香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

COMP 4901B

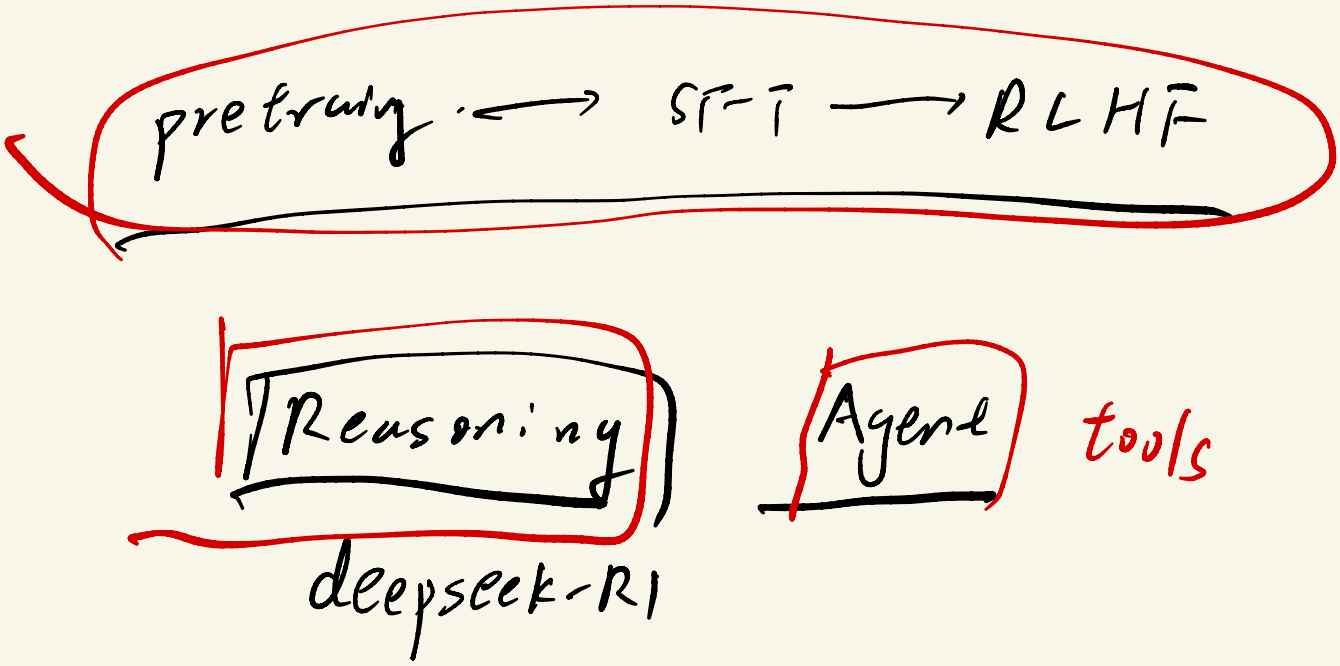
Large Language Models

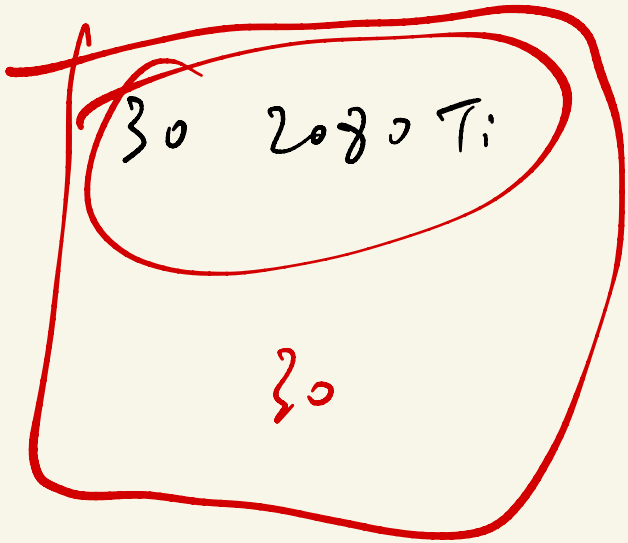
RLHF and LLMs Scaling Laws

Junxian He

Oct 22, 2025

Part of the slides are adapted from Stanford CS336


pretrain
SFT
RLHF
Reasoning
deepseek-R1
Agent
tools



CSP

API

Recap: Why Do We Want Reinforcement Learning

SFT

- Imitation external data
- Performance limited by external data

RL

- Maximize reward rather than imitation
- The model may surpass humans (e.g., AlphaGo)

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

This objective looks kinda like weighted log likelihood maximization?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

This objective looks kinda like weighted log likelihood maximization?

What is different?

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

This objective looks kinda like weighted log likelihood maximization?

What is different?

1. Have a weight of R(x)

Review: REINFORCE / Policy Gradient

$$\hat{g} = \mathbb{E}_{x \sim p_\theta(x)} R(x) \nabla_\theta \log p_\theta(x)$$

How to implement?

$$\text{Objective} = \sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)}) \qquad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$$

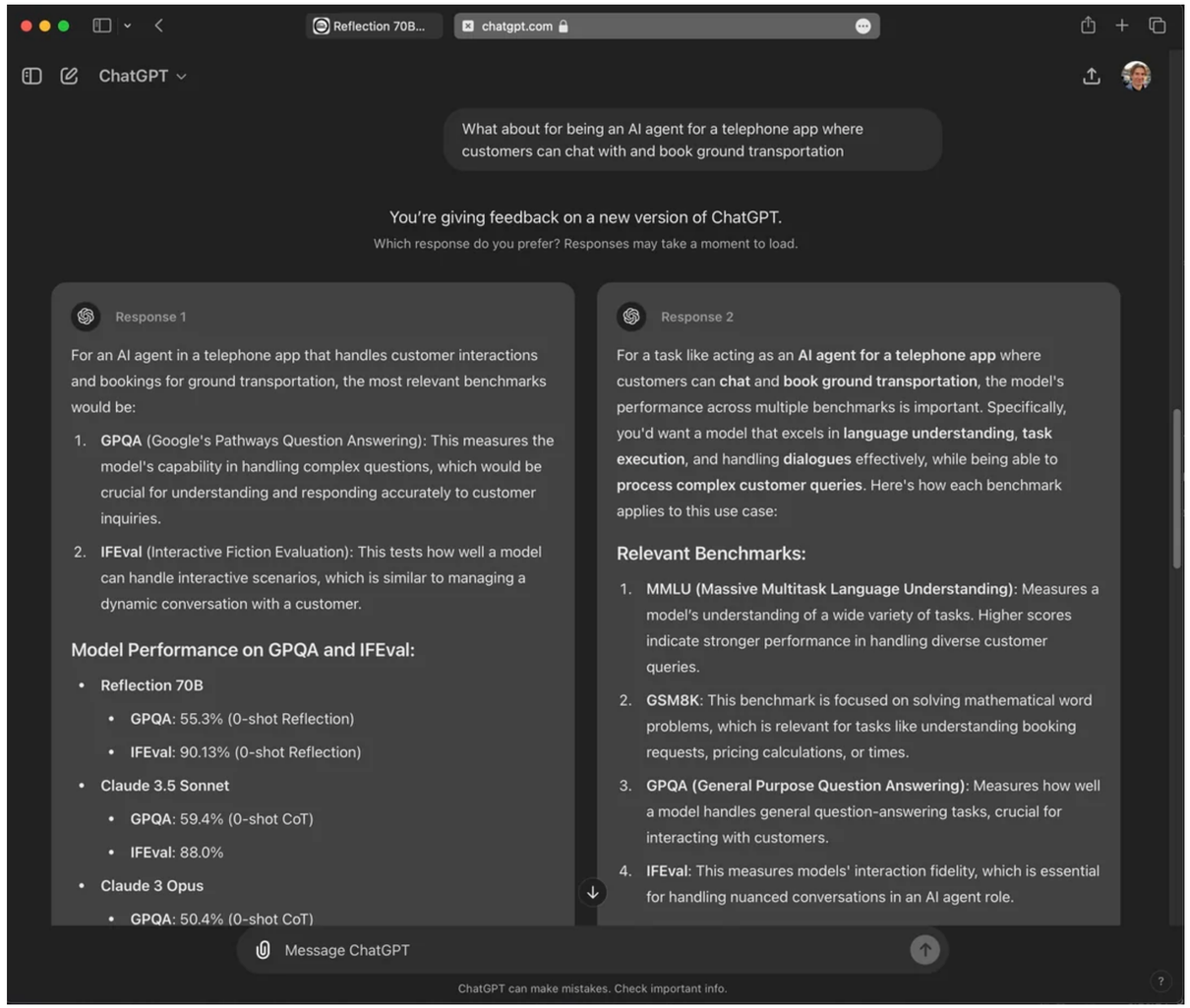
This objective looks kinda like weighted log likelihood maximization?

