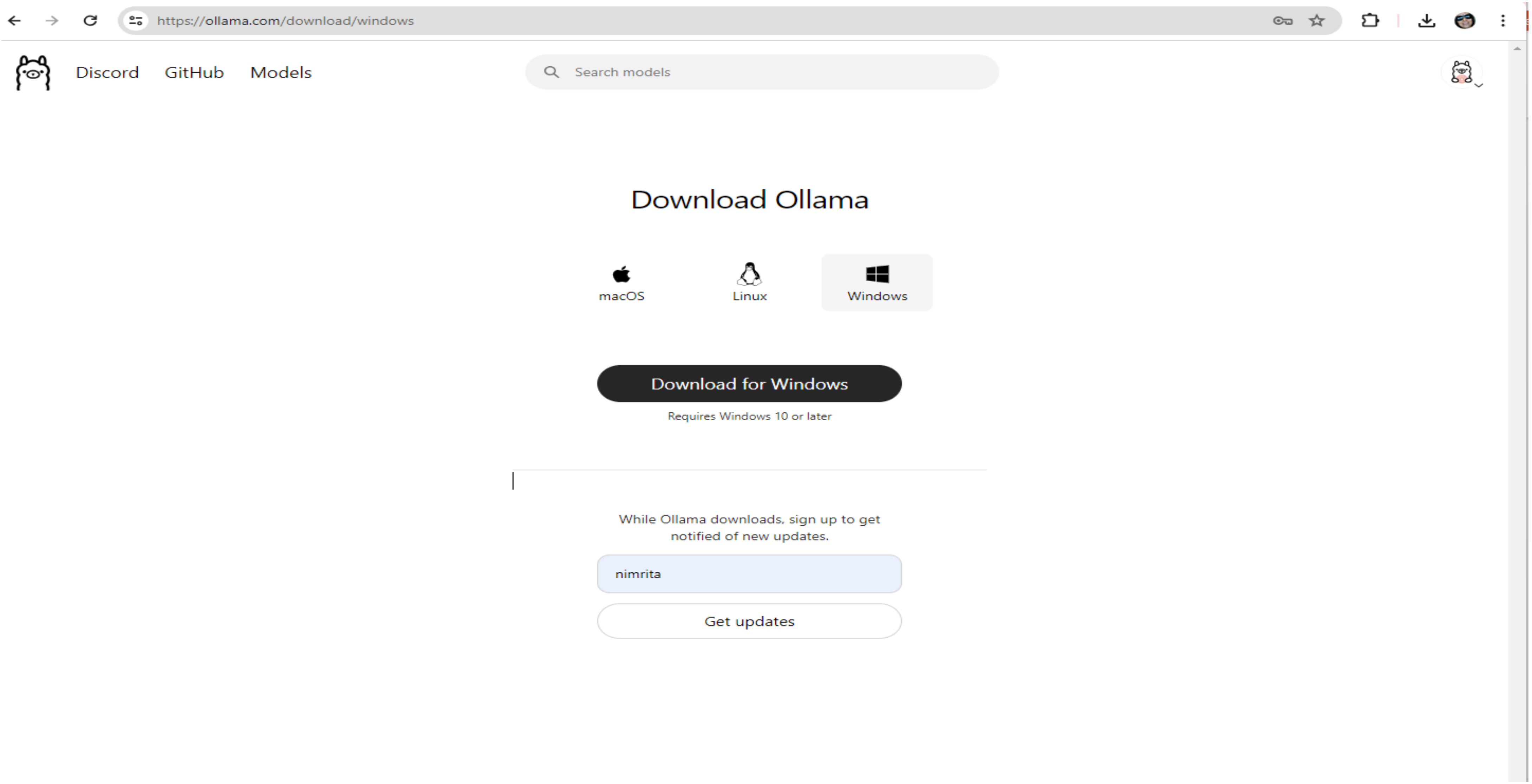


https://chat.deepseek.com

Dr. Nimrita Koul

Hands On DeepSeek-1 using Ollama

Download and Install Ollama for your machine



- Go to terminal and type the command:
  - **<u>ollama run deepseek-r1:1.5b</u>**

- Now you are chatting with deepseek-r1:1.5b distilled model locally.

- Use command “ollama list” to see all available models in ollama.

```
C:\Users\Nimrita Koul>ollama run deepseek-r1:1.5b
>>> if a > b and b > c, then is c < a?
<think>
First, I will examine the given inequalities.

The first inequality states that \( a \) is
greater than \( b \). This can be written as:
\[
a > b
\]

The second inequality indicates that \( b \) is
greater than \( c \), which can be expressed as:
\[
b > c
\]

To determine the relationship between \( a \)
and \( c \), I will combine these two
inequalities. Since \( a \) is greater than \( b
\) and \( b \) is greater than \( c \), it
logically follows that \( a \) must be greater
than \( c \). Therefore, we can conclude:
\[
a > c
\]
</think>

Sure, let's analyze the given inequalities step
by step.

**Given:**
1. \( a > b \)
2. \( b > c \)

**Step 1:** Understand the inequalities.

- The first inequality (\( a > b \)) means that
\( a \) is greater than \( b \).
- The second inequality (\( b > c \)) means that
\( b \) is greater than \( c \).

**Step 2:** Combine the inequalities.

Since \( a \) is greater than \( b \), and \( b
\) is greater than \( c \), we can infer the
```

Using DeepSeek-r1 in python using ollama

- First install “ollama” library in your environment. (Pip install ollama)
- Then open a new editor window and import ollama to use deepseek-r1

```
(pytorch_cuda) C:\Users\Nimrita Koul>python
Python 3.11.9 | packaged by conda-forge | (main, Apr 19 2024, 18:27:10) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import ollama
>>> response = ollama.chat(
... model = 'deepseek-r1:1.5b',
... messages=[{'role':'user','content':"what is the second law of thermodynamics"},],)
>>> print(response["message"]["content"])
<think>
Okay, so I need to figure out what the Second Law of Thermodynamics is. I remember hearing about two laws in my ph
class, but I'm not exactly sure what each one says. Let me think.

The First Law, or the Law of Conservation of Energy, makes sense because it's all about energy being conserved. Bu
 does that mean? It probably means that energy can't be created or destroyed, just transformed from one form to an
 That seems straightforward enough. So, if a battery is charging, some electrical potential energy is converted in
mical energy in the battery itself.

Now, onto the Second Law. I think it's about entropy and why certain processes are irreversible. Maybe it has some
to do with disorder? If you take a hot object and put it in a cold place, the heat from the hot object spreads out
 increasing the total disorder of the system. That makes sense because when you mix hot and cold drinks at a party
 don't just stay together; they spread out.
```

Thank you for your time.
Your feedback is valuable.

https://tinyurl.com/RLLMFeedback