

# An Introduction to Reinforcement Learning from Human Feedback and Post-training

SALA 2026

Nathan Lambert

Quito, Ecuador

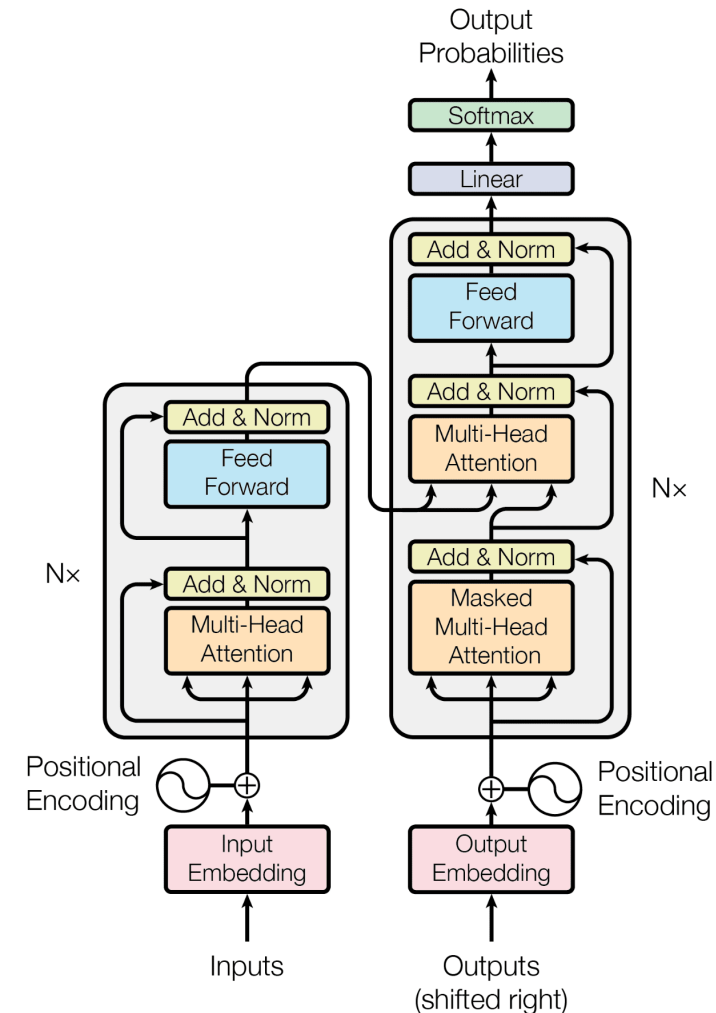
11 March 2026

A cursory overview of RLHF, RLVR, and modern post-training recipes for language models.

# What is a language model?

Core properties:

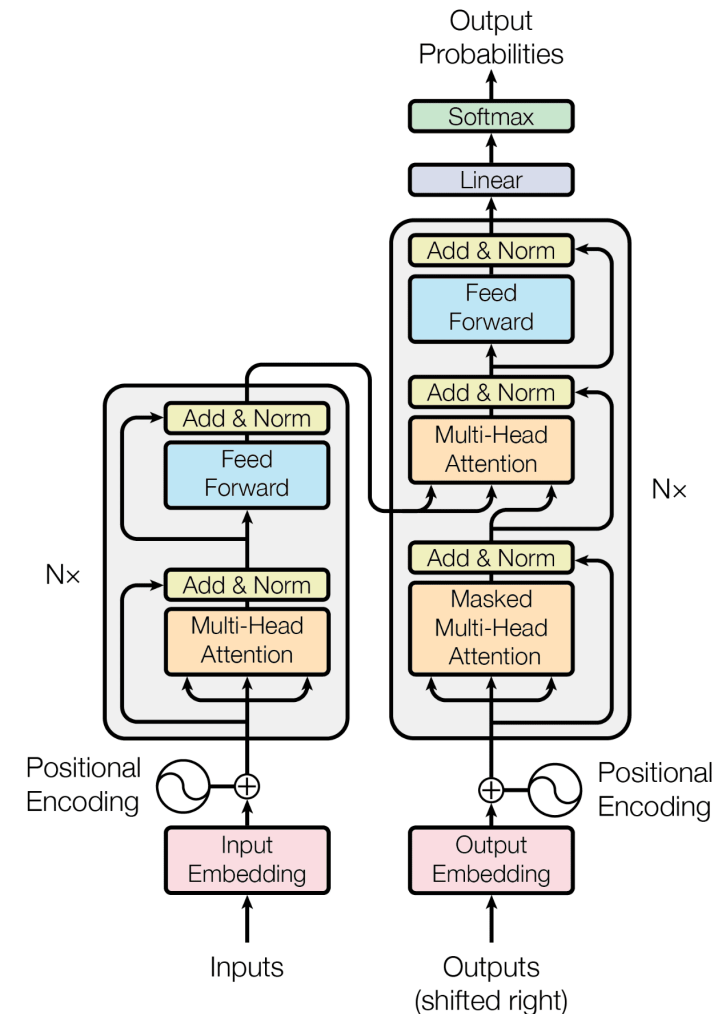
- A language model assigns probabilities to text.
- Chunks of words are broken down as **tokens**, which are the internal representation of the model.
- Given previous tokens, it predicts the next token. Repeating this produces a completion one step at a time (this is called **autoregressive**).



# What is a (modern) language model?

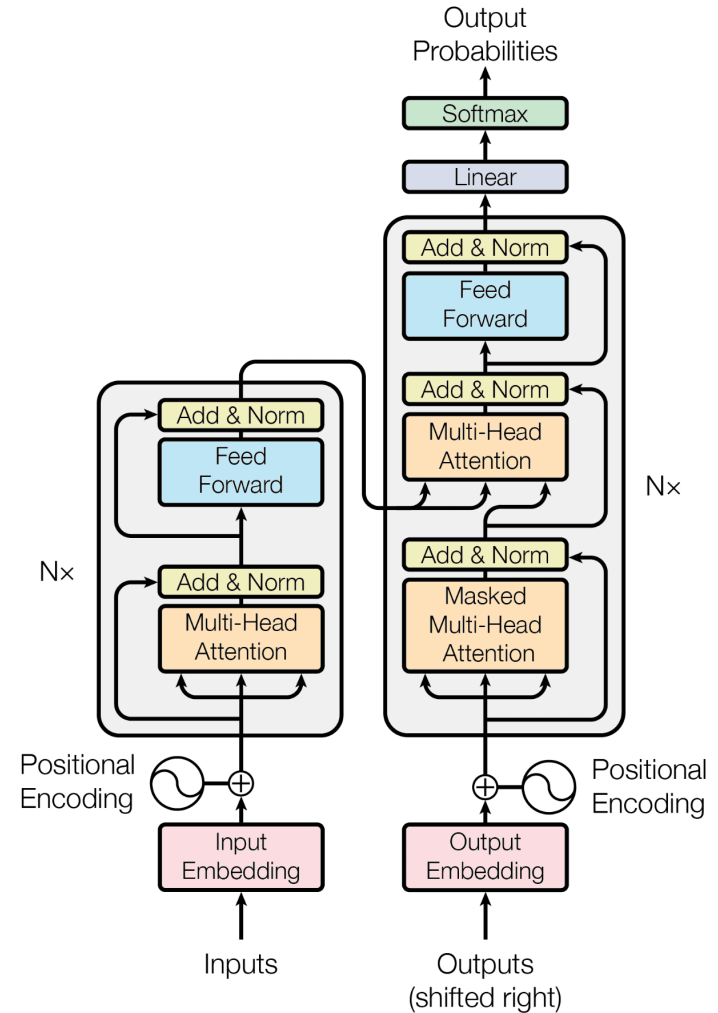
Modern language models:

- Have billions to trillions of parameters.
- Largely downstream of The Transformer architecture, which popularized the use of the **self-attention** mechanism along with fully-dense layers.
- Predict and work over much more than text: Gemini and ChatGPT work with images, audio, and video.