What is different?

1. Have a weight of R(x)
2. The data x is sampled from the model itself, not from a static dataset

Collecting high-quality Human Feedbacks matters a lot, but not easy

Human Feedback is Precious



Many products already track our feedback to help improve the models

RLHF: Human Annotation Matters

In this task, you will be provided with a **Prompt** from a user (e.g., a question, instruction, statement) to an AI chatbot along with two potential machine-generated **Responses** to the Prom

Your job is to assess which of the two Responses is better for the Prompt, considering the following for each Response:

Helpfulness: To what extent does the Response provide useful information or satisfying content for the Prompt? Responses should: ▪ **Address the intent of the user's Prompt** such that a user would not feel the Prompt was ignored or misinterpreted by the Response. ▪ **Provide specific, comprehensive, and up-to-date information** for the user needs expressed in the Prompt. ▪ **Be sensible and coherent.** The response should not contain any nonsensical information or contradict itself across sentences (e.g., refer to two different people with the same name as if they are the same person). ▪ **Adhere to any requirements indicated in the Prompt** such as an explicitly specified word length, tone, format, or information that the Response should include. ▪ **Not contain inaccurate, deceptive, or misleading information** (based on your current knowledge or quick web search - you do not need to perform a rigorous fact check) ▪ **Not contain harmful, offensive, or overly sexual content** A Response may sometimes intentionally avoid or decline to address the question/request of the Prompt and may provide a reason for why it is unable to respond. For example, "Sorry, there may not be a helpful answer to this question." These responses can be considered helpful in cases where an appropriate helpful response to the Prompt does not seem possible.	Rating scale: ▪ **Not at All Helpful:** Response is useless/irrelevant, contains even a single piece of nonsensical/inaccurate/deceptive/misleading information, and/or contains harmful/offensive/overly sexual content. ▪ **Slightly Helpful:** Response is somewhat related to the Prompt, does not address important aspects of the Prompt, and/or contains outdated information. ▪ **Somewhat Helpful:** Response partially addresses the intent of the Prompt (most users would want more information), contains extra unhelpful information, and/or is lacking helpful details/specifics. ▪ **Very Helpful:** Response addresses the intent of the Prompt with a satisfying response. Some users might want a more comprehensive response with additional details or context. It is comparable to a response an average human with basic subject-matter knowledge might provide. ▪ **Extremely Helpful:** Response completely addresses the intent of the Prompt and provides helpful details/context. It is comparable to a response a talented/well-informed human with subject-matter expertise might provide.
Presentation: To what extent is the content of the Response conveyed well? Responses should: ▪ **Be organized in a structure that is easy to consume and understand.** Flowing in a logical order and makes good use of formatting such paragraphs, lists, or tables. ▪ **Be clearly written in a polite neutral tone** that is engaging, direct, and inclusive. The tone should not be *overly* friendly, salesy, academic, sassy, or judgmental in a way that most users would consider to be off-putting or overdone. ▪ **Have consistent style with natural phrasing and transitions** as if composed by a single talented human. ▪ **Not be rambling, repetitive. or contain clearly off-topic information.** Similar information should not be repeated multiple times. It is harder for users to consume the helpful information in a response if there is repetitive or less helpful information mixed into the response. ▪ **Not include notable language issues or grammatical errors**	Rating scale: ▪ **Poor:** Response is poorly written or has notable structural, formatting. language, or grammar issues. Or Response has an awkward or inappropriate tone. Or the Response repeats similar information. Or only a small portion of the Response contains helpful information. ▪ **Adequate:** Response could have been written/organized better or may have minor language/grammar issues. A minimal amount of less helpful information may be present. Users would still feel the content of the Response was easy to consume. ▪ **Excellent:** Response is very well written and organized. Sentences flow in a logical order with smooth transitions and consistent style. The content of the Response is conveyed in a way that is comparable to a response a talented human might produce.

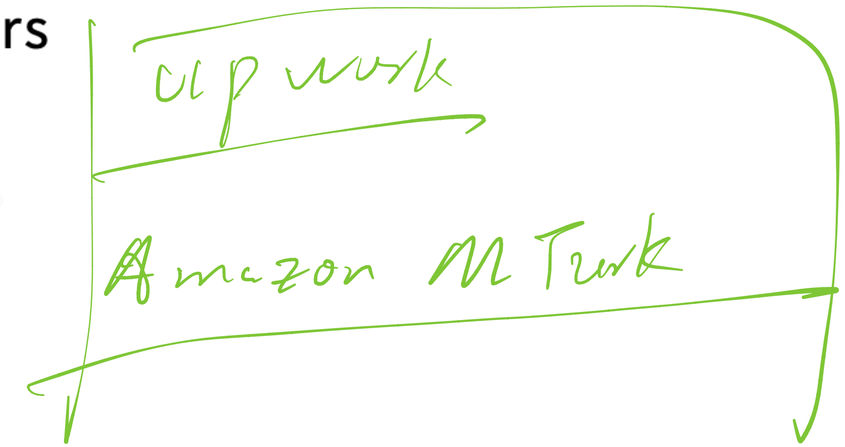
Overall, you should consider both factors in your SxS rating of which response is better. A more concise response presenting the most helpful information directly and clearly is usually better than a longer response that may be harder to consume and/or contains clearly off-topic information. Responses with Poor Presentation (e.g., rambling, inappropriate tone) should play a significant role in your assessment of which side is better. It may help to imagine the user chatting with a real person and consider which Response most users would prefer to receive from a real person.

How you annotate the reward model training data will subsequently decide the model behaviors

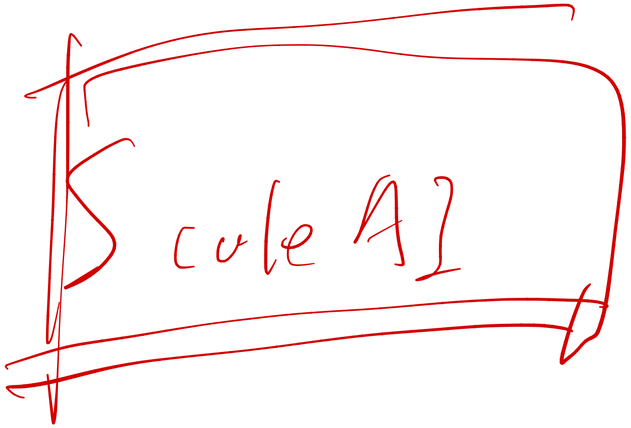
Crowdsourcing is difficult

from strangers at scale

- Hard to get really high-quality, verifiable annotators
- Hard to get them to really check correctness
- Have to be careful about GPT4 use..



Crowdsourcing is difficult



- Hard to get really high-quality, verifiable annotators
- Hard to get them to really check correctness
- Have to be careful about GPT4 use..

data

Nowadays, many annotators just use LLMs to annotate to make money…

Crowdsourcing Ethics

Data collection at scale can have significant issues

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

15 MINUTE READ

This image was generated by OpenAI's image-generation software, Dall-E 2. The prompt was: "A seemingly endless view of African workers at desks in front of computer screens in a printmaking style." TIME does not typically use AI-generated art to illustrate its stories, but chose to in this instance in order to draw attention to the power of OpenAI's technology and shed light on the labor that makes it possible. Image generated by Dall-E 2/OpenAI

TECHNOLOGY

AMERICA ALREADY HAS AN AI UNDERCLASS

Search engines, ChatGPT, and other AI tools wouldn't function without an army of contractors. Now those workers say they're underpaid and mistreated.

