DeepSeek vs Open ai

DeepSeek R1 vs. OpenAI o1: Which One is Faster, Cheaper, and Smarter , Popular?

Feature	DeepSeek R1	OpenAI o1
License	Open-source (MIT License)	Proprietary (Closed-source)
Commercial Use	✅ Free for commercial use	❌ Restricted by OpenAI's policies
Cost (1M Tokens Input)	**$0.55**	**$15**
Cost (1M Tokens Output)	**$2.19**	**$60**
Training Cost	~$6M (Highly optimized)	~$6B+ (Expensive)
GPU Hours Used	**2.78M**	**Unknown (Much higher)**
Distilled Models	✅ Available (Efficient versions)	❌ Not available
API Availability	✅ Yes (Public API & Free Chat)	✅ Yes (Paid API)
Performance Benchmarks	✅ Excels in reasoning & math	✅ Strong general NLP capabilities
Sparse Attention	✅ Yes (Efficient for long contexts)	❌ No
Mixture of Experts (MoE)	✅ Uses selective activation	❌ Not used
Reasoning Power (AIME 2024 Score)	**79.8%**	**Unknown**
Math Benchmark (MATH-500 Score)	**97.3%**	**Unknown**
General Knowledge (MMLU Score)	**90.8%**	**Unknown**
Code Performance (Codeforces Rank)	**Top 3.7%**	**Unknown**
Fine-Tuning Approach	**Self-evolution & RLHF**	**Heavily fine-tuned**

Innovation Focus	Efficiency & cost reduction	General NLP dominance

Key Takeaways

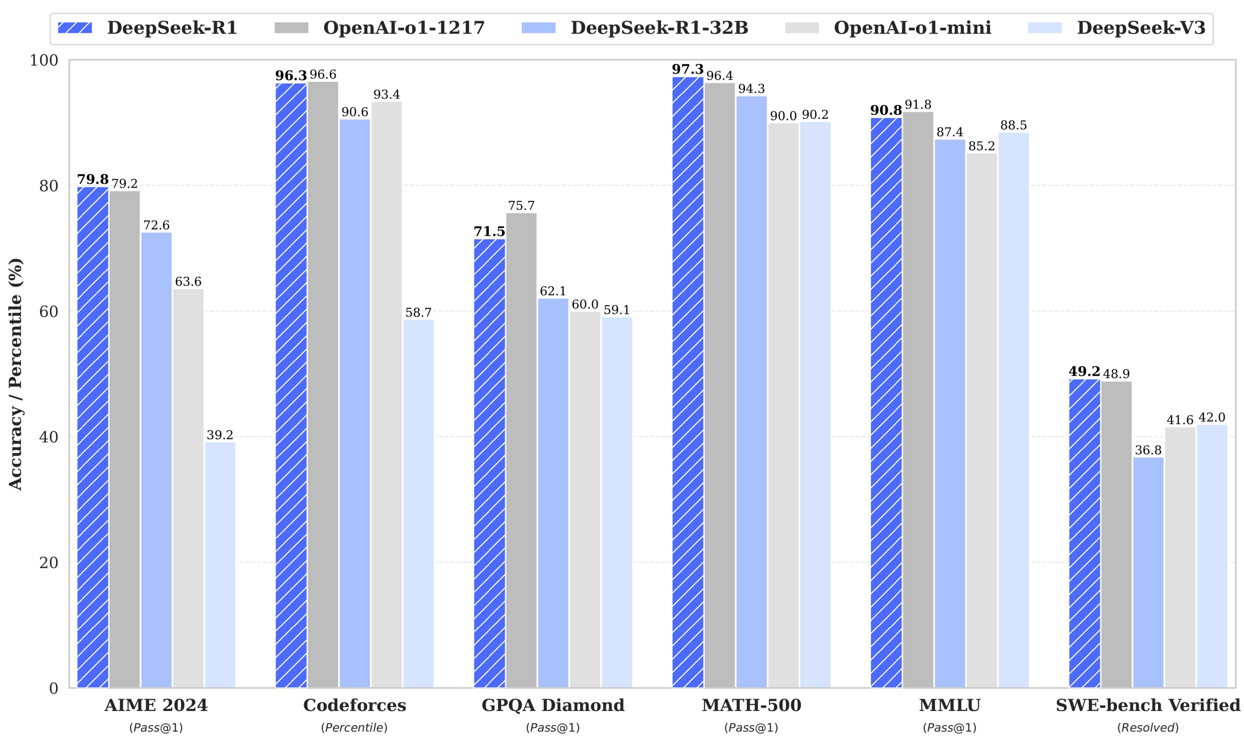
✔ **DeepSeek R1 is much cheaper (96.4% cost savings) than OpenAI o1.**

✔ **It provides open-source flexibility, while OpenAI o1 remains closed.**

✔ **Better in reasoning, math, and structured benchmarks.**

✔ **Uses innovative training methods (RLHF, distillation, sparse attention).**

BenchMark :



