FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

# NI-NLM – Lecture 6

## Scaling laws, chain-of-thought, reasoning models

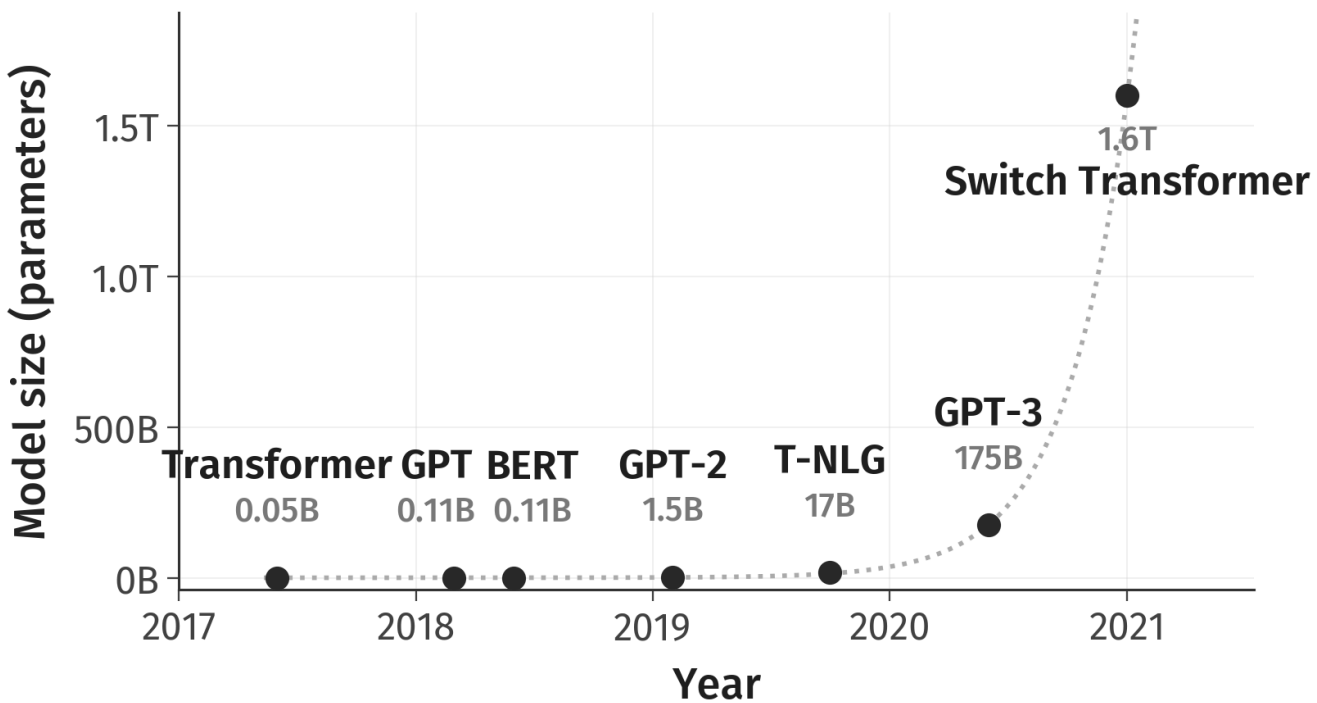
**Zdeněk Kasner**

24 Mar 2026

# Scaling laws

# LLMs getting exponentially bigger

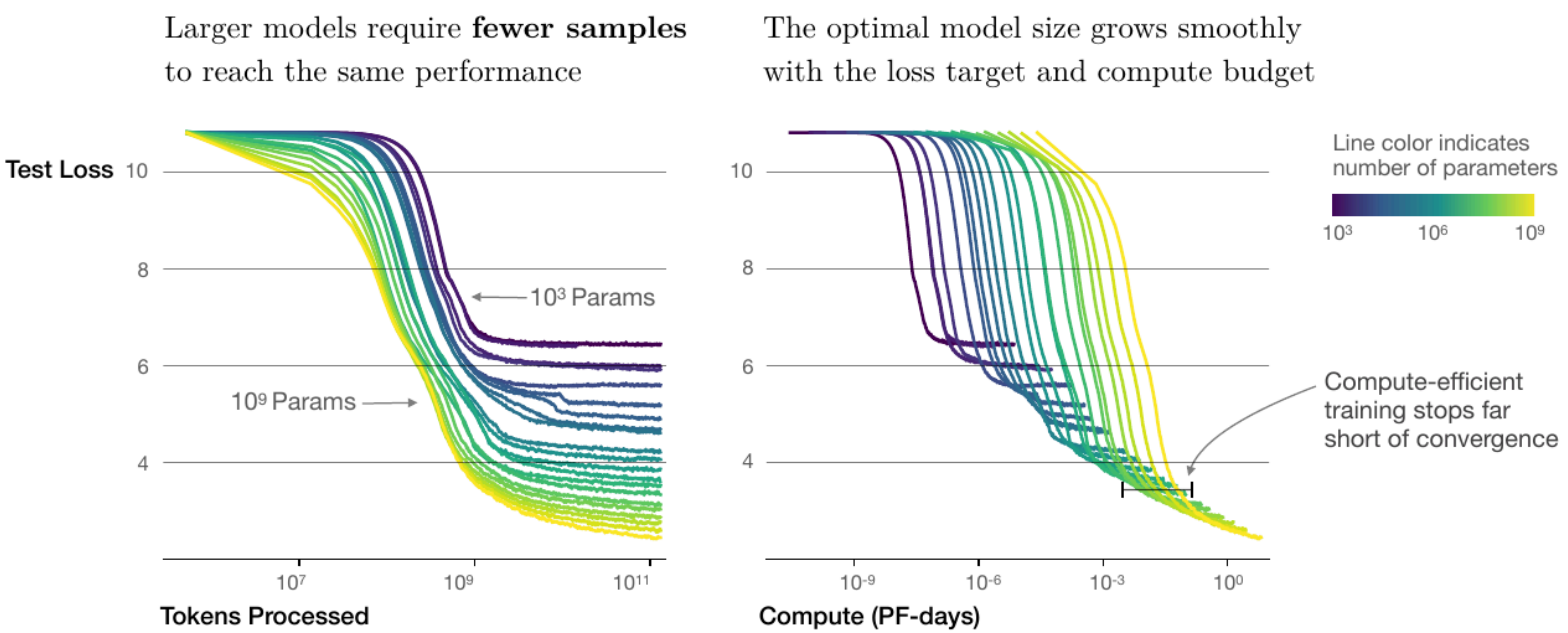
Until 2021, it seemed that the way to improve the models (lower their loss/perplexity on the training set) is **adding more parameters**:



# LLM scaling laws

source: Kaplan et al. (2020)

This trend was supported by the research from OpenAI (Kaplan et al., 2020), who showed that larger models **learn more efficiently and can attain lower loss**:



# LLM scaling laws

source: Kaplan et al. (2020)

According to Kaplan et al., test **loss** $L$ can be predicted solely based on $N$ (# of parameters), $D$ (# of training tokens), and $C$ (compute budget in FLOPs).