By Matteo Wong

Crowdsource Biases

The annotator distribution for RLHF can significantly shift its behaviors

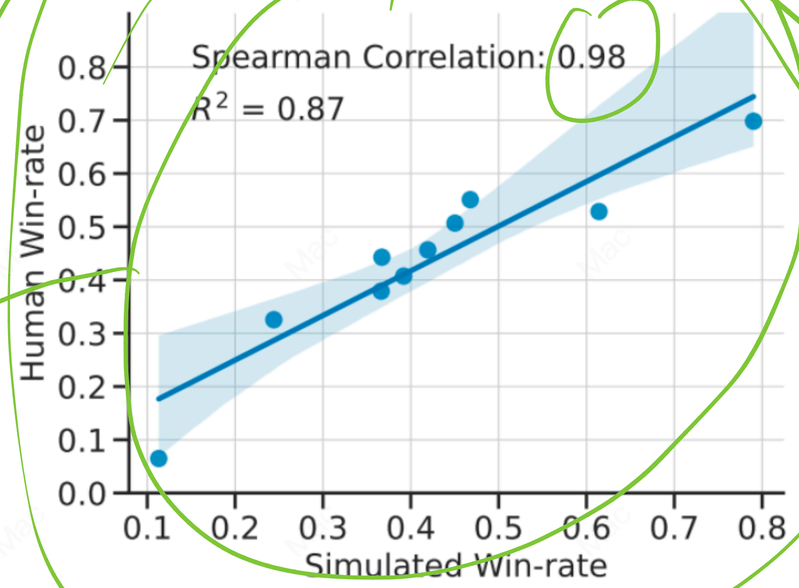
Table 12: Labeler demographic data

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%
What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

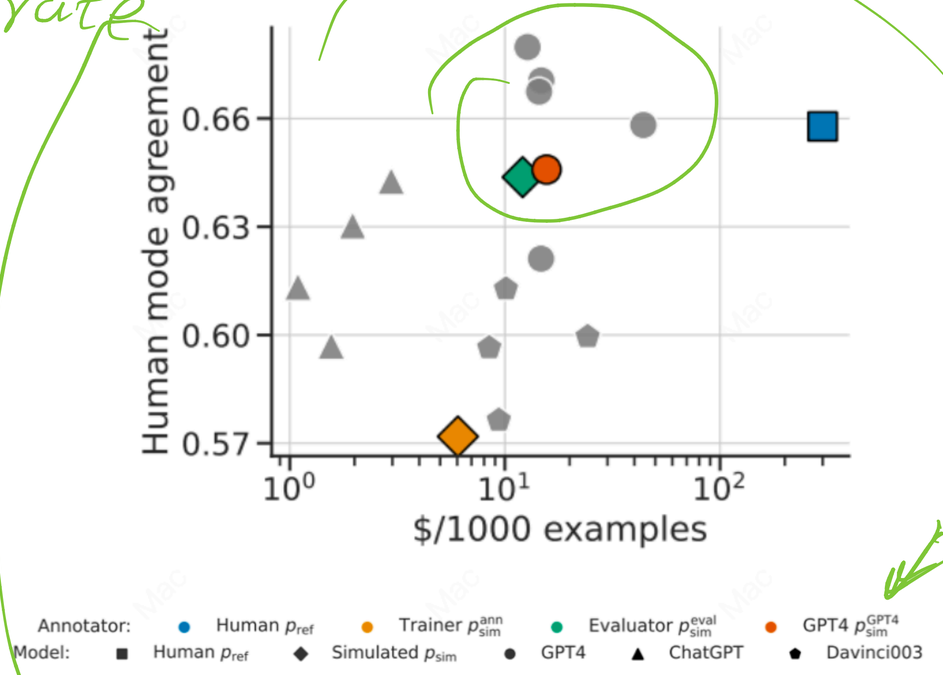
Group	AI21			OpenAI					
	J1-grande	J1-jumbo	j1-grande-v2-beta	ada	davinci	text-ada-001	text-davinci-001	text-davinci-002	text-davinci-003
RELIG									
Protestant	0.813	0.814	0.797	0.821	0.788	0.709	0.715	0.755	0.694
Roman Catholic	0.815	0.820	0.806	0.825	0.794	0.709	0.716	0.759	0.700
Mormon	0.792	0.794	0.778	0.803	0.772	0.700	0.709	0.752	0.694
Orthodox	0.771	0.776	0.762	0.783	0.754	0.688	0.698	0.733	0.693
Jewish	0.794	0.796	0.785	0.801	0.773	0.699	0.710	0.758	0.706
Muslim	0.786	0.796	0.788	0.793	0.775	0.684	0.704	0.730	0.698
Buddhist	0.771	0.784	0.776	0.783	0.764	0.682	0.703	0.747	0.709
Hindu	0.778	0.798	0.793	0.789	0.776	0.683	0.703	0.728	0.707
Atheist	0.774	0.778	0.772	0.786	0.761	0.690	0.707	0.766	0.713
Agnostic	0.783	0.789	0.781	0.795	0.768	0.698	0.715	0.771	0.715
Nothing in particular	0.815	0.819	0.802	0.826	0.791	0.712	0.715	0.765	0.698

RLAIF — When the Feedback is from AI

GPT4 is a surprisingly good pairwise feedback system



Near-perfect rank correlation at the system level



Agreement near human inter-annotator levels

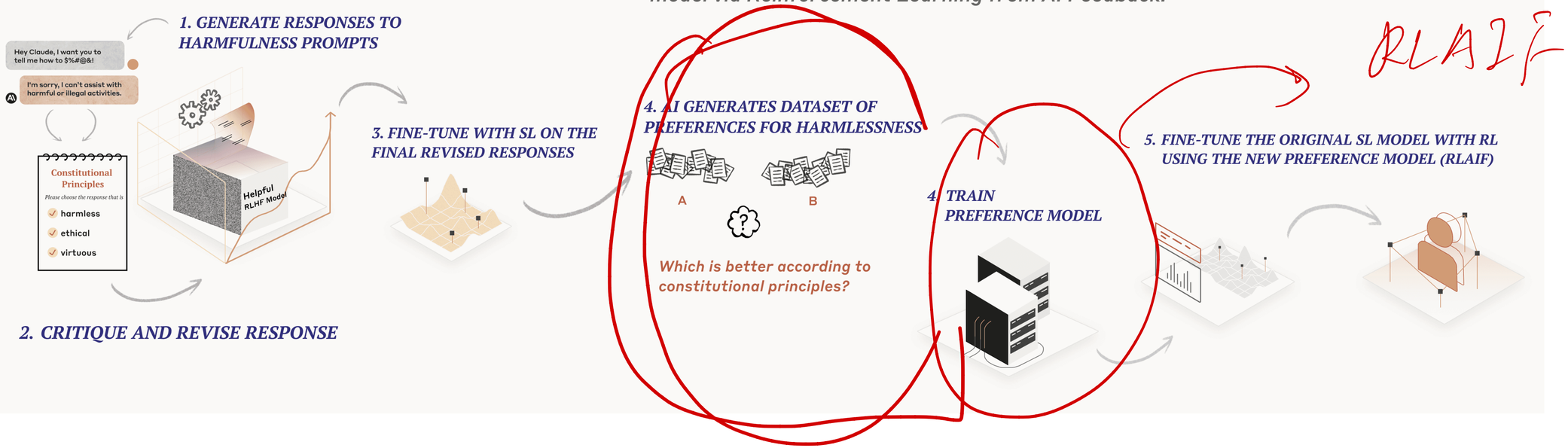
RLAIF — Constitutional AI

1. Supervised Learning (SL) Stage

Revises harmful AI responses through iterative self-critique and fine-tuning.

2. Reinforcement Learning (RL) Stage

Uses AI evaluations of responses according to constitutional principles to generate preference data for harmlessness and uses it to train a new model via Reinforcement Learning from AI Feedback.



Anthropic. Constitutional AI: Harmlessness from AI Feedback. 2022

Challenges of RL(HF)

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, ..., x^{(n)} \sim p_\theta(x)$

Training Inference

Each training step, the algorithm needs to run inference again

Inefficient

Challenges of RL(HF)

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$

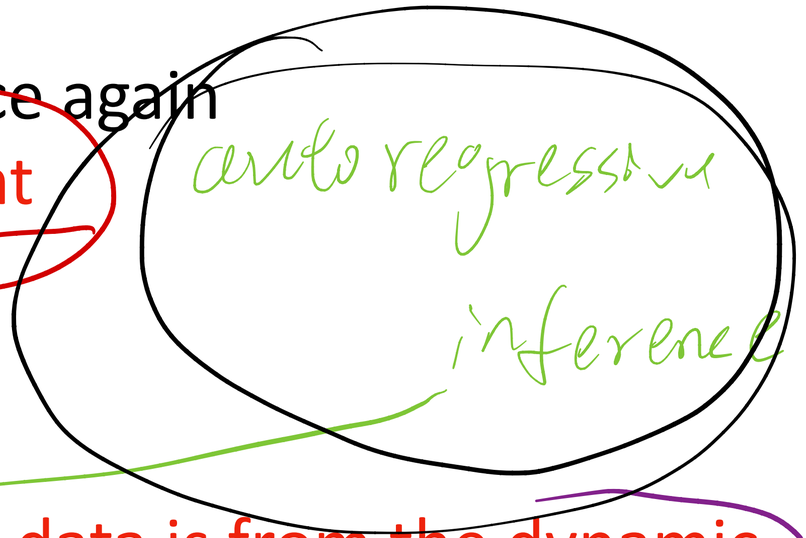
Training

$x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Inference

Each training step, the algorithm needs to run inference again

Inefficient



The original RL is called "on-policy" because the training data is from the dynamic policy itself. SFT is like "off-policy" where the training data is from a fixed distribution