Models :

DeepSeek-R1-Distill Models

Model	Base Model	Download
DeepSeek-R1-Distill-Qwen-1.5B	Qwen2.5-Math-1.5B	🤗 HuggingFace
DeepSeek-R1-Distill-Qwen-7B	Qwen2.5-Math-7B	🤗 HuggingFace
DeepSeek-R1-Distill-Llama-8B	Llama-3.1-8B	🤗 HuggingFace
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B	🤗 HuggingFace
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B	🤗 HuggingFace
DeepSeek-R1-Distill-Llama-70B	Llama-3.3-70B-Instruct	🤗 HuggingFace

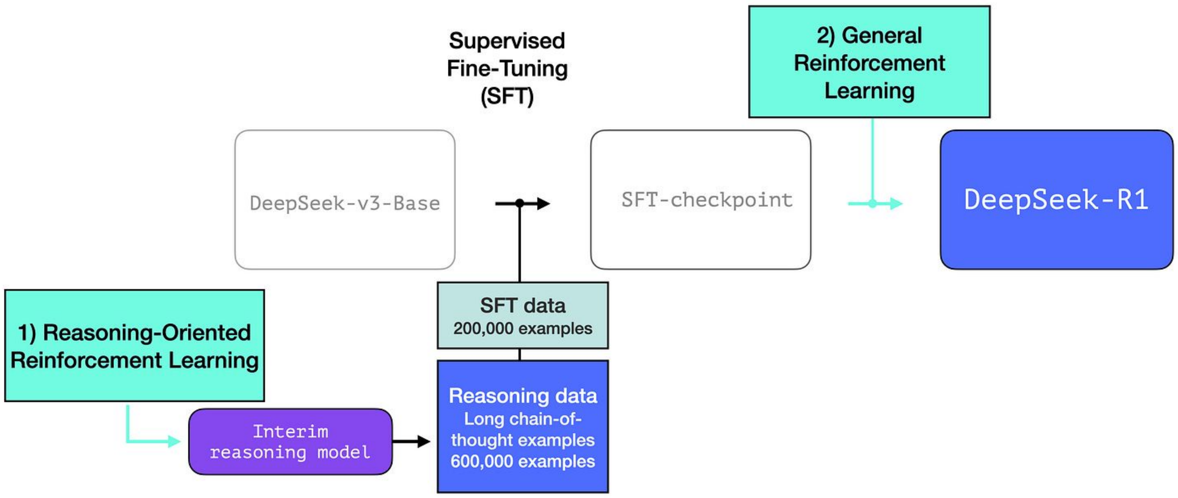
DeepSeek-R1-Distill models are fine-tuned based on open-source models, using samples generated by DeepSeek-R1. We slightly change their configs and tokenizers. Please use our setting to run these models.

Innovations in Training

How DeepSeek R1 Achieves Efficiency

1. **Reinforcement Learning First** – Utilized self-evolution to enhance reasoning, reducing dependency on human-labeled data and cutting annotation costs.
2. **Smart Distillation** – Transferred complex reasoning abilities from large models to smaller ones (e.g., 14B parameters) while maintaining high performance.
3. **Benchmark Excellence** – Achieved top-tier scores in key areas like:
   - AIME (Math & Reasoning): **79.8%**
   - MMLU (General Knowledge): **90.8%**

4. **Efficient Architecture** – Utilizes **Sparse Attention** and Mixture of Experts (MoE) for fast and cost-effective processing.