They derived the following **empirical laws**:

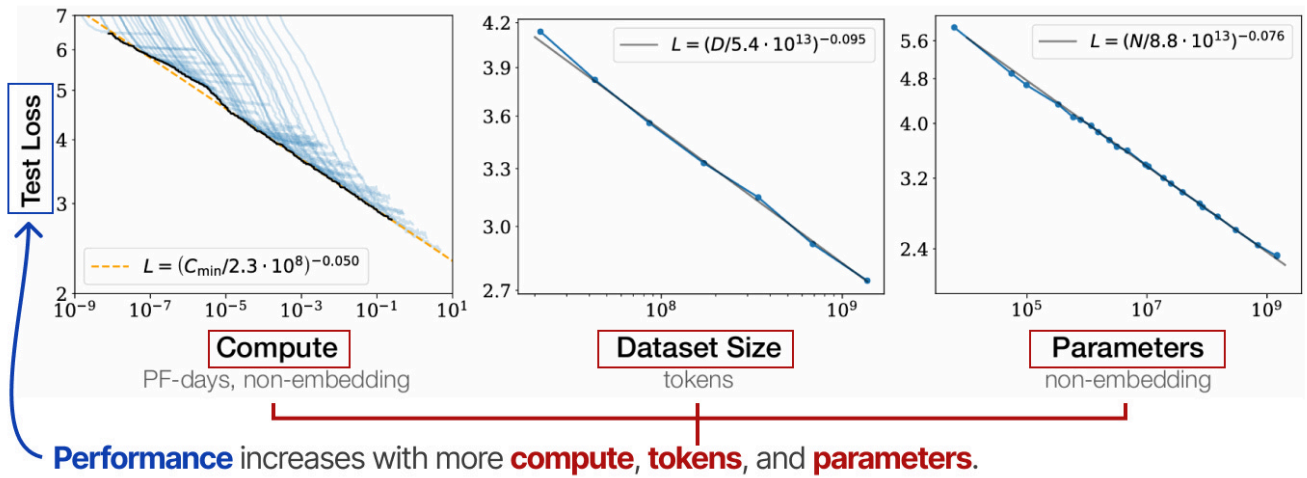
$$
\begin{array}{lll}
L(N) = (N_c \ / \ N)^{\alpha_N}; & \alpha_N \approx 0.076, & N_c \approx 8.8 \times 10^{13} \text{ parameters} \\
L(D) = (D_c \ / \ D)^{\alpha_D}; & \alpha_D \approx 0.095, & D_c \approx 5.4 \times 10^{13} \text{ tokens} \\
L(C) = (C_c \ / \ C)^{\alpha_C}; & \alpha_C \approx 0.050, & C_c \approx 3.1 \times 10^{8} \text{ PFLOP-days}
\end{array}
$$

**Example consequence**

Given a 10× increase in compute, Kaplan et al. suggest to increase model size ≈ 5.5× but data only ≈ 1.8×.

# LLM scaling laws

source: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-reasoning-llms




> *As the computational budget C increases, it should be spent primarily on larger models, without dramatic increases in training time or dataset size.*
>
> *— Kaplan et al. 2020*

→ These findings lead to large models like GPT-3 (175B) being trained on relatively limited data (≈ 300B tokens).

# LLM scaling laws: Chinchilla models

source: Hoffmann et al. (2022)

Research from DeepMind (Hoffmann et al., 2022) **challenged the scaling laws**.

They trained on a wider range of $N$ and $D$ and found that they should be scaled **equally** ($C$ is the compute budget):

$$N_{\text{opt}} \propto C^a, \qquad D_{\text{opt}} \propto C^b, \qquad a \approx b \approx 0.5$$

Their Chinchilla (70B, 1.4T tokens) model outperformed another model Gopher (280B, 300B tokens), having 4× fewer parameters.

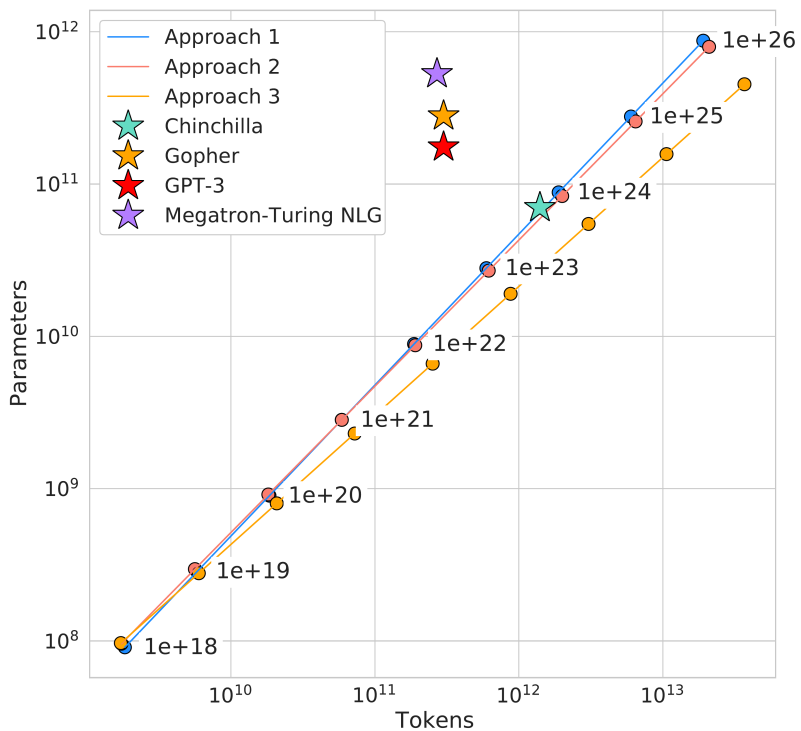
**Chinchilla scaling law**

For compute-optimal training, the number of training tokens should be $\approx$ 20× the number of parameters.