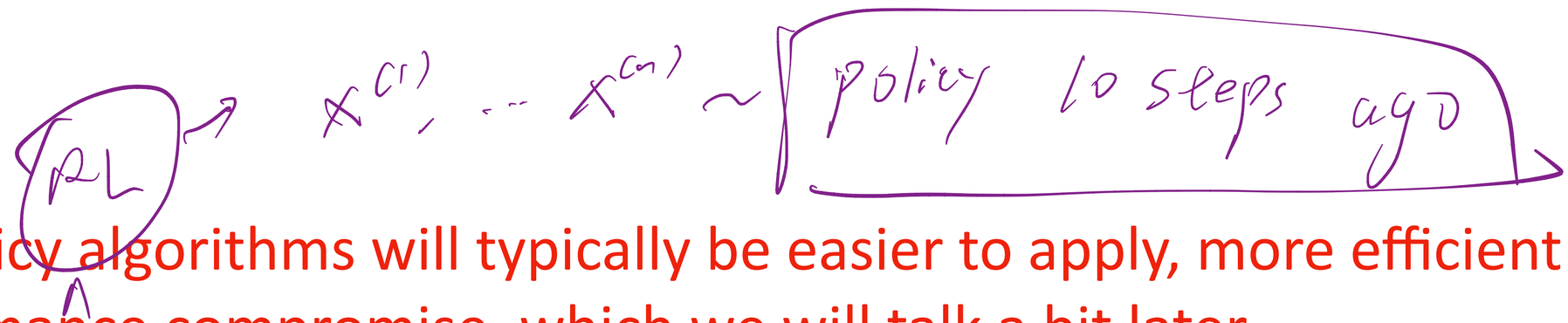
Challenges of RL(HF)

Objective = $\sum_{i=1}^{n} \frac{1}{n} R(x^{(i)}) \log p_\theta(x^{(i)})$ $\quad x^{(1)}, \ldots, x^{(n)} \sim p_\theta(x)$

Training Inference

Each training step, the algorithm needs to run inference again

Inefficient

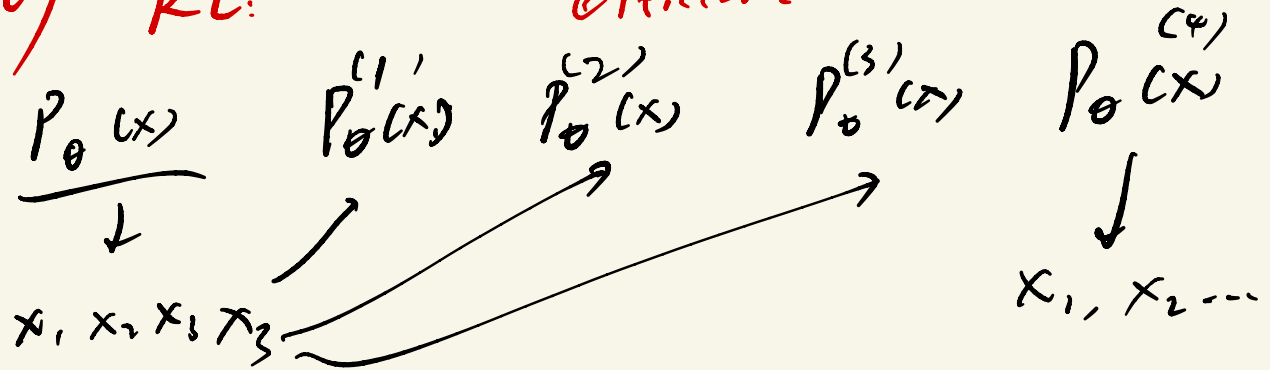


Off-policy algorithms will typically be easier to apply, more efficient and stable, with performance compromise, which we will talk a bit later

example

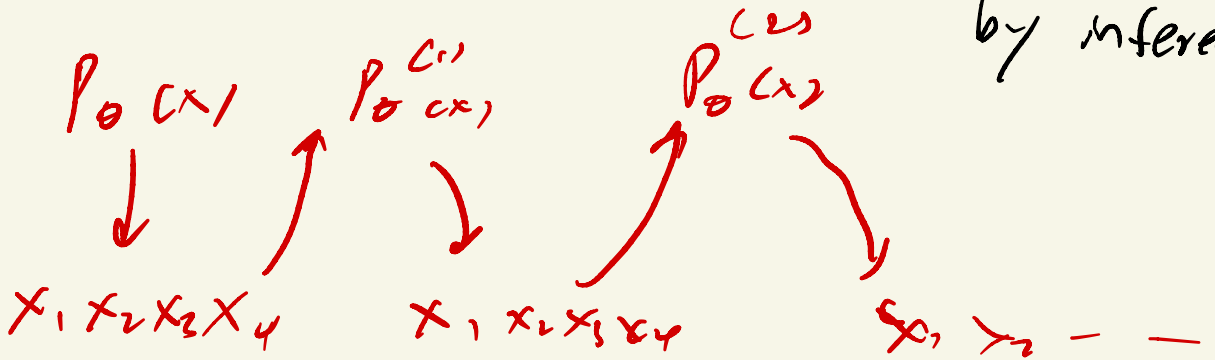
off-policy RL:

efficient



efficiency is mainly bottleneck by inference

on-policy RL:



香港科技大學
THE HONG KONG
UNIVERSITY OF SCIENCE
AND TECHNOLOGY

LLM Scaling Laws (for pretraining)

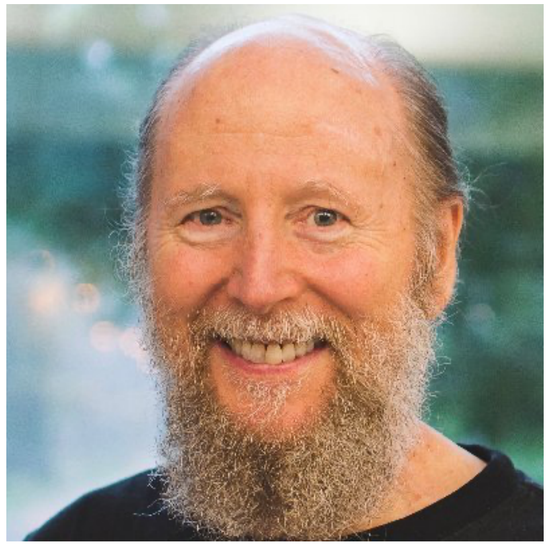
The Bitter Lesson

The Bitter Lesson

Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.