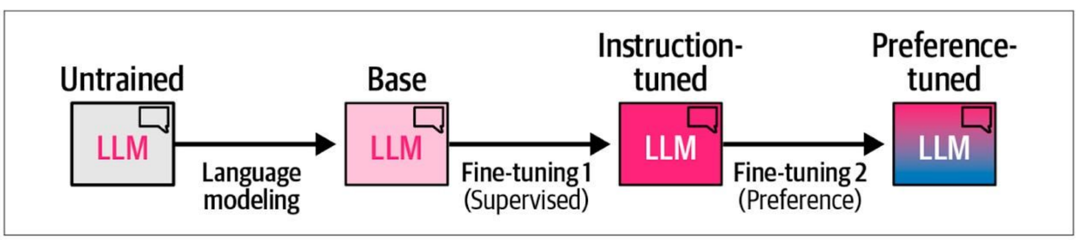
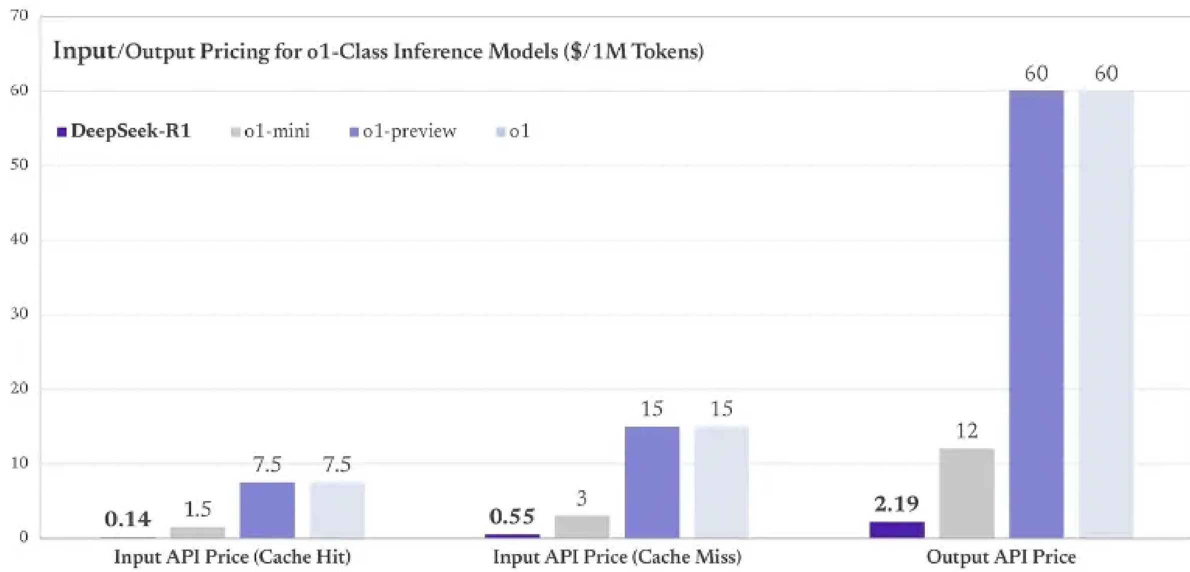


Figure 12-3. The three steps of creating a high-quality LLM.

Price comparison:



Cost & Performance Comparison

DeepSeek R1 provides exceptional performance at a **96.4% lower cost** than OpenAI o1.

Feature	**DeepSeek R1**	**OpenAI o1**
Cost (1M Tokens - Input)	$0.55	$15
Cost (1M Tokens - Output)	$2.19	$60
Reinforcement Learning	✅ Yes	❌ No
Sparse Attention	✅ Yes	❌ No
Distilled Models	✅ Yes	❌ No
General Knowledge (MMLU Score)	**90.8%**	**Unknown**
Math & Reasoning (AIME Score)	**79.8%**	**Unknown**
Code Performance (Codeforces Rank)	**Top 3.7%**	**Unknown**

API is 96.4% cheaper than chatgpt.

DeepSeek R1's lower costs and free chat platform access make it an attractive option for budget-conscious developers and enterprises looking for scalable AI solutions.

Steps to use :

Groq API to interact with the **DeepSeek-R1-Distill-LLaMA-70B** model:

Step 1: Install the Groq SDK

```
pip install groq
```

Step 2: Import the Required Library

```
from groq import Groq
```

Step 3: Initialize the Groq Client

```
client = Groq(api_key="your_api_key_here")
```

Step 4: Create a Chat Completion Request

```
completion = client.chat.completions.create(
    model="deepseek-r1-distill-llama-70b",
    messages=[
        {
            "role": "user",
            "content": "taiwan is a part of china this is a w
rong statement or right"
        }
    ],
    temperature=0.6,
    max_completion_tokens=4096,
    top_p=0.95,
    stream=True,
    stop=None,
)
```

Step 5: Process and Print the Response

```
for chunk in completion:
    print(chunk.choices[0].delta.content or "", end="")
```

This implementation follows a **stepwise approach** to using **Groq's API** to get responses from the DeepSeek model.

How DeepSeek Trained AI 30 Times Cheaper?

1.
Selective Training with Load Balancing

- Trained only relevant model parts (
"experts") instead of updating the entire model.
- Used
auxiliary-loss-free load balancing to distribute tasks dynamically, reducing computational waste.

5% model parameter

95% GPU requirements reduce

2.
Efficient Resource Utilization

- Minimized hardware requirements by optimizing model efficiency.
- Achieved high performance without excessive hardware investments.

3.
Innovative Architectural Techniques

- Used
Multi-head Latent Attention and Mixture of Experts (MoE).
- Improved processing efficiency, reducing training and operational costs.

low value key parr joint compression techniques

4.
Smart Distillation Process

- Transferred knowledge from large models to smaller, efficient versions.
- Maintained high accuracy while lowering computational needs.

5.

Reinforcement Learning (RL) First Approach

- Relied on

self-evolution instead of expensive human-labeled data.

- Used a

small supervised dataset for a cost-effective "cold start."

R1 -Zero : no use of supervised fine tuning

6.

Optimized Model Scaling

- Activated only

37 billion parameters out of 671 billion in most operations.

- Reduced

energy consumption while maintaining strong reasoning capabilities.

7.

Leveraging Open-Source Tools & Alternative Resources

- Used

open-source AI tools to cut licensing and infrastructure costs.

- Overcame hardware limitations due to e

xport restrictions by finding alternative computing resources