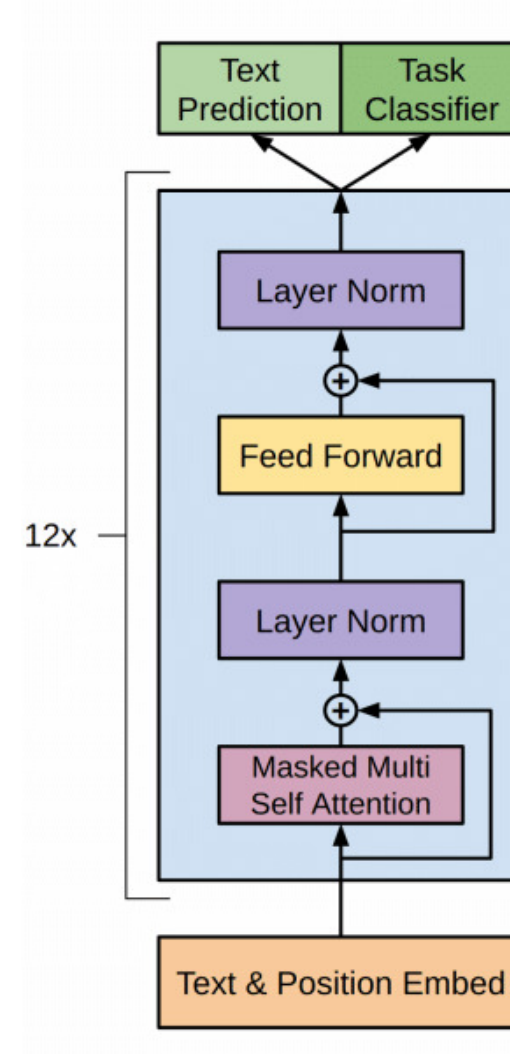
# 2017: the Transformer is born

- 2017: the Transformer is born



# 2018: GPT-1, ELMo, and BERT

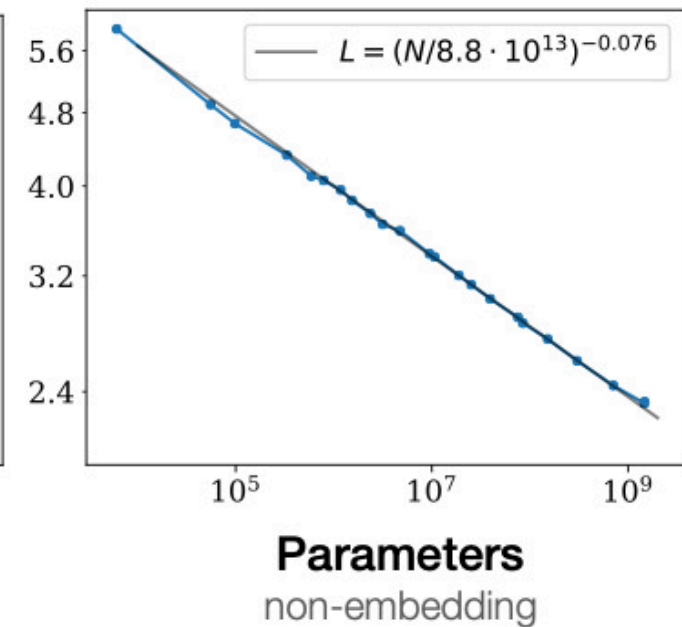
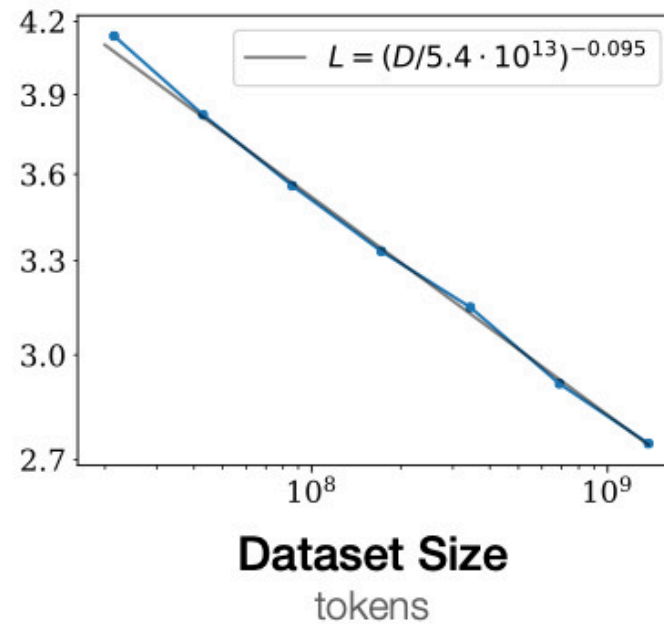
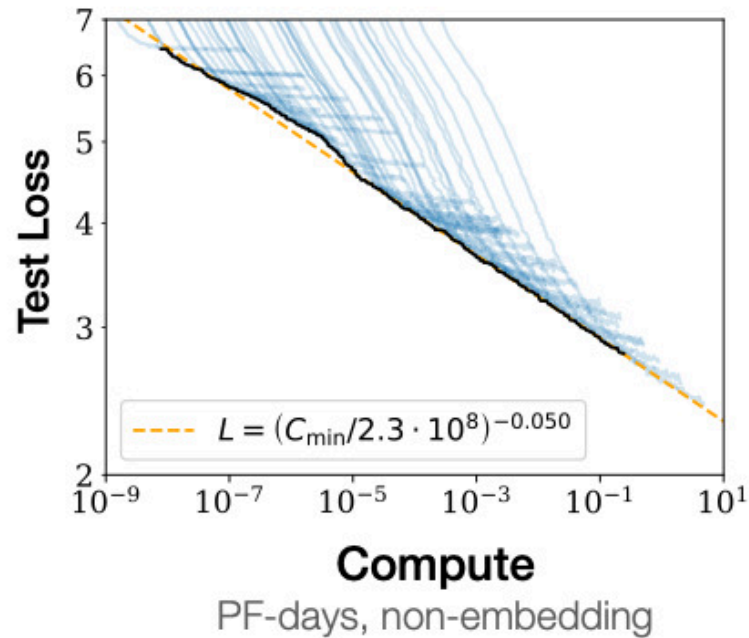
- 2017: the Transformer is born
- **2018:** GPT-1, ELMo, and BERT released