The Bitter Lesson

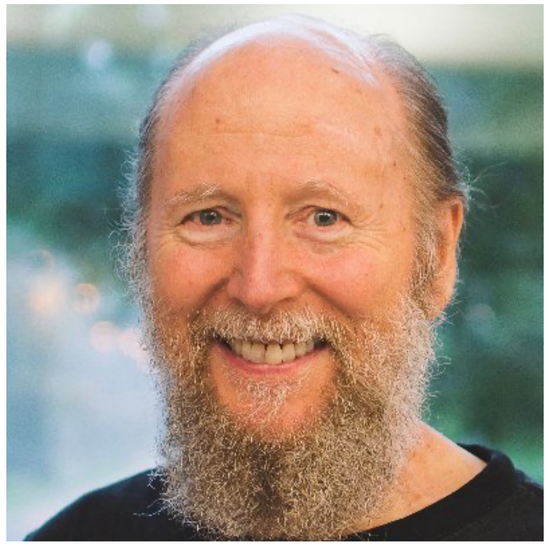
The Bitter Lesson

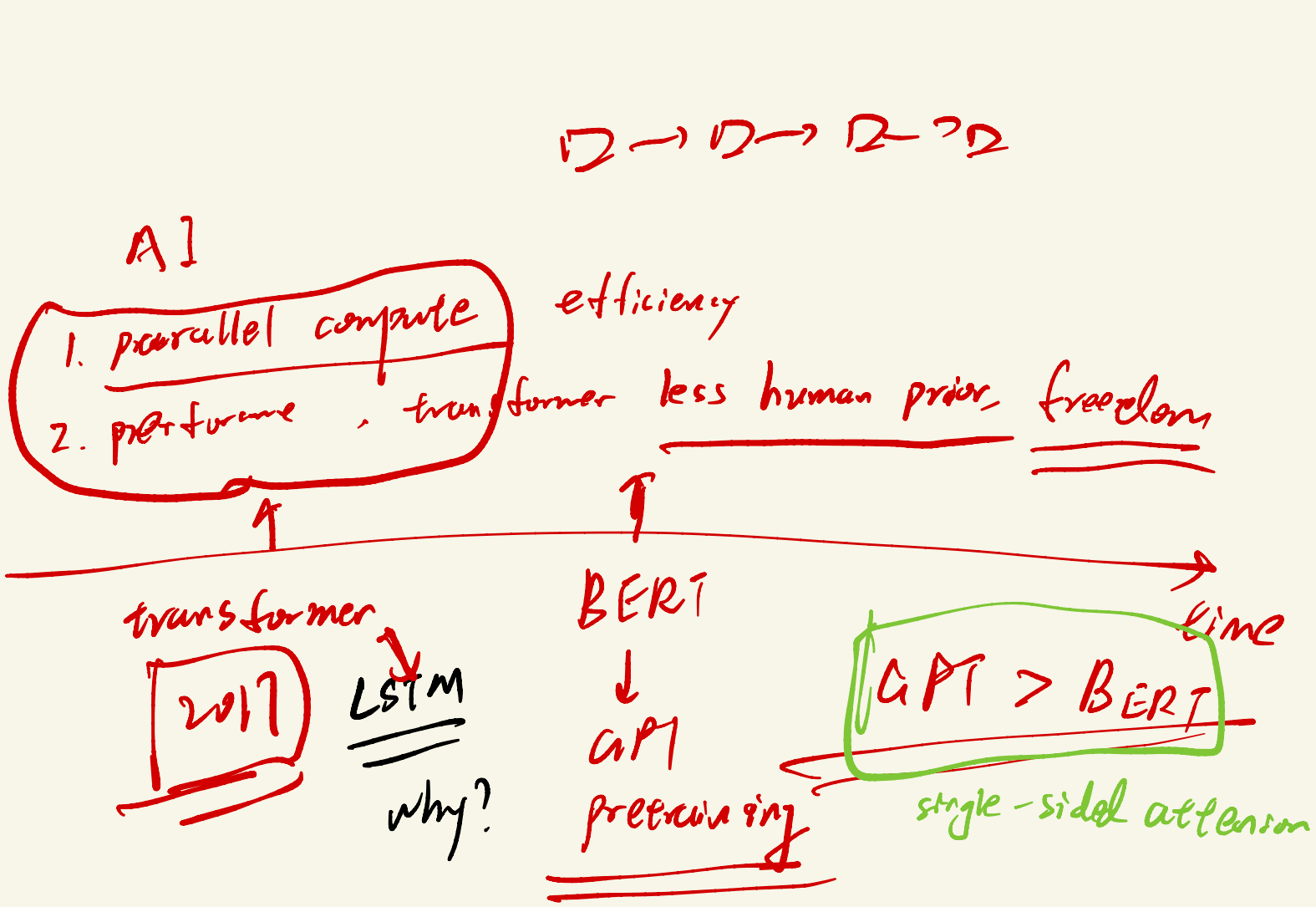
Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

Turing Award → RL



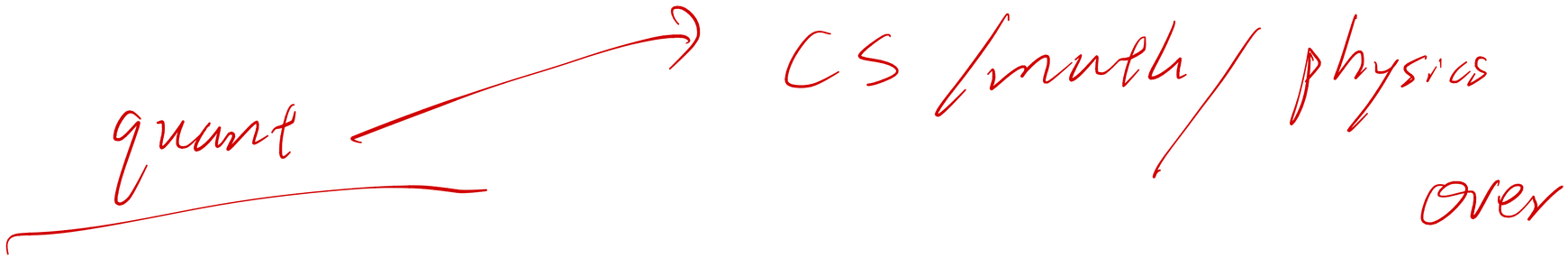

AI
1. parallel compute
2. performance
transformer
efficiency
less human prior
freedom
transformer
2017
LSTM
why?
BERT
API
pretraining
fine
GPT > BERT
single-sided attension

The Bitter Lesson

“The bitter lesson is a claim in artificial intelligence that, in the long run, simpler systems that can scale with available computational power will outperform more complex systems that integrate domain-specific human knowledge“

The Bitter Lesson

“The bitter lesson is a claim in artificial intelligence that, in the long run, simpler systems that can scale with available computational power will outperform more complex systems that integrate domain-specific human knowledge“



"Every time I fire a linguist, the performance of the speech recognizer goes up”

— Frederick Jelinek at IBM Research to develop speech recognizer

The Bitter Lesson

“The bitter lesson is a claim in artificial intelligence that, in the long run, simpler systems that can scale with available computational power will outperform more complex systems that integrate domain-specific human knowledge“

The Bitter Lesson

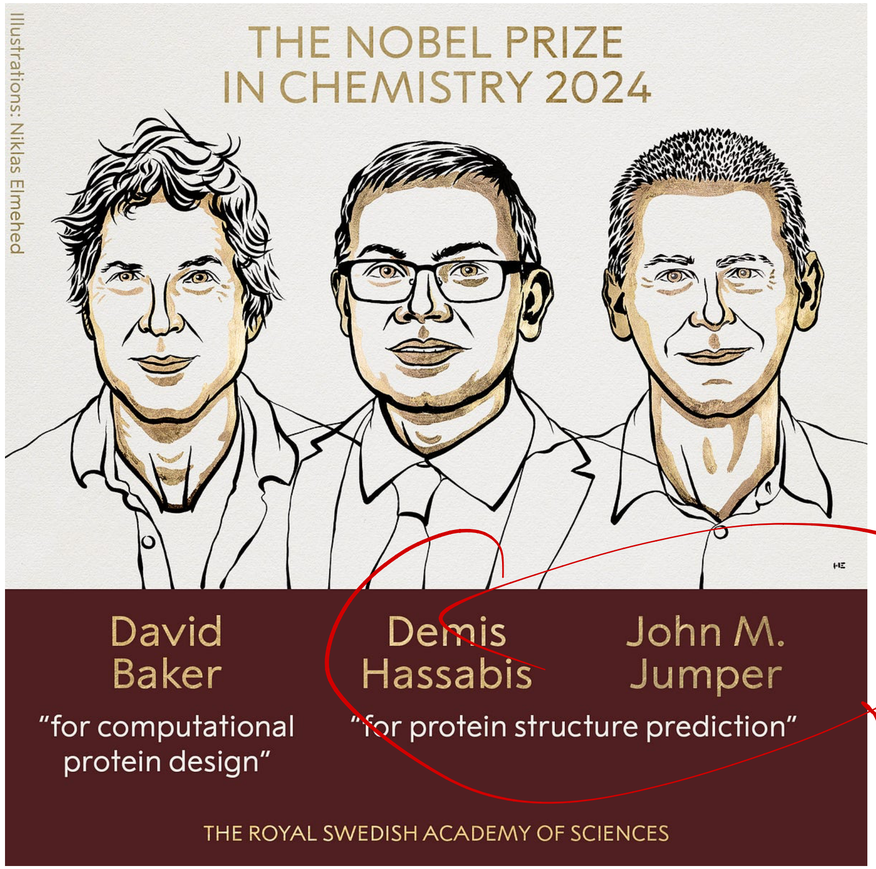
“The bitter lesson is a claim in artificial intelligence that, in the long run, simpler systems that can scale with available computational power will outperform more complex systems that integrate domain-specific human knowledge“