# LLM scaling laws: Chinchilla models

source: Hoffmann et al. (2022)

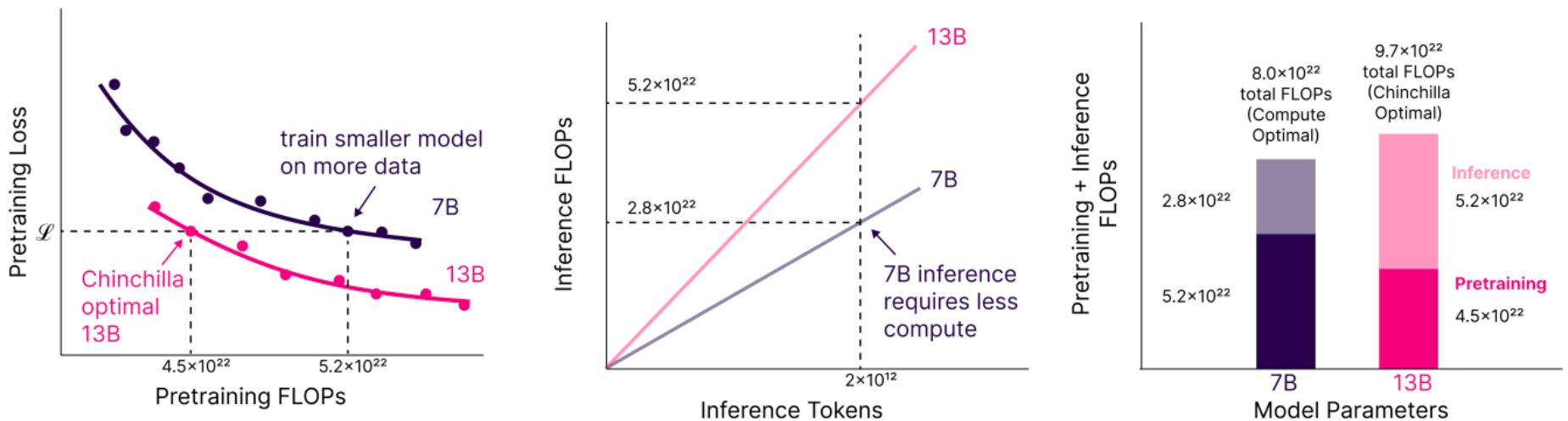
→ According to their findings, most existing LLMs at the time (GPT-3, Gopher, Megatron) were **undertrained**:



# Mind the inference

source: Sardana et al. (2024)

It may be also worth training **a smaller model on more data** to reduce inference cost (Sardana et al., 2024):




→ This was embraced by the Llama and Mistral models: **overtrain smaller models** so that they are cheaper at inference.

# The case of GPT-4.5

**Sam Altman** ✓
@sama

GPT-4.5 is ready!

good news: it is the first model that feels like talking to a thoughtful person to me. i have had several moments where i've sat back in my chair and been astonished at getting actually good advice from an AI.

bad news: it is a giant, expensive model. we really wanted to launch it to plus and pro at the same time, but we've been growing a lot and are out of GPUs. we will add tens of thousands of GPUs next week and roll it out to the plus tier then. (hundreds of thousands coming soon, and i'm pretty sure y'all will use every one we can rack up.)

this isn't how we want to operate, but it's hard to perfectly predict growth surges that lead to GPU shortages.

a heads up: this isn't a reasoning model and won't crush benchmarks. it's a different kind of intelligence and there's a magic to it i haven't felt before. really excited for people to try it!

Přeložit post

9:05 odp. · 27. 2. 2025 **5,5 mil.** Zobrazení

source: https://x.com/sama/status/1895203654103351462

| Model | Input Cost (per 1M tokens) | Output Cost (per 1M tokens) | Context Window | Comments |
|---|---|---|---|---|
| **GPT-4.5** | $75.00 | $150.00 | 128k tokens | Premium pricing for advanced emotional and conversational capabilities |
| **GPT-4o** | $2.50 | $10.00 | 128k tokens | Cost-effective baseline with fast, multimodal support |

source: dev.to

### 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` |

source: OpenAI

# What can we scale next?

Scaling **pretraining** has its limits:

- We are running out of high-quality text data (see Lecture 5).
- Training compute costs are enormous ($100M+ for frontier models).
- **Diminishing returns**: each 10× increase in compute yields smaller improvements.

**Question**

Can you think of another way to improve model performance?

**Idea**

"Squeeze out" more from the models during inference → **test-time scaling.**

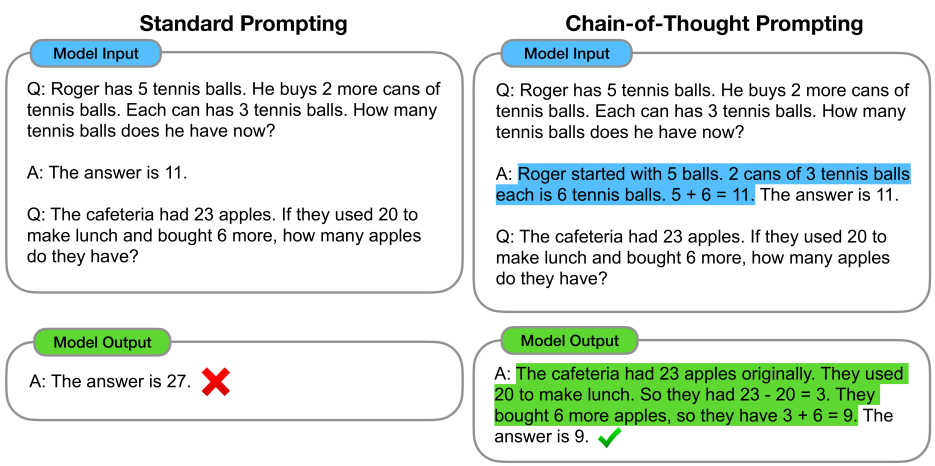
# Chain-of-thought prompting

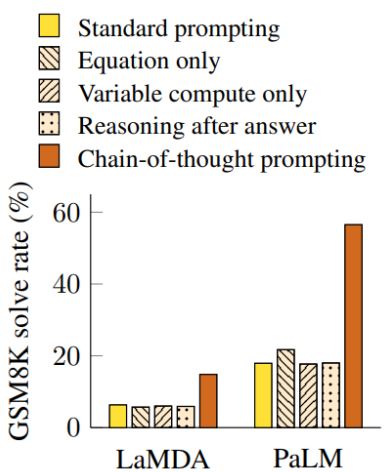
# Chain-of-thought (CoT) prompting

source: Wei et al. (2022)

**Idea**

Wei et al. (2022): LLMs struggle with math and multi-step reasoning. What if we showed them **how to do intermediate reasoning steps**?