Devlin et al., 2018; Peters et al., 2018; Radford et al., 2018

# 2019: GPT-2 and scaling laws

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- **2019: GPT-2 and scaling laws**



# 2020: GPT-3 surprising capabilities

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- **2020: GPT-3 surprising capabilities**

The three settings we explore for in-context learning

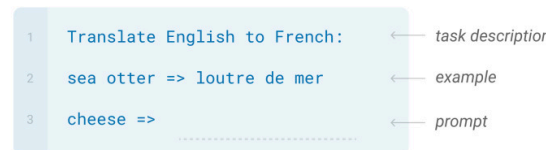
## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



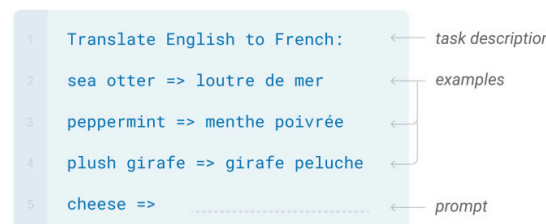
## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



## Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Traditional fine-tuning (not used for GPT-3)

## Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



# 2021: Stochastic Parrots

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- 2020: GPT-3 surprising capabilities
- **2021: Stochastic Parrots**

## On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?

Emily M. Bender\*  
ebender@uw.edu  
University of Washington  
Seattle, WA, USA

Angelina McMillan-Major  
aymm@uw.edu  
University of Washington  
Seattle, WA, USA

Timnit Gebru\*  
timnit@blackinai.org  
Black in AI  
Palo Alto, CA, USA

Shmargaret Shmitchell  
shmargaret.shmitchell@gmail.com  
The Aether

### ABSTRACT

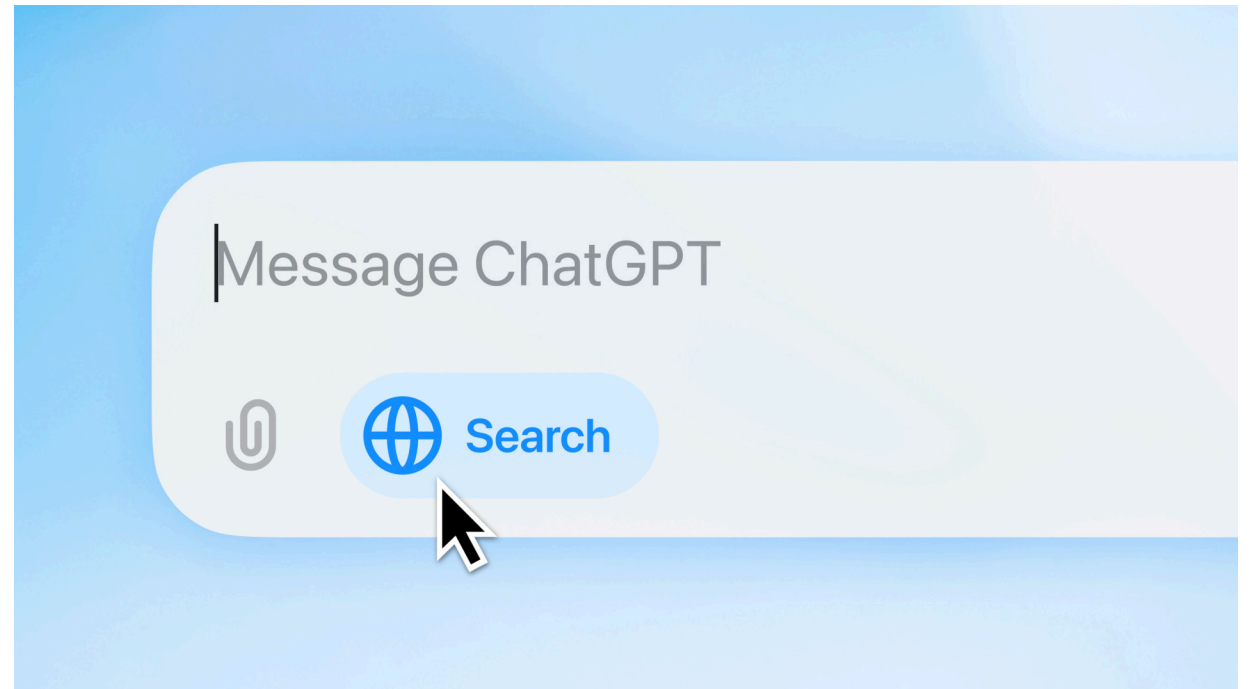
The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English. BERT, its variants, GPT-2/3, and others, most recently Switch-C, have pushed the boundaries of the possible both through architectural innovations and through sheer size. Using these pretrained models and the methodology of fine-tuning them for specific tasks, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English. In this paper, we take a step back and ask: How big is too big? What are the possible risks associated with this technology and what paths are available for mitigating those risks? We provide recommendations including weighing the environmental and financial costs first, investing resources into curating and carefully documenting datasets rather than ingesting everything on

alone, we have seen the emergence of BERT and its variants [39, 70, 74, 113, 146], GPT-2 [106], T-NLG [112], GPT-3 [25], and most recently Switch-C [43], with institutions seemingly competing to produce ever larger LMs. While investigating properties of LMs and how they change with size holds scientific interest, and large LMs have shown improvements on various tasks (§2), we ask whether enough thought has been put into the potential risks associated with developing them and strategies to mitigate these risks.

We first consider environmental risks. Echoing a line of recent work outlining the environmental and financial costs of deep learning systems [129], we encourage the research community to prioritize these impacts. One way this can be done is by reporting costs and evaluating works based on the amount of resources they consume [57]. As we outline in §3, increasing the environmental and financial costs of these models doubly punishes marginalized com-