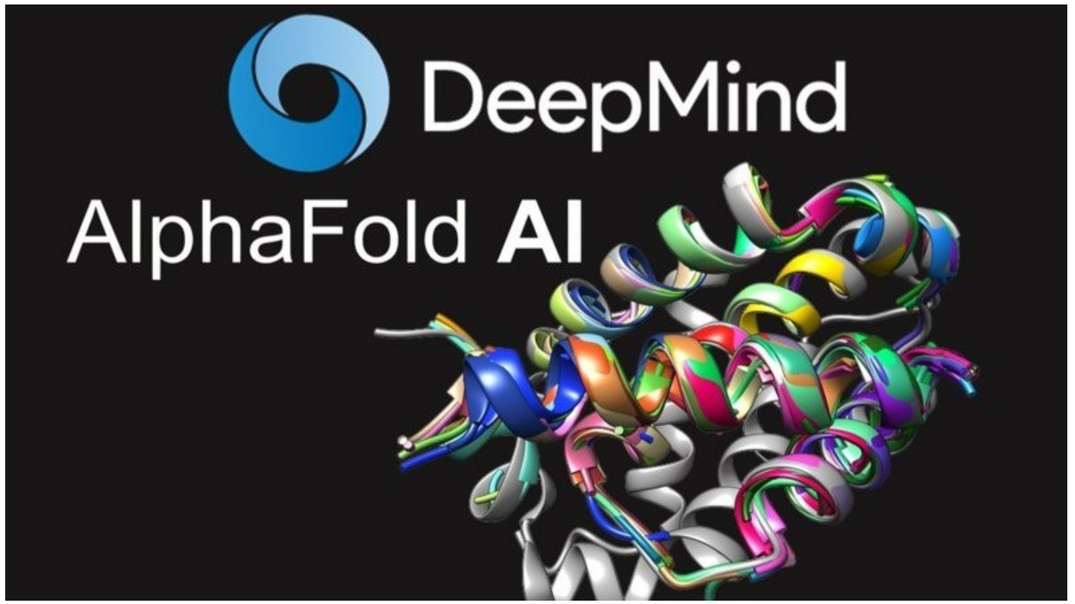
nlp

"Every time I fire a linguist, the performance of the speech recognizer goes up”

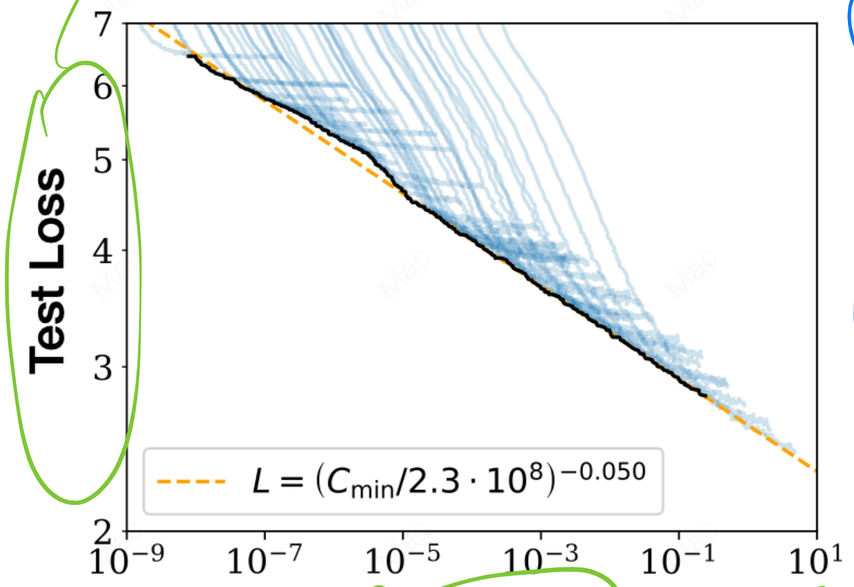
— Frederick Jelinek at IBM Research to develop speech recognizer

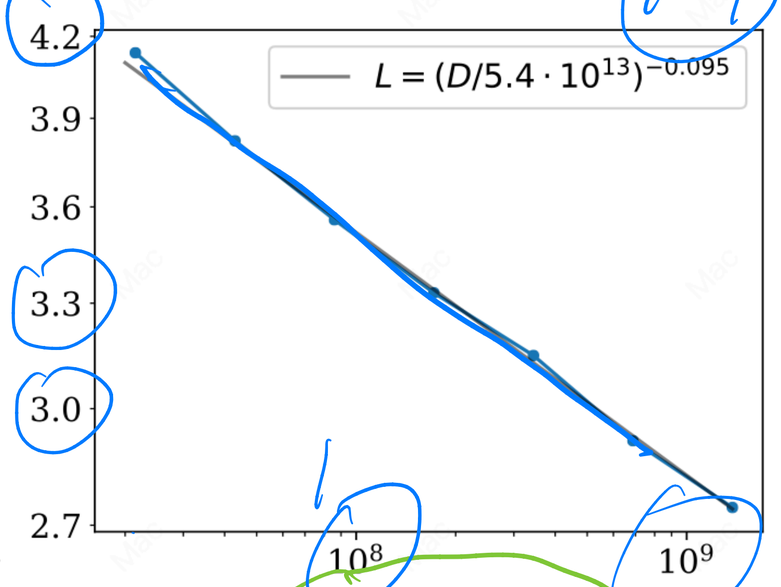
The Bitter Lesson

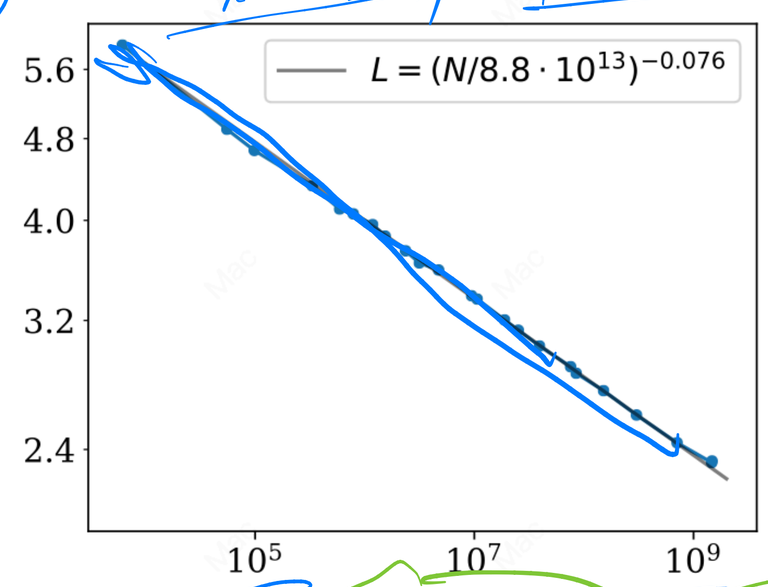




Language Model Scaling Law







Required compute, dataset and parameters need to increase exponentially

Kaplan et al. Scaling Laws for Neural Language Models. 2020

Power Law for Language Model Scaling

Non-embedding parameters N, dataset size D, compute budget C_{min}

For loss to decrease 10% relatively, the N, D, or C_{min} needs to increase around 10 times

Kaplan et al. Scaling Laws for Neural Language Models. 2020

Power Law for Language Model Scaling

Non-embedding parameters N, dataset size D, compute budget C_{min}

1. For models with a limited number of parameters, trained to convergence on sufficiently large datasets:

$$L(N) = (N_c/N)^{\alpha_N}; \quad \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13} \text{ (non-embedding parameters)} \qquad (1.1)$$

For loss to decrease 10% relatively, the N, D, or C_{min} needs to increase around 10 times

Kaplan et al. Scaling Laws for Neural Language Models. 2020

Power Law for Language Model Scaling

Non-embedding parameters N, dataset size D, compute budget C_{min}

1. For models with a limited number of parameters, trained to convergence on sufficiently large datasets:

$$L(N) = (N_c/N)^{\alpha_N}\ ;\ \ \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13} \text{ (non-embedding parameters)} \qquad (1.1)$$

2. For large models trained with a limited dataset with early stopping:

$$L(D) = (D_c/D)^{\alpha_D}\ ;\ \ \alpha_D \sim 0.095, \quad D_c \sim 5.4 \times 10^{13} \text{ (tokens)} \qquad (1.2)$$

For loss to decrease 10% relatively, the N, D, or C_{min} needs to increase around 10 times

Kaplan et al. Scaling Laws for Neural Language Models. 2020

Power Law for Language Model Scaling

Non-embedding parameters N, dataset size D, compute budget C_{min}

1. For models with a limited number of parameters, trained to convergence on sufficiently large datasets:

$$L(N) = (N_c/N)^{\alpha_N}\ ;\ \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13} \text{ (non-embedding parameters)} \qquad (1.1)$$

2. For large models trained with a limited dataset with early stopping:

$$L(D) = (D_c/D)^{\alpha_D}\ ;\ \alpha_D \sim 0.095, \quad D_c \sim 5.4 \times 10^{13} \text{ (tokens)} \qquad (1.2)$$

3. When training with a limited amount of compute, a sufficiently large dataset, an optimally-sized model, and a sufficiently small batch size (making optimal[3] use of compute):

$$L(C_{\min}) = (C_c^{\min}/C_{\min})^{\alpha_C^{\min}}\ ;\ \alpha_C^{\min} \sim 0.050, \quad C_c^{\min} \sim 3.1 \times 10^8 \text{ (PF-days)} \qquad (1.3)$$