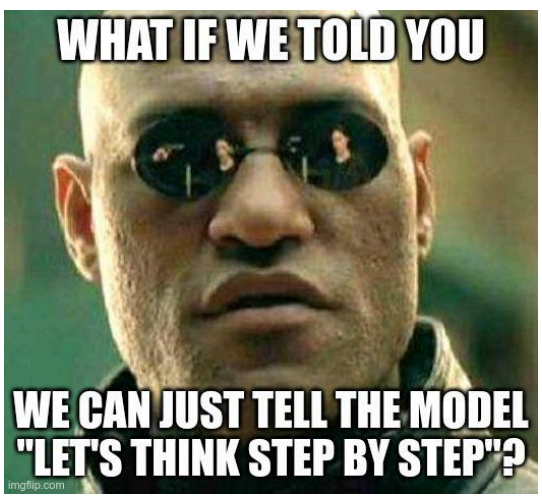

→ Dramatic improvement on arithmetic, commonsense, and symbolic reasoning.

# Zero-shot CoT: “Let’s think step by step”

source: Kojima et al. (2022)

**Idea**

Kojima et al. (2022): Writing out the steps is difficult. Do we need to do it at all?




(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

*(Output) The answer is 8.* ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

*(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls.* ***The answer is 4.*** ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

*(Output) 8* ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let’s think step by step.***

*(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

→ Prompting with **“Let’s think step by step”** is enough to trigger reasoning behavior.

# Chain-of-thought (CoT) prompting

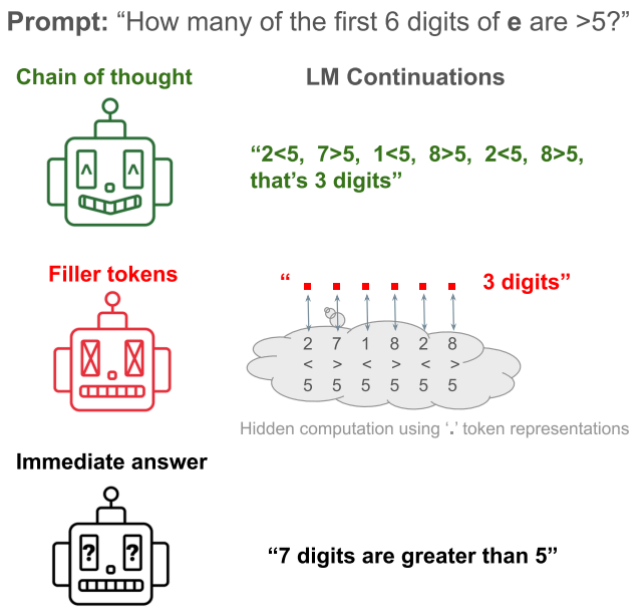
- Can be applied to any instruction-tuned model.
  - Does not require more than appending (a variant of) “Think step-by-step” to the prompt.
- Generally increases performance on problems requiring **multi-step reasoning**
  - Helps to break down complex problems into simpler subproblems.
- Nowadays falling out of favor compared to **large reasoning models** trained to reason explicitly.

# Why does CoT work?

source: Chu et al. (2023)

Several hypotheses:

- The "emergent" ability can be **learned from pretraining on code** data (Ma et al., 2024; Puerto et al., 2024).
- However, the model may be also just using **extended inference time** to perform more computation (Pfau et al., 2024).




> ⚠️ **CoT as an explanation?**
>
> Research also shows that CoT may **not** reflect what is happening inside models.

# Test-time scaling

# Test-time scaling

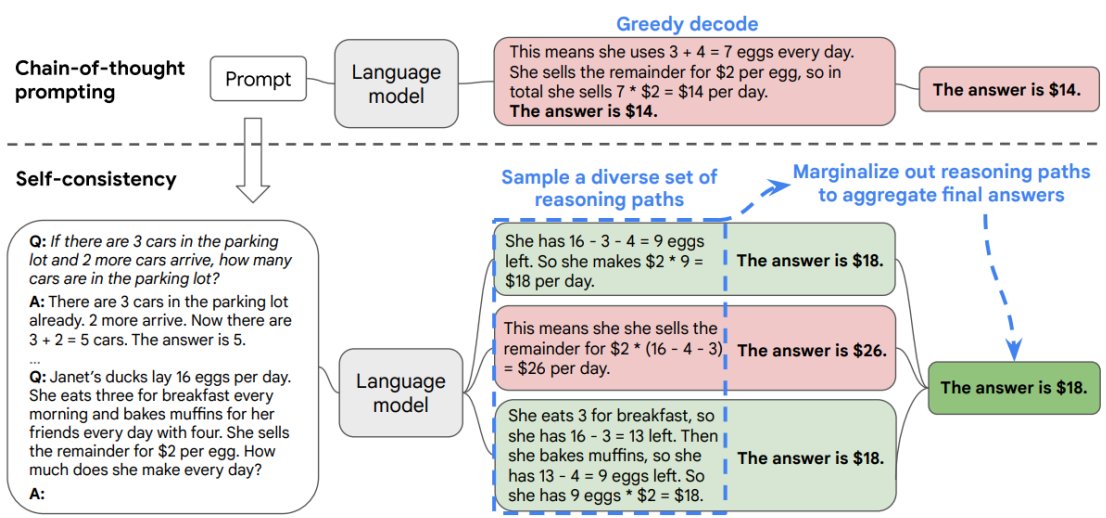
source: Wang et al. (2022)

## Idea: test-time scaling

Can we make use of the reasoning capability to improve model performance without further training, solely at **inference time**?

The simplest example of **test-time scaling**:

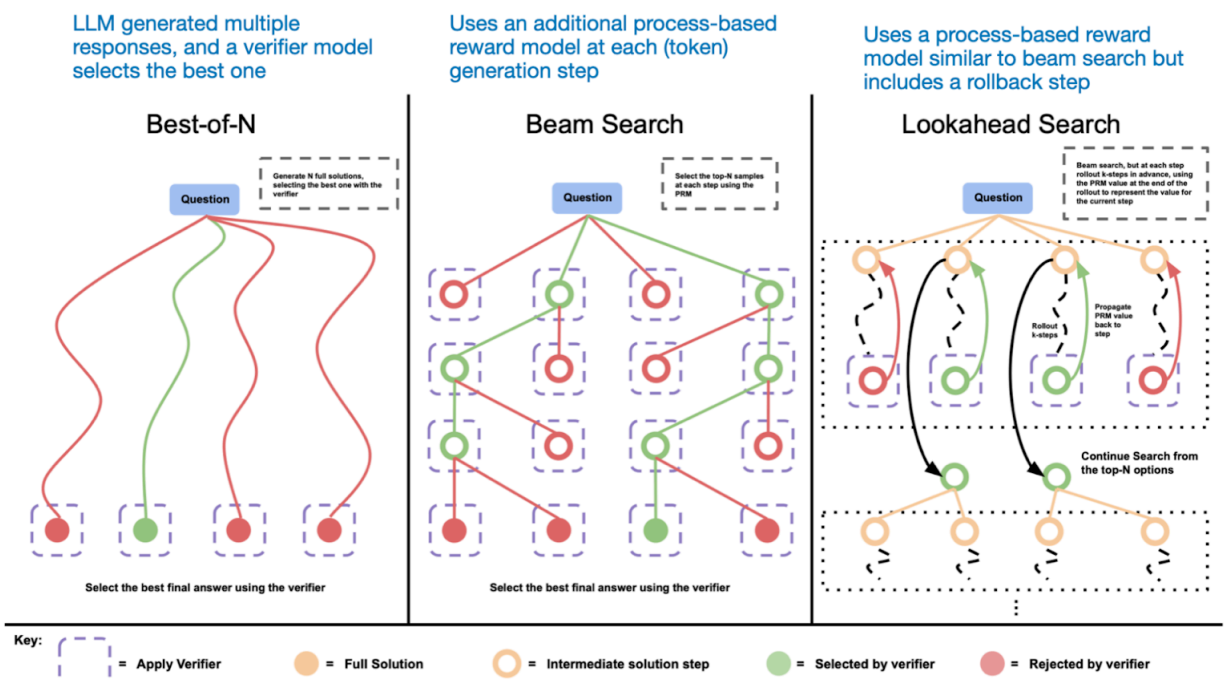
- Generate multiple CoT paths for the given problem.
- Use majority voting to select the final answer.