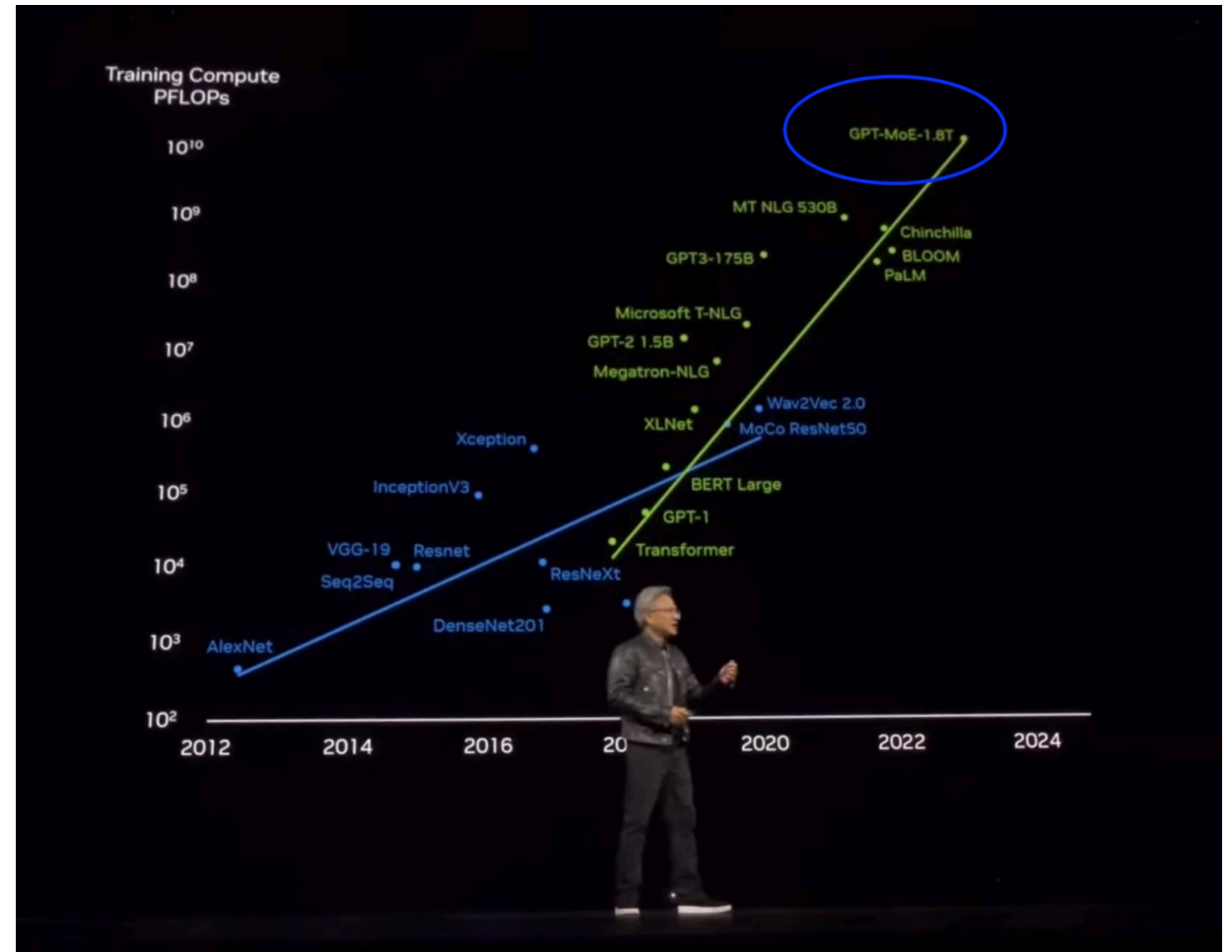
# 2022: ChatGPT

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- 2020: GPT-3 surprising capabilities
- 2021: Stochastic Parrots
- **2022: ChatGPT**



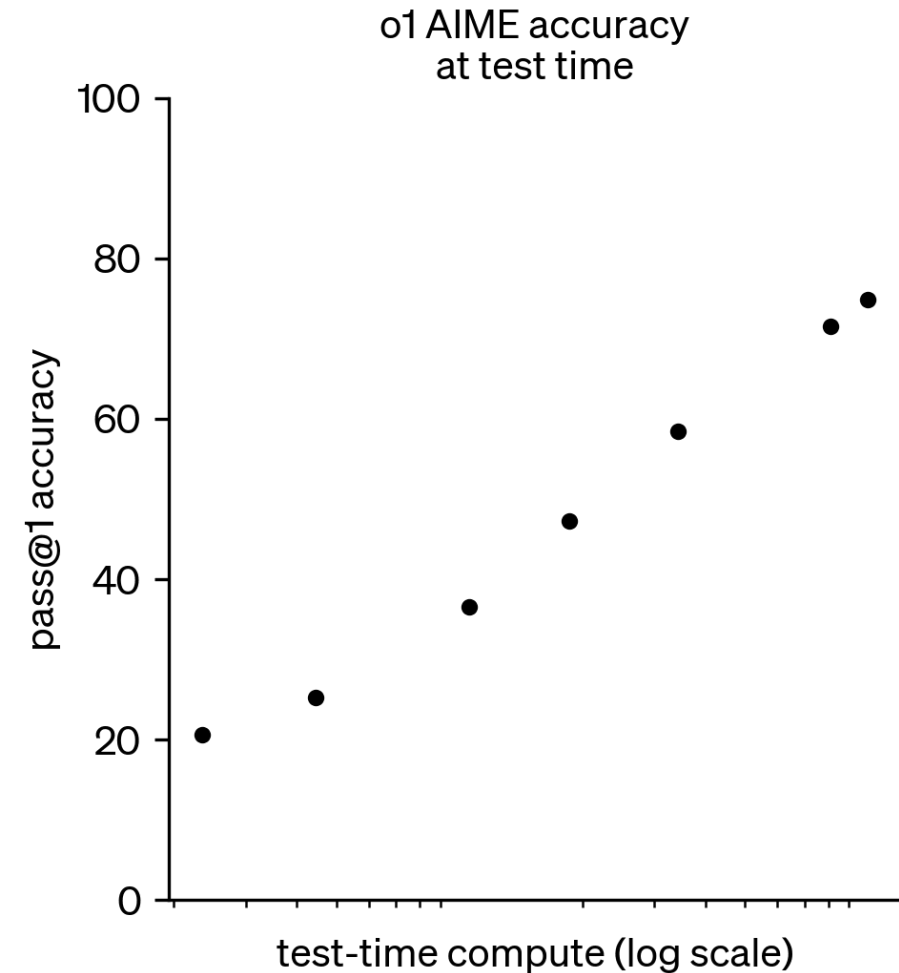
# 2023: GPT-4 and frontier-scale

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- 2020: GPT-3 surprising capabilities
- 2021: Stochastic Parrots
- 2022: ChatGPT
- **2023: GPT-4 and frontier-scale**