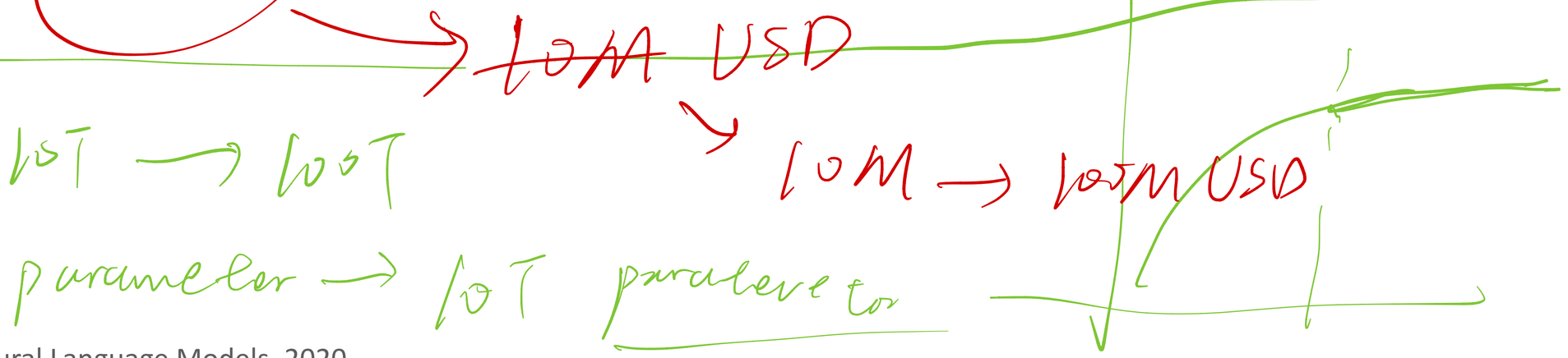
For loss to decrease 10% relatively, the N, D, or C_{min} needs to increase around 10 times

Kaplan et al. Scaling Laws for Neural Language Models. 2020

Power Law for Language Model Scaling

Power law is exciting because it means as we scale up, the performance can continue getting better

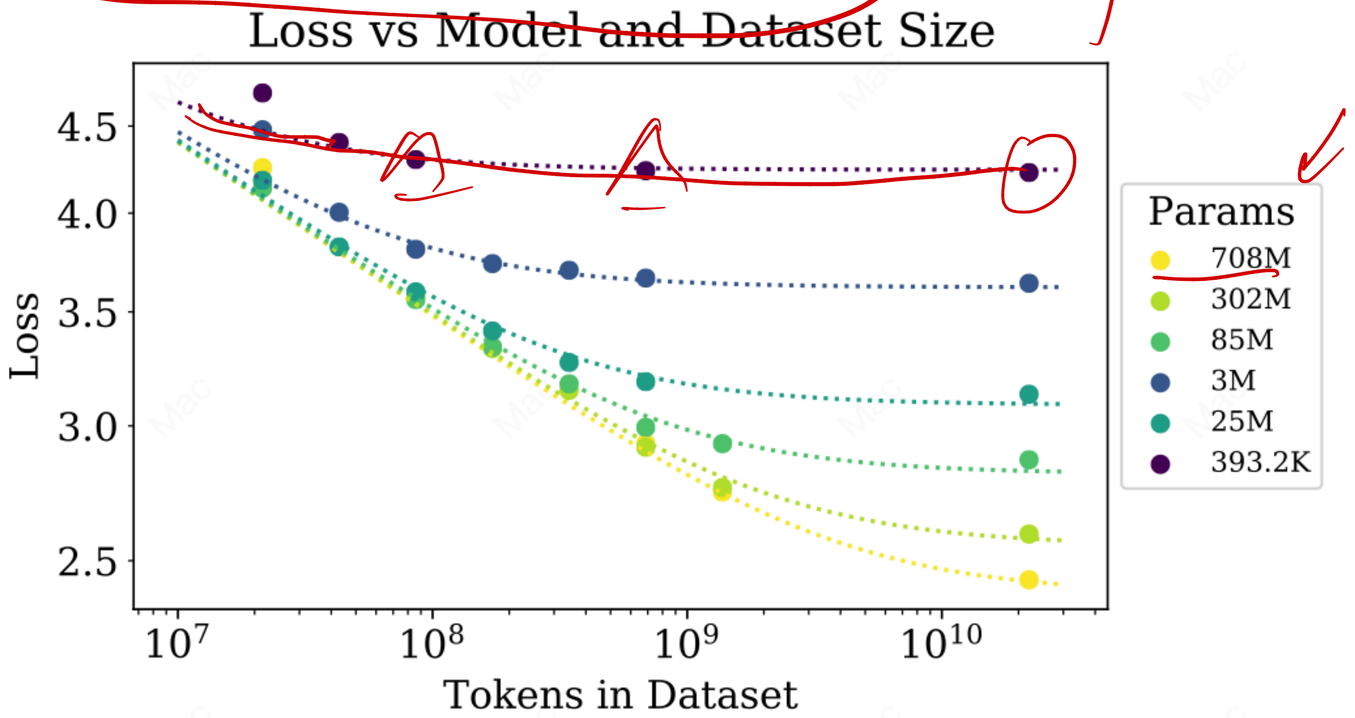
Power law is also pessimistic, because it's harder and harder to increase model size, data, compute exponential.



Kaplan et al. Scaling Laws for Neural Language Models. 2020

Loss is Predictable

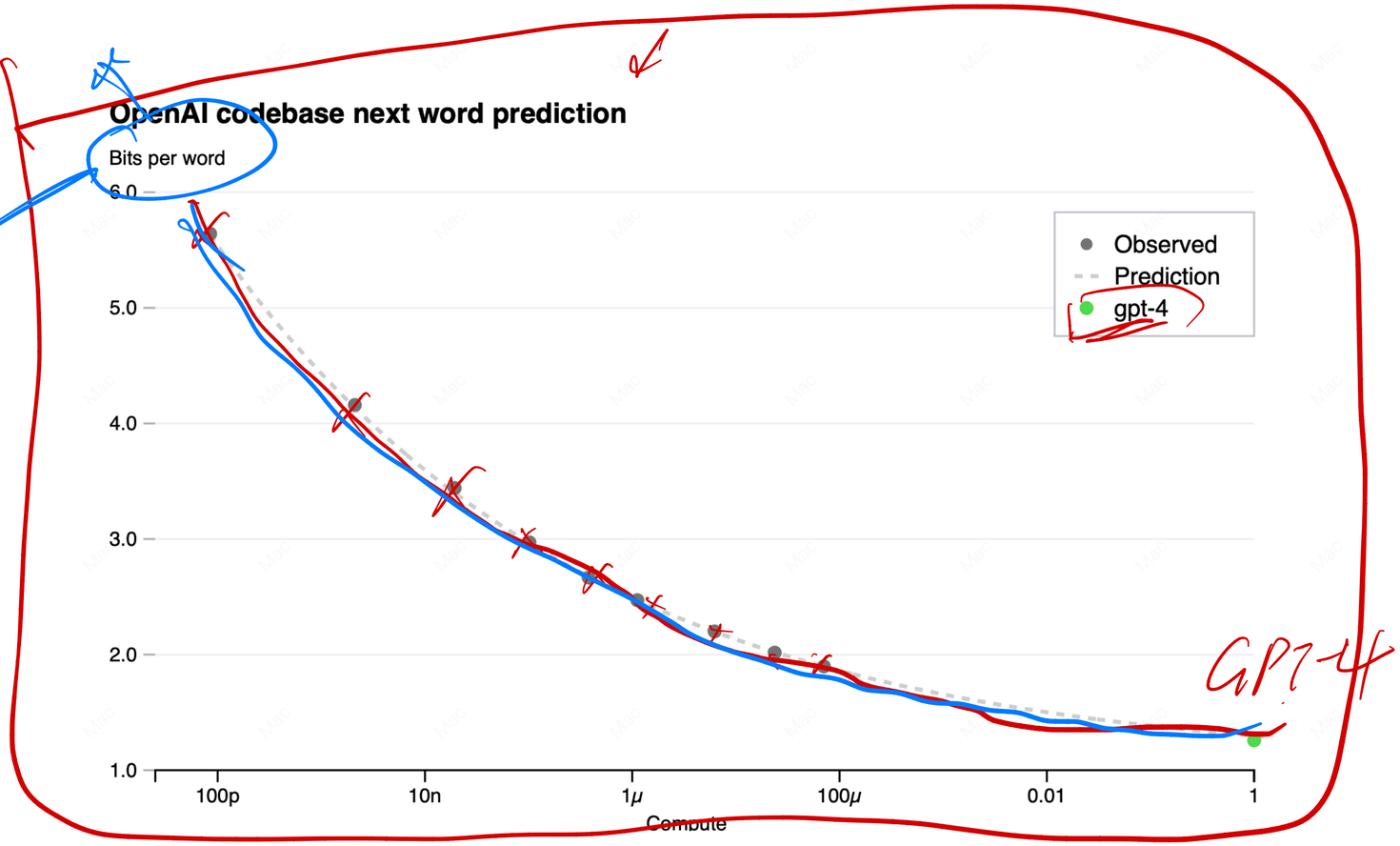
$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$