# Test-time scaling strategies

source: Snell et al. (2024)

More advanced extensions involve **tree search** and a verifying the solution with a **verifier model** (e.g., code interpreter):



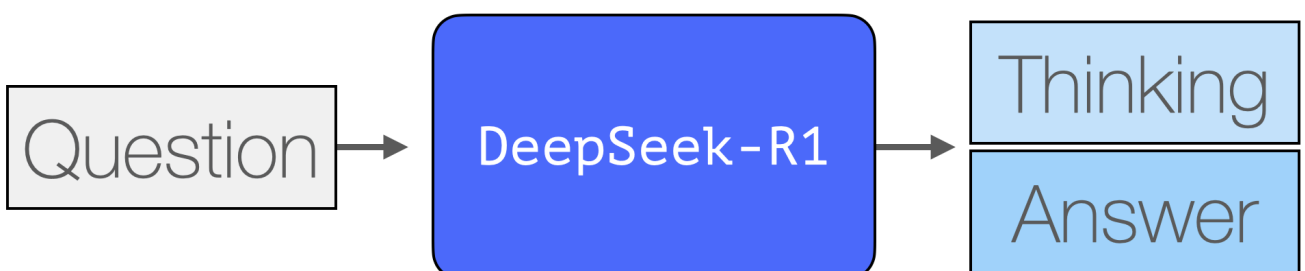
# Large reasoning models

# LLMs → LRMs

source: https://newsletter.languagemodels.co/p/the-illustrated-deepseek-r1

**Idea**

What if we trained the model to **always produce the reasoning trace** before answering?




We call such a model a **large reasoning model (LRM)**.

# LLMs with CoT → LRMs

**Question**

Why not just use the chain-of-thought prompting?

→ Chain-of-thought reasoning is **not a robust capability**. It is largely an artifact of training data (most likely code data, Ma et al., 2023).

→ It is also **too simple**. Ideally, we would like to automate test-time scaling. That means giving the model the ability to:

- Decompose the problem into subproblems.
- Follow multiple reasoning paths.
- Back-track from invalid paths.

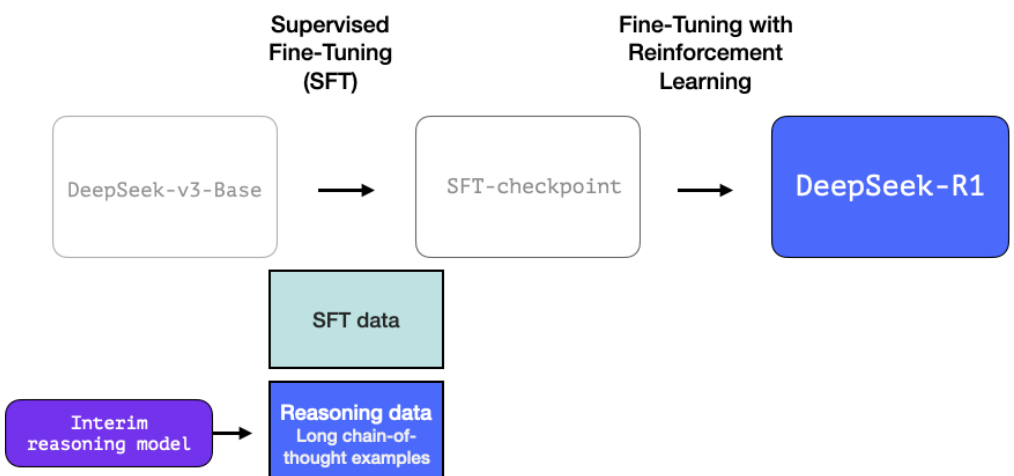
# DeepSeek-R1: overview

source: DeepSeek-AI (2025)

**DeepSeek-R1** (DeepSeek-AI, 2025): the first **open** reasoning model. 671B parameters, competitive with OpenAI o1.

Their recipe on how to build a strong reasoning model:

1. Take a **base model** (they took their `deepseek-v3-base`).
2. **Finetune** the model on a dataset of reasoning traces.
3. Improve the reasoning process using **reinforcement learning**.




source: Jay Alammar (2025)

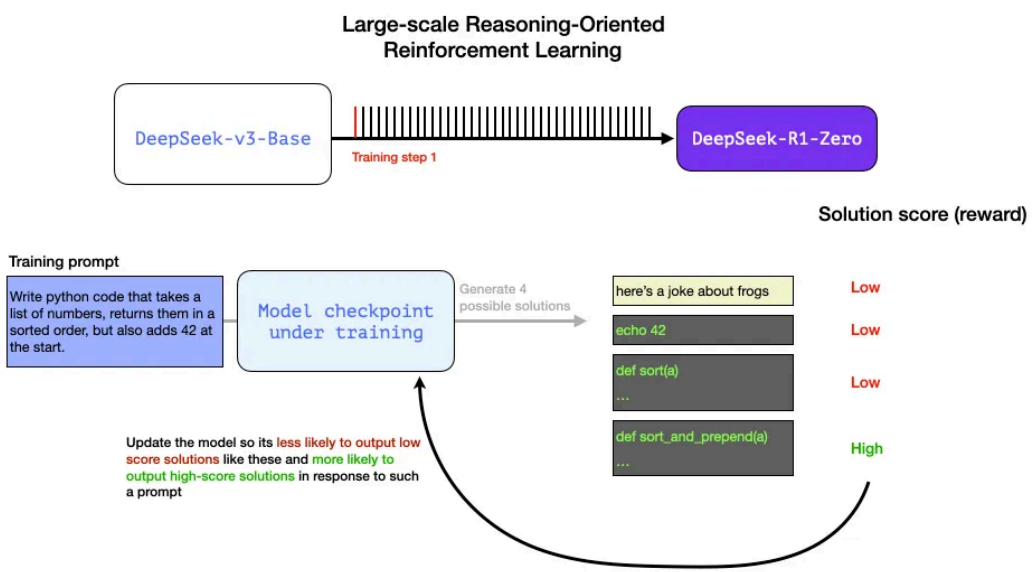
# DeepSeek-R1-Zero

**Question**

Where to get the dataset with reasoning traces for Step 2?