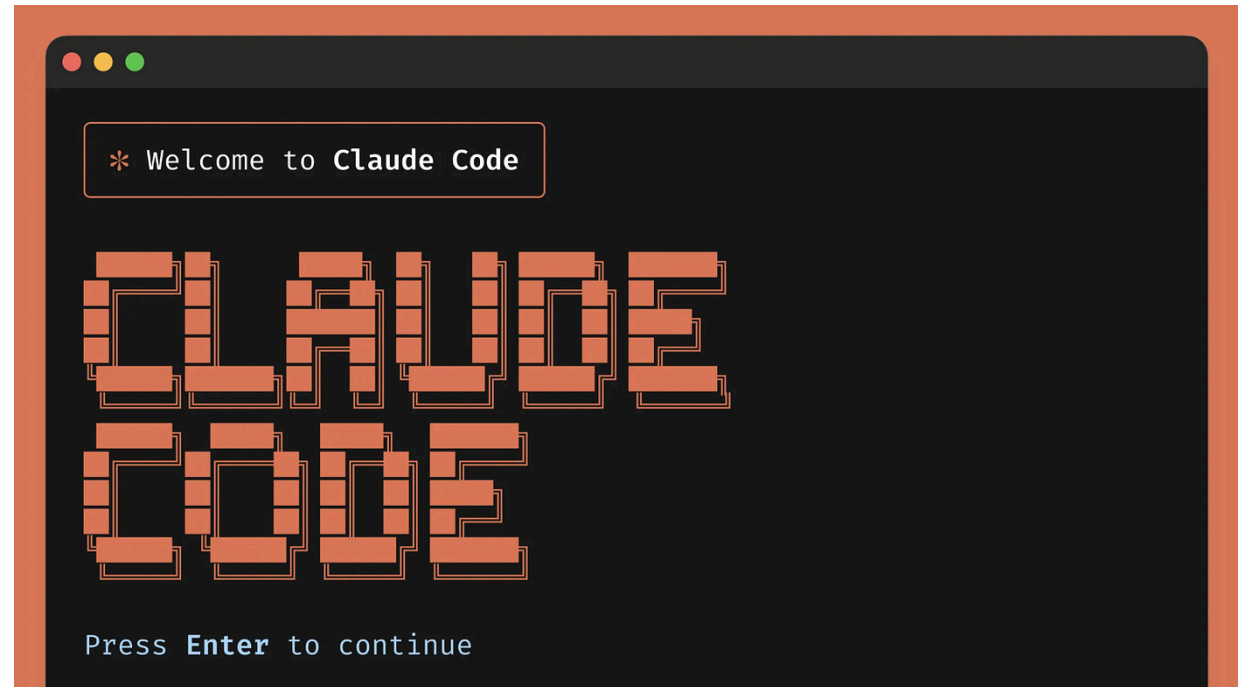
# 2024: o1 and reasoning models

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- 2020: GPT-3 surprising capabilities
- 2021: Stochastic Parrots
- 2022: ChatGPT
- 2023: GPT-4 and frontier-scale
- **2024: o1 and reasoning models**



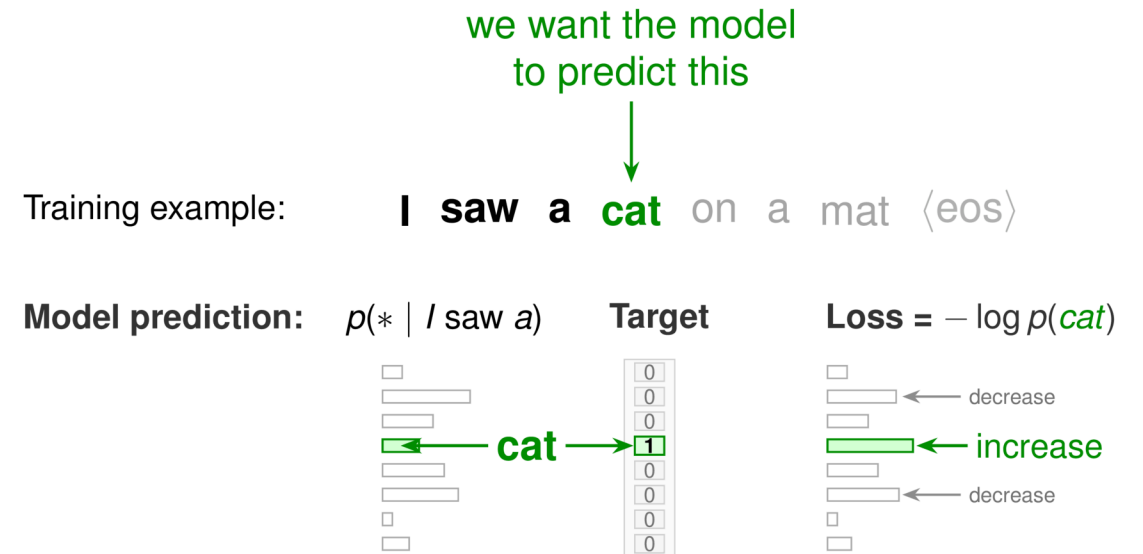
# 2025: o3, Claude Code, and agents

- 2017: the Transformer is born
- 2018: GPT-1, ELMo, and BERT released
- 2019: GPT-2 and scaling laws
- 2020: GPT-3 surprising capabilities
- 2021: Stochastic Parrots
- 2022: ChatGPT
- 2023: GPT-4 and frontier-scale
- 2024: o1 and reasoning models
- **2025: o3, Claude Code, and agents**



# Pretraining: next-token prediction

- Train on a trillions of tokens of text from the web, books, code, and documents
  - Models are often trained on 5-50+ trillion tokens
  - 1T of text tokens is about 3-5 TB of data
  - Labs gather and filter 10-20X more data than is used for the model
  - Total data funnel targetted for models is on the order of petabytes
- Objective: predict the next token in each sequence
- Result: Incredible, flexible, useful models



# A base model completes text

After pretraining we are left with a glorified autocomplete model, for example:<sup>1</sup>

USER

The president of the United States in 2006 was

ASSISTANT

George W. Bush, the governor of Florida in 2006 was Jeb Bush, and John McCain was an Arizona senator in 2006...

<sup>1</sup>: Base models are also becoming more flexible through midtraining and better data mixtures.

# Post-training makes it answer like a chatbot

USER

The president of the United States in 2006 was

ASSISTANT

George W. Bush was the president of the United States in 2006. He served two terms in office, from January 20, 2001, to January 20, 2009.

# ChatGPT made language models usable

- Better **format**: direct answers instead of raw continuations
- Better **style**: helpful, concise, conversational
- Better **product**: people could actually use it every day

USER

I'm giving a talk on RLHF tomorrow. Can you help me structure it?

ASSISTANT

Absolutely — here's a simple structure:

## 1. Start with the basics

- What a language model is
- Why next-token prediction is not enough

## 2. Explain post-training

- Instruction tuning
- Preference tuning / RLHF
- RLVR for verifiable tasks

## 3. End with why it matters

- Better products like ChatGPT
- Better reasoning in models like o1

# The Elicitation Theory

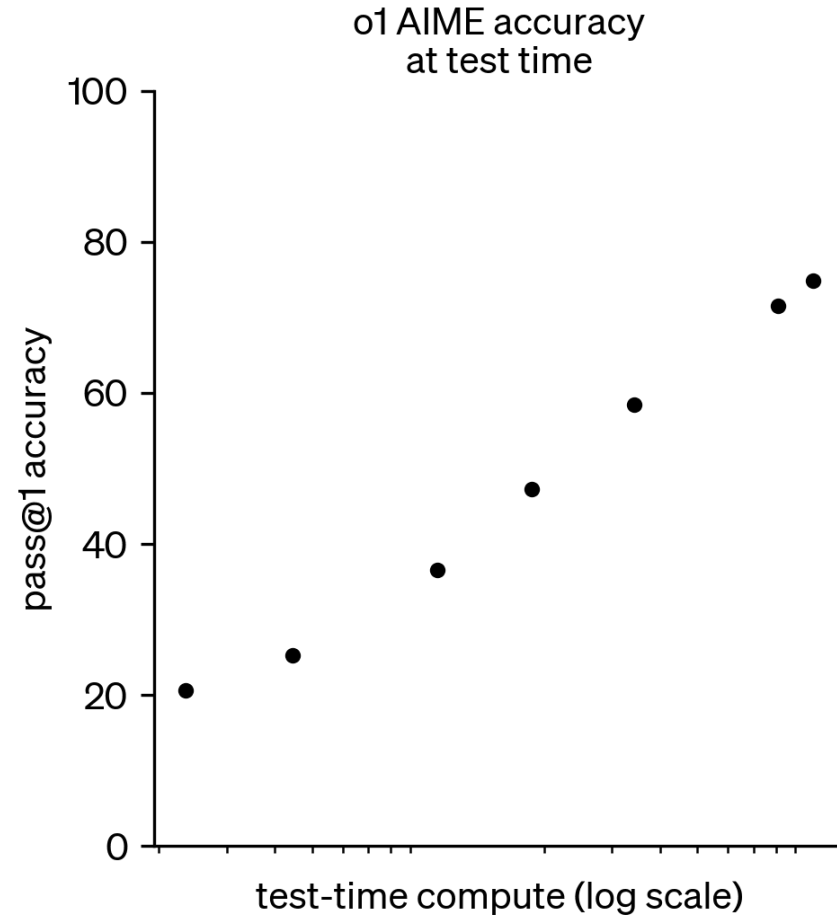
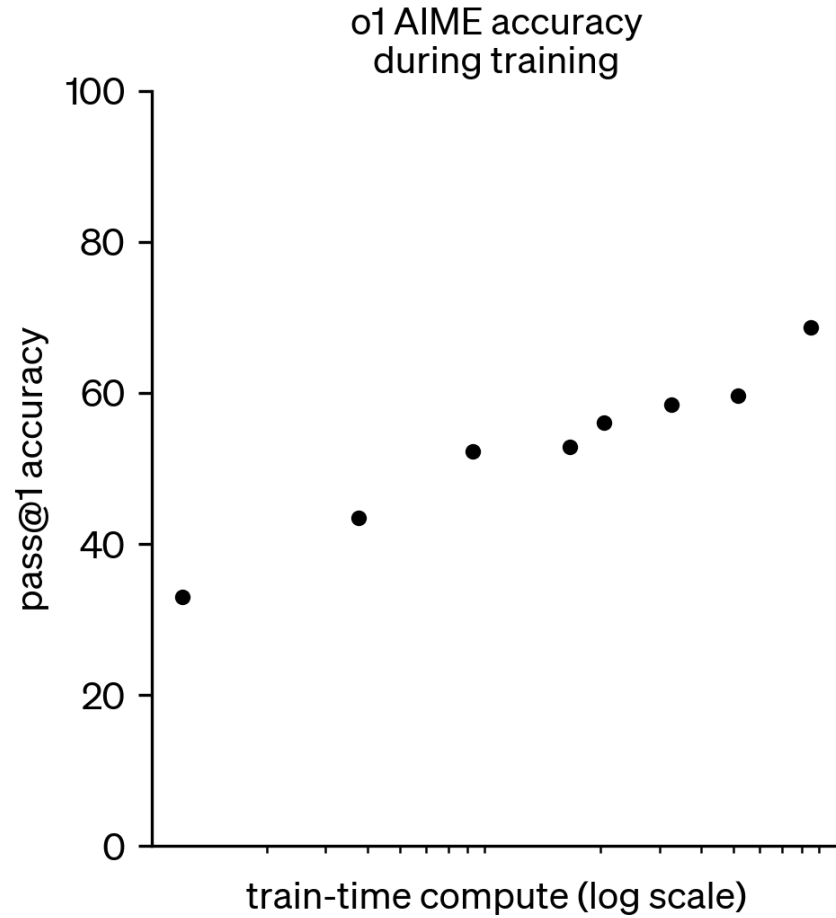
Post-training **extracts latent potential** from the base model

**OLMoE** — same base model, updated only post-training:

- Version 1: **35** benchmark average
- Version 2: **48** benchmark average

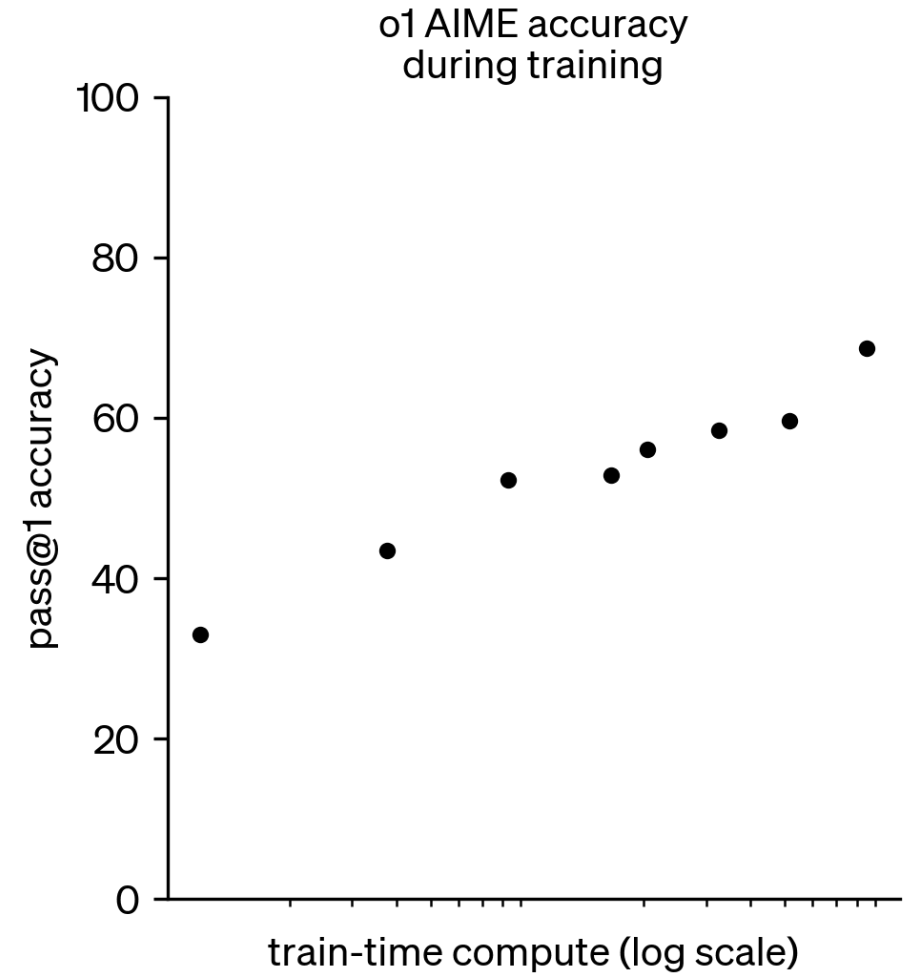
Base models determine the *ceiling*. Post-training's job is to **reach it**.