Dashed lines show the predicted curve

Kaplan et al. Scaling Laws for Neural Language Models. 2020

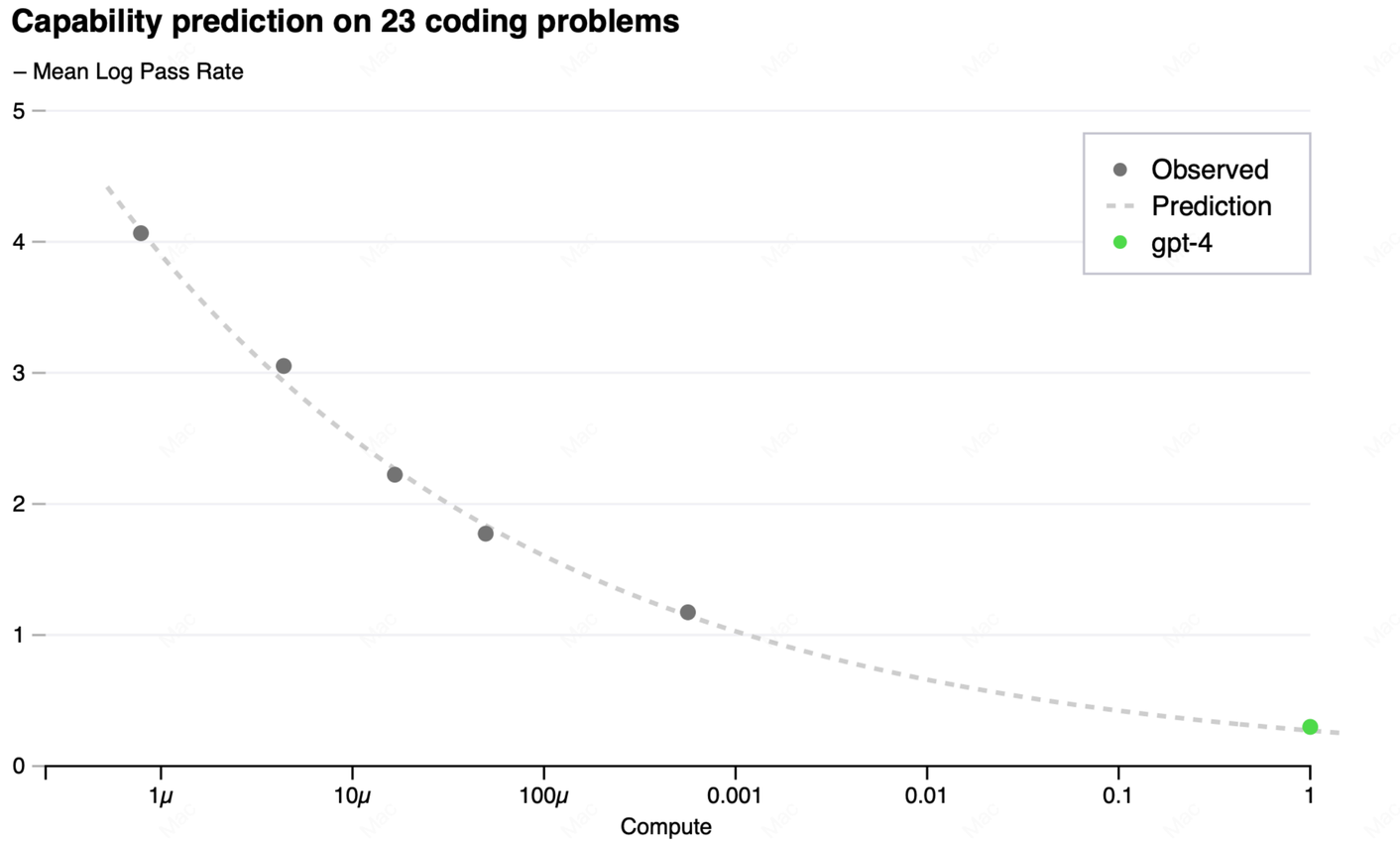
Scaling Laws for Prediction



Before launching large-scale training, we can predict its performance in advance by fitting smaller-scale experiments!

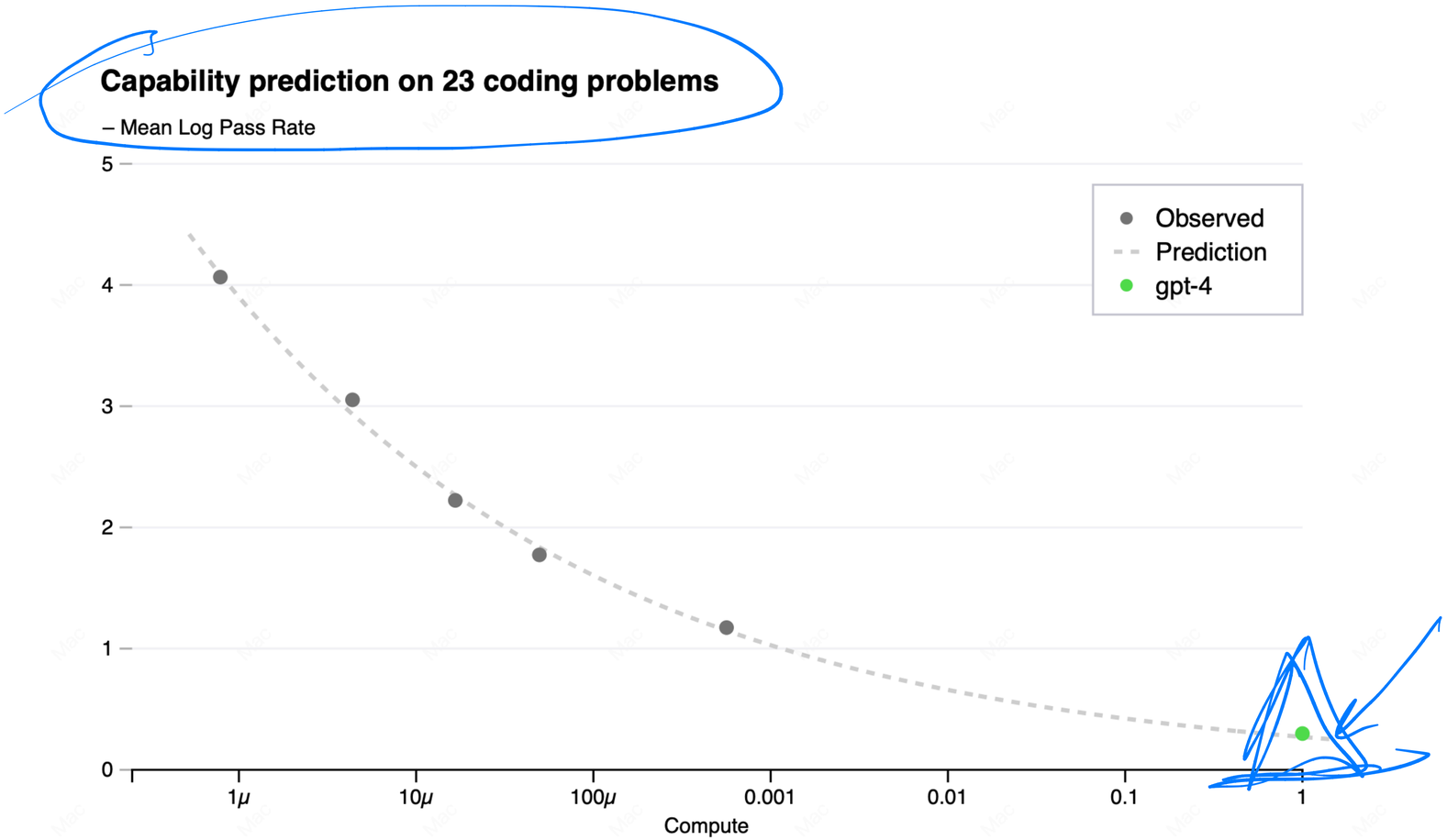
OpenAI. GPT-4 Technical Report. 2024

Scaling Laws for Prediction



OpenAI. GPT-4 Technical Report. 2024