Recipe of DeepSeek-R1-Zero:

1. Apply the model with a **regular CoT prompting** on difficult problems.
2. Automatically **verify the solutions**.
3. **Reward** the model for good solutions with using RL.



# DeepSeek-R1-Zero

source: DeepSeek-AI (2025)

During RL training, the model **spontaneously** learns to re-evaluate its reasoning and try alternative approaches:

| |
|---|
| Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to |
| Response: <think><br>To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$<br>$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2$.<br>Rearrange to isolate the inner square root term:<br>$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$<br>$\ldots$<br>Wait, wait. Wait. That's an aha moment I can flag here.<br>Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$<br>We started with the equation:<br>$\sqrt{a - \sqrt{a + x}} = x$<br>First, let's square both sides:<br>$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$<br>Next, I could square both sides again, treating the equation: $\cdots$<br>$\ldots$ |

Table 3 | An interesting "aha moment" of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

# DeepSeek-R1-Zero: limitations

source: DeepSeek-AI (2025)

**Question**

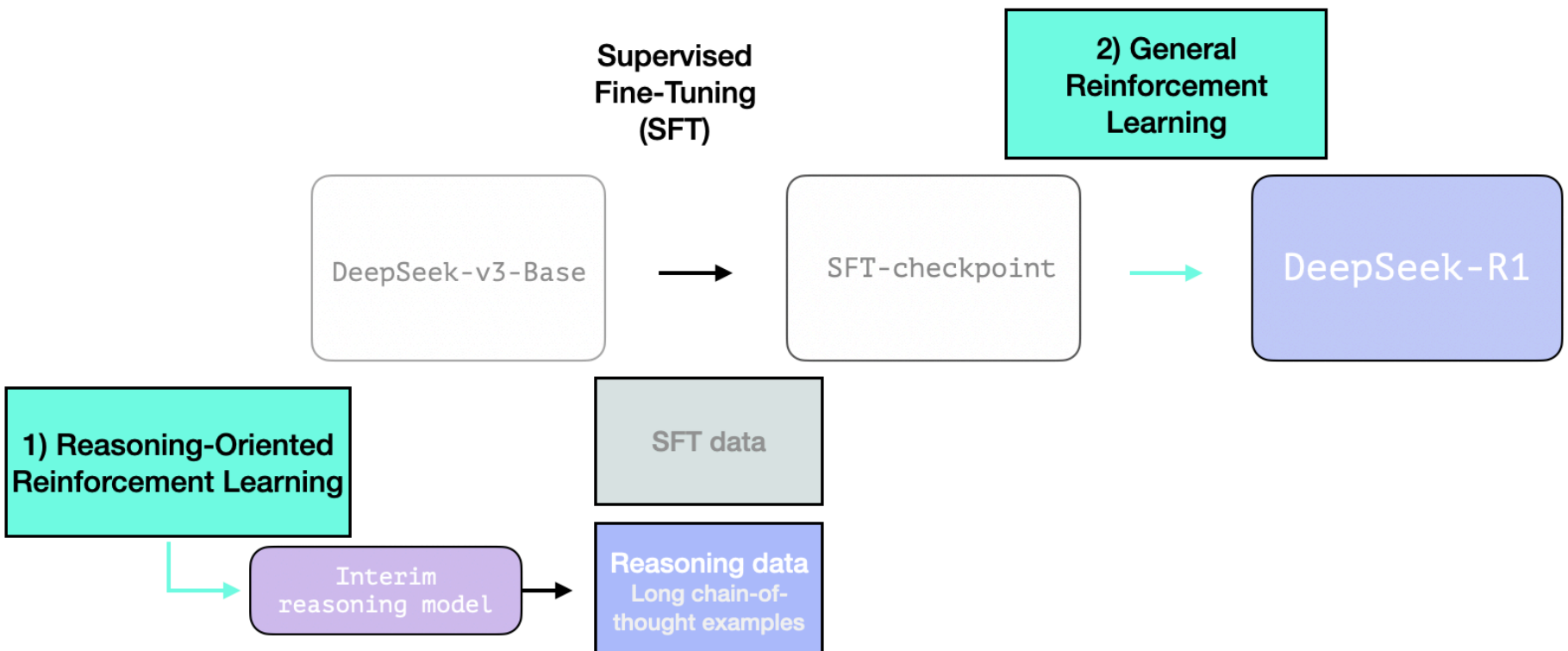
Can we just use DeepSeek-R1-Zero as *the* reasoning model?

Pure RL training has issues:

- Early RL training is unstable and hard to get going.
- The model mixes languages and produces messy formatting.
- The model gets good at math/code but struggles with general tasks.

→ However, we can still use the model to **generate a dataset of reasoning traces** for the finetuning step.

# DeepSeek-R1: Full training pipeline



# Distillation: smaller reasoning models

source: DeepSeek-AI (2025)

**Idea**

Now we have a *strong* reasoning model. Can we use its outputs to get a high-quality dataset of reasoning traces?

Yes: we can **finetune a model directly on the reasoning traces** of DeepSeek-R1-671B.

→ This idea lead to **distilled models** based on Llama and Qwen (between 1.5B to 70B).

**Why distillation?**

Distillation – in this context – means finetuning a smaller model ("student") on the reasoning traces generated by a larger model ("teacher").