# o1 scaling post training



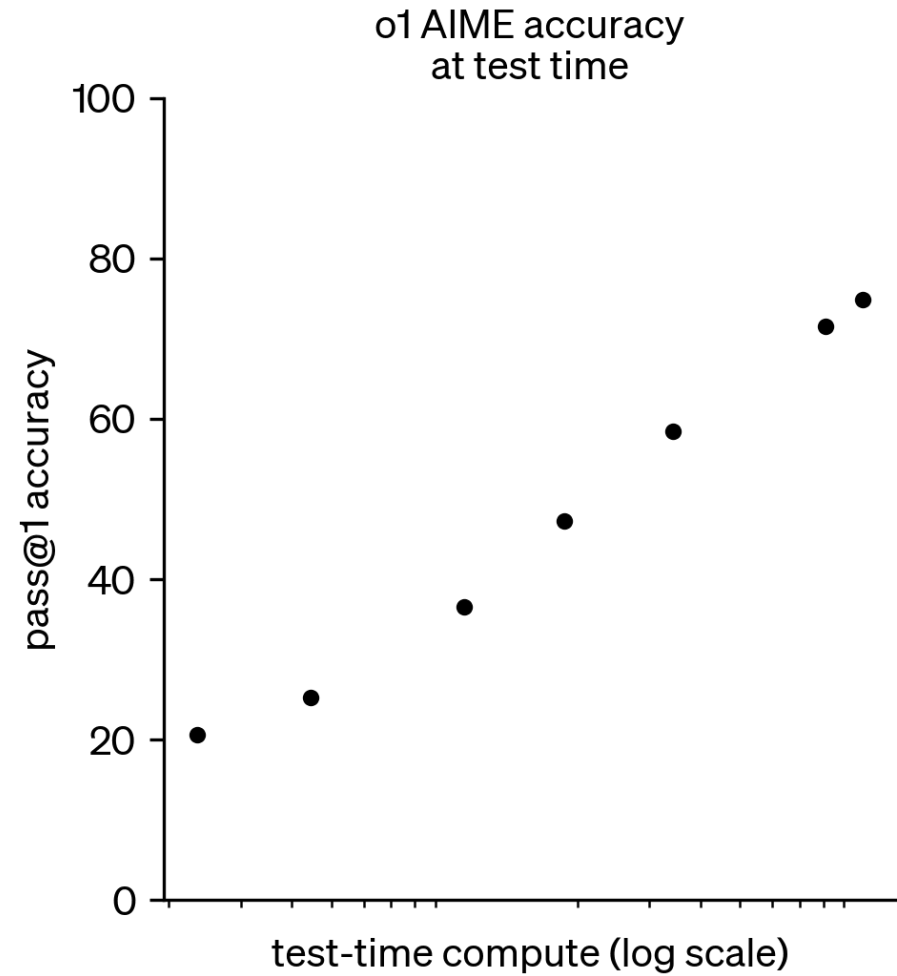
# o1: train-time scaling

text goes here



# o1: test-time scaling asd asds

text goes here



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

🌕  
Explain the moon landing to a 6 year old

A labeler demonstrates the desired output behavior.

👤  
Some people went to the moon...

This data is used to fine-tune GPT-3 with supervised learning.

SFT  
🧠  
📄📄📄

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

🌕  
Explain the moon landing to a 6 year old

A Explain gravity... B Explain war...  
C Moon is natural satellite of... D People went to the moon...

A labeler ranks the outputs from best to worst.

👤  
D > C > A = B

This data is used to train our reward model.

RM  
🧠  
D > C > A = B

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

🐸  
Write a story about frogs

The policy generates an output.

PPO  
🧠  
Once upon a time...

The reward model calculates a reward for the output.

RM  
🧠

The reward is used to update the policy using PPO.

$r_k$

# ChatGPT's success

- Turned language models into a **mass-market product**
- Made post-training feel as important as pretraining
- Changed what users expect from AI systems

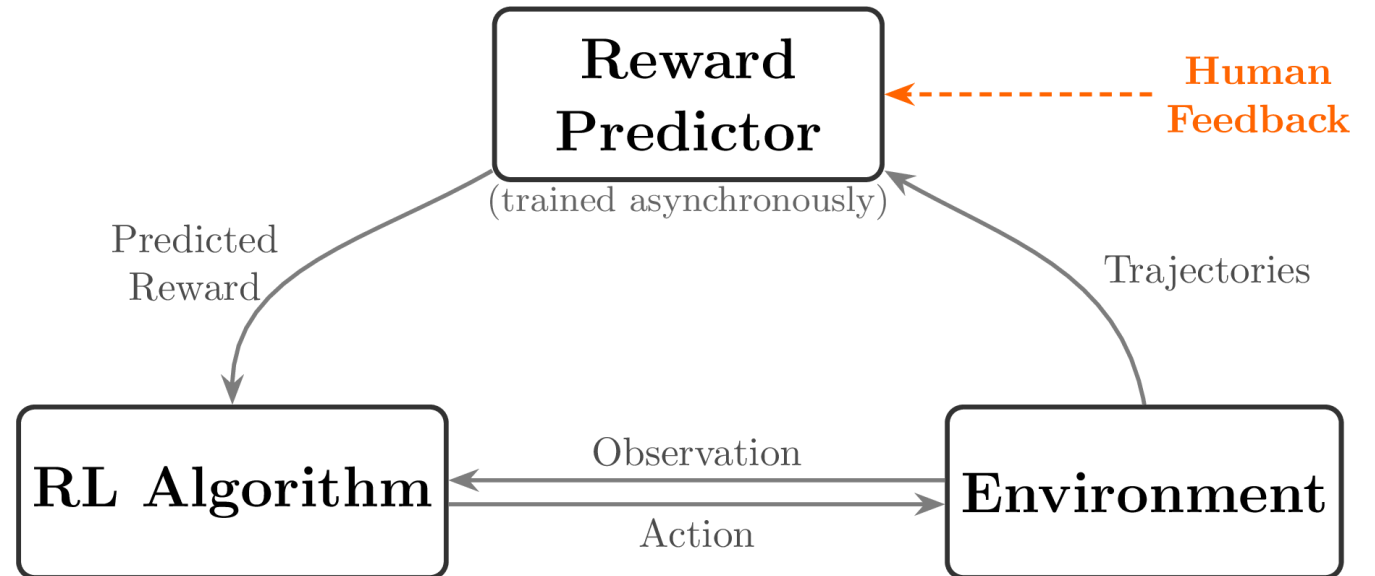
# The Scale of Post-Training

- **DeepSeek R1**: RL used ~147K H800 GPU hours (~5% of total training)
- Individual ablation runs: **10–100K GPU hours**
- The trend: **more compute going to post-training every year**

# How We Got Here

# RLHF Before Language Models

- **TAMER** (Knox & Stone, 2008) — humans score agent actions to learn a reward
- **Christiano et al. 2017** — RLHF on Atari trajectory preferences
- **Ziegler et al. 2019** — first RLHF on language models