# Summary: how to build a reasoning model

source: Raschka (2025)

## Option 1: Pure RL

- Base LLM + RL with verifier rewards
- Emergent reasoning
- Unstable, poor readability

→ DeepSeek-R1-Zero

## Option 2: SFT + RL

- SFT on reasoning traces, then RL
- Stable training
- Expensive

→ DeepSeek-R1, current frontier LRMs

## Option 3: Pure SFT

- SFT on reasoning traces distilled from the teacher model
- Cheapest approach
- Limited performance

→ distilled LRMs

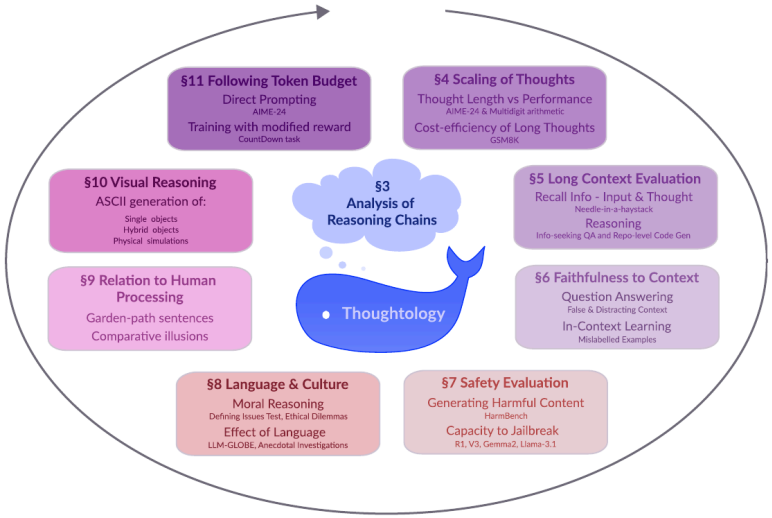
RL = reinforcement learning, SFT = supervised finetuning

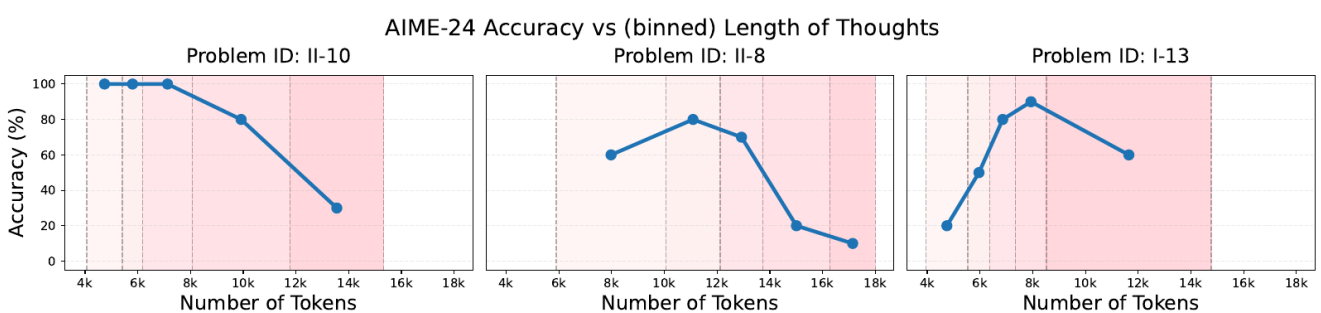
# “Thoughtology”

source: Marjanović et al. (2025)

Many properties of reasoning models are **yet to be properly investigated**, such as:

- How well does the trace reflect the internal thinking process?
- What is the optimal length of the trace? Can we enforce certain length?
- etc.



# When to use reasoning models

## Makes sense:

- Math and logic problems
- Complex coding tasks
- Multi-step planning

## A bit of an overkill:

- Simple Q&A
- Factual retrieval
- Text summarization

**⚠️ Why not use reasoning models for all the tasks**

Reasoning models use **significantly more tokens** (→ cost and time), possibly up to 10–100× the output length of a regular instruction-tuned model.

# Practical aspects

# What does reasoning look like in practice?

Reasoning models typically wrap their **thinking traces** in `<think>` tags:

```
User: How many r's are in "strawberry"?

<think>
Let me count the r's in "strawberry" letter by letter.
s-t-r-a-w-b-e-r-r-y
r at position 3 → count: 1
r at position 8 → count: 2
r at position 9 → count: 3
So there are 3 r's.
</think>

There are 3 r's in "strawberry".
```

The thinking part is typically hidden from the user in chat interfaces.

# Accessing reasoning in APIs

Open LLM APIs expose the reasoning content in a separate field.

Example: vLLM (https://docs.vllm.ai/en/latest/features/reasoning_outputs):

```
messages = [{"role": "user", "content": "9.11 and 9.8, which is greater?"}]
response = client.chat.completions.create(model=model, messages=messages)

reasoning = response.choices[0].message.reasoning
content = response.choices[0].message.content
```

**⚠ Warning**

Full reasoning traces are typically **not** available with commercial models.

For example, Google only offers so called “thinking summaries”: https://ai.google.dev/gemini-api/docs/thinking

# Where to get LRMs

source: vLLM docs

Now there is quite a lot of open LRMs to choose from:

| Model Series | Parser Name | Structured Output Support | Tool Calling |
|---|---|---|---|
| DeepSeek R1 series | `deepseek_r1` | `json`, `regex` | ❌ |
| DeepSeek-V3.1 | `deepseek_v3` | `json`, `regex` | ❌ |
| ERNIE-4.5-VL series | `ernie45` | `json`, `regex` | ❌ |
| ERNIE-4.5-21B-A3B-Thinking | `ernie45` | `json`, `regex` | ✅ |
| GLM-4.5 series | `glm45` | `json`, `regex` | ✅ |
| Holo2 series | `holo2` | `json`, `regex` | ✅ |
| Hunyuan A13B series | `hunyuan_a13b` | `json`, `regex` | ✅ |
| IBM Granite 3.2 language models | `granite` | ❌ | ❌ |
| MiniMax-M2 | `minimax_m2_append_think` | `json`, `regex` | ✅ |
| Qwen3 series | `qwen3` | `json`, `regex` | ✅ |
| QwQ-32B | `deepseek_r1` | `json`, `regex` | ✅ |

# LRMs and current frontiers

# LRMs and current frontiers

July 21, 2025 Research

## Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad

Thang Luong and Edward Lockhart

## OPENAI O3 BREAKTHROUGH HIGH SCORE ON ARC-AGI-PUB

OpenAI has released a new version of o3. Read our analysis to learn how it differs from the preview below.

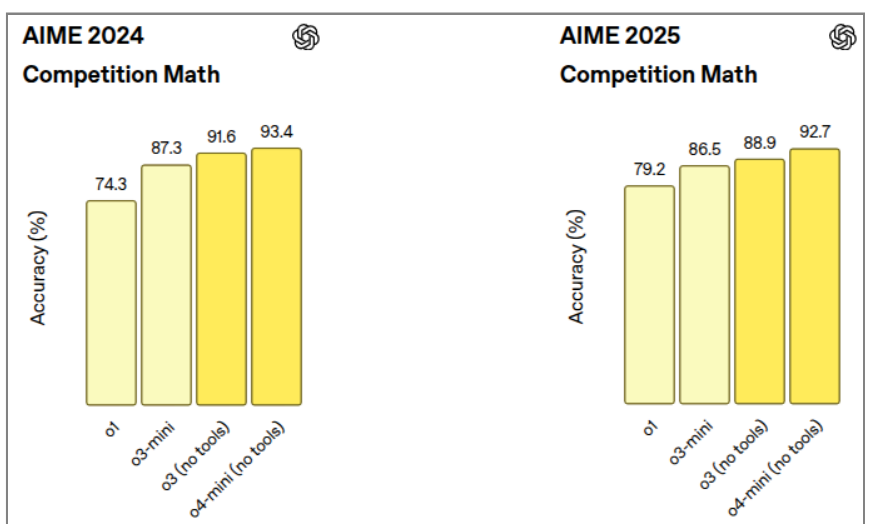
Updated (April 16, 2025): OpenAI has officially released o3. OpenAI has confirmed that this version is not the same as the one we tested in this original post. See more information on this. We will publish updated results for released o3 shortly.

OpenAI's new o3 system - trained on the ARC-AGI-1 Public Training set - has scored a breakthrough **75.7%** on the Semi-Private Evaluation set at our stated public leaderboard $10k compute limit. A high-compute (172x) o3 configuration scored **87.5%**.

**OpenAI** @OpenAI

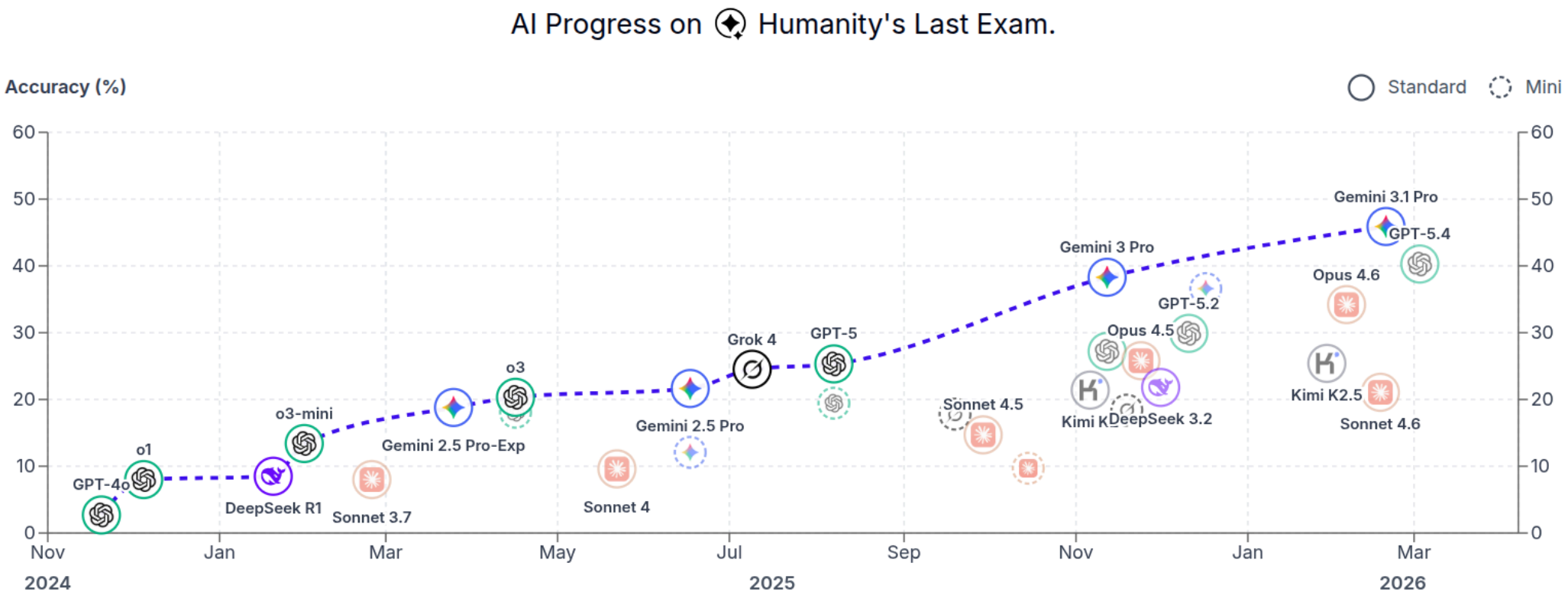
We achieved gold medal-level performance 🥇 on the 2025 International Mathematical Olympiad with a general-purpose reasoning LLM!

Our model solved world-class math problems—at the level of top human contestants. A major milestone for AI and mathematics.



# LRMs and current frontiers

source: https://agi.safe.ai



# LRMs and current frontiers

Reasoning models are the main driver of the remarkable results of LLMs in mathematical competitions and other difficult benchmarks:

- **AIME 2024** (math olympiad difficulty-level problems): OpenAI o3 scored 91.6%.
- **International Mathematical Olympiad 2025**: both Google's Gemini and OpenAI models have started solving IMO problems at the gold-medal level.
- **ARC-AGI:** benchmark for abstract reasoning, even its successor (ARC-AGI-2) is now approaching saturation (83.3% for GPT-5.4).
- **Humanity's Last Exam (HLE)**: benchmark of questions that LLMs were not able to solve in 2024 → as of 03/26, the best models have above 40%

# Midterm feedback

Please fill in! 👇

https://forms.gle/5vK6HhgPBhKHQAtB7

# Summary

# Summary

- **Scaling laws**: LLM performance follows predictable empirical laws based on model size, data size, and compute.
- **Chain-of-thought**: prompting with intermediate reasoning steps dramatically improves multi-step reasoning.
- **Test-time scaling**: using more compute at inference can be more efficient than scaling pretraining.
- **Large reasoning models** (o1/o3, DeepSeek-R1, ...): train models to produce reasoning traces using RL and/or SFT.
- Reasoning abilities are **rapidly improving**: math olympiad results, ARC-AGI saturation.

# Links and resources

- Kaplan et al. (2020): Scaling Laws for Neural Language Models
- Hoffmann et al. (2022): Training Compute-Optimal Large Language Models (Chinchilla)
- Sardana et al. (2024): Beyond Chinchilla-Optimal
- Wei et al. (2022): Chain-of-Thought Prompting
- Kojima et al. (2022): Zero-shot CoT
- Wang et al. (2022): Self-Consistency
- Snell et al. (2024): Scaling LLM Test-Time Compute
- DeepSeek-AI (2025): DeepSeek-R1
- Jay Alammar: The Illustrated DeepSeek-R1
- Sebastian Raschka: Understanding Reasoning LLMs
- Maarten Grootendorst: A Visual Guide to Reasoning LLMs