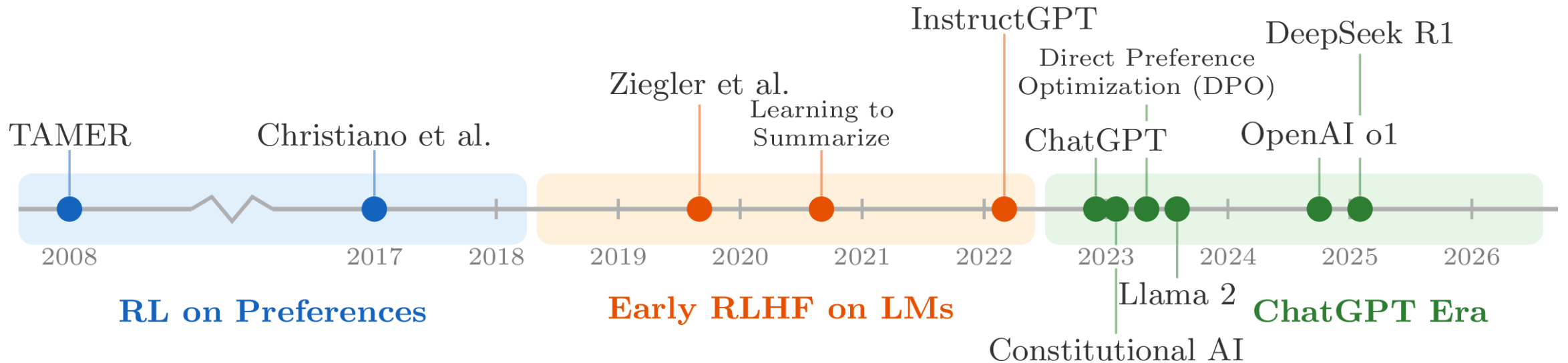
# The Post-ChatGPT Acceleration

- **Early 2023:** Alpaca era — limited data, impressive but narrow
- **Late 2023:** DPO — direct alignment without a reward model
- **2024:** Complex multi-stage recipes (Llama 3, Tulu 3)
- **2025:** Reasoning via RL (DeepSeek R1, o1)

# The Classic RLHF Pipeline

1. **Instruction fine-tuning** — teach Q&A format from examples
2. **Reward model training** — learn a scoring function from human preferences
3. **RL optimization** — optimize the model against the reward model

# Landmark Papers



- **Ziegler 2019** (Ziegler et al., 2019) — first RLHF on language models
- **InstructGPT** (Ouyang et al., 2022) — the recipe behind ChatGPT
- **Constitutional AI** (Bai et al., 2022) — AI feedback and the Claude line
- **DPO** (Rafailov et al., 2023) — direct alignment without a reward model
- **Llama 3** (Grattafiori et al., 2024) and **Tulu 3** (Lambert et al., 2024) — modern multi-stage recipes
- **DeepSeek R1** (Guo et al., 2025) — reasoning-first RL

# Training Recipes Have Evolved

	InstructGPT (2022)	Tulu 3 (2024)	DeepSeek R1 (2025)
<b>IFT data</b>	~10K	~1M	100K+
<b>Preference data</b>	~100K	~1M	On-policy
<b>RL stage</b>	~100K prompts	~10K (RLVR)	"Until convergence"
<b>Stages</b>	3	3	4

More data, more stages, more RL compute.

# **RLHF vs. RLVR vs. Classical RL**

# Classical RL

- Agent takes actions in an **environment** with state transitions
- Reward is a **known function** of the environment
- Multi-step, fine-grained rewards at each timestep
- Goal: maximize cumulative return over a trajectory

# RLHF

- No environment — prompts sampled from a dataset
- Reward is **learned** from human preferences (a proxy)
- **Response-level** reward (bandit-style, not per-token)
- Regularized with **KL penalty** to stay close to the base model

$$J(\pi) = \mathbb{E} [r_{\theta}(x, y)] - \beta D_{\text{KL}}(\pi \parallel \pi_{\text{ref}})$$

# RLVR

- Same RL setup as RLHF, but reward is **verifiable**
- Math: check the final answer. Code: run the tests.
- No learned reward model — **no proxy objective**
- Enables scaling RL compute on reasoning tasks

# Comparison

	Classical RL	RLHF	RLVR
<b>Reward</b>	Environment	Learned (proxy)	Verifiable (exact)
<b>State transitions</b>	Yes	No	No
<b>Reward granularity</b>	Per-step	Per-response	Per-response
<b>Failure mode</b>	Exploration	Over-optimization	Task coverage
<b>Example</b>	CartPole	Chat style tuning	Math reasoning

# Where Things Are Heading

- RLHF and RLVR are **complementary** — style vs. capabilities
- Modern recipes use **both** in sequence
- The boundary between them is blurring (generative reward models, self-correction)

# Beyond elicitation?

- Maybe post-training does more than just **extract** existing ability
- Long RL runs may reshape how models **reason**, not just how they respond
- Open question: when does scaling RL create **new capabilities**?

# Thank You

[rlhfbook.com](http://rlhfbook.com)

# References (1/3)

Anthropic. “*Claude Code*.” 2025. [link]

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., et al.. “*Constitutional ai: Harmlessness from ai feedback*.” *arXiv preprint arXiv:2212.08073*, 2022.

Bender, E., Gebru, T., McMillan-Major, A., and Shmitchell, S.. “*On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*.” *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021. [link]

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., et al.. “*Language Models are Few-Shot Learners*.” *Advances in Neural Information Processing Systems*, 2020. [link]

Christiano, P., Leike, J., Brown, T., Martic, M., Legg, S., et al.. “*Deep Reinforcement Learning from Human Preferences*.” *Advances in Neural Information Processing Systems*, 2017.

Devlin, J., Chang, M., Lee, K., and Toutanova, K.. “*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.” *arXiv preprint arXiv:1810.04805*, 2018. [link]

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., et al.. “*The llama 3 herd of models*.” *arXiv preprint arXiv:2407.21783*, 2024.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., et al.. “*Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*.” *arXiv preprint arXiv:2501.12948*, 2025.

# References (2/3)

Kaplan, J., McCandlish, S., Henighan, T., Brown, T., Chess, B., et al.. “*Scaling Laws for Neural Language Models.*” *arXiv preprint arXiv:2001.08361*, 2020. [\[link\]](#)

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., et al.. “*Tulu 3: Pushing Frontiers in Open Language Model Post-Training.*” *arXiv preprint arXiv:2411.15124*, 2024.

OpenAI. “*ChatGPT: Optimizing Language Models for Dialogue.*” *OpenAI Blog*, 2022. [\[link\]](#)

OpenAI. “*GPT-4 Technical Report.*” *arXiv preprint arXiv:2303.08774*, 2023. [\[link\]](#)

OpenAI. “*Introducing OpenAI o1-preview.*” *OpenAI Blog*, 2024. [\[link\]](#)

OpenAI. “*Introducing OpenAI o3 and o4-mini.*” *OpenAI Blog*, 2025. [\[link\]](#)

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., et al.. “*Training language models to follow instructions with human feedback.*” *Advances in Neural Information Processing Systems*, 2022.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., et al.. “*Deep Contextualized Word Representations.*” *Proceedings of NAACL-HLT*, 2018. [\[link\]](#)

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I.. “*Improving Language Understanding by Generative Pre-Training.*” 2018. [\[link\]](#)

# References (3/3)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., et al.. “*Language Models are Unsupervised Multitask Learners.*” 2019. [\[link\]](#)

Rafailov, R., Sharma, A., Mitchell, E., Manning, C., Ermon, S., et al.. “*Direct preference optimization: Your language model is secretly a reward model.*” *Advances in Neural Information Processing Systems*, 2023.

Team, O.. “*New Tools for Building Agents.*” *OpenAI Blog*, 2025. [\[link\]](#)

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al.. “*Attention Is All You Need.*” *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017. [\[link\]](#)

Ziegler, D., Stiennon, N., Wu, J., Brown, T., Radford, A., et al.. “*Fine-tuning language models from human preferences.*” *arXiv preprint arXiv:1909.08593*, 2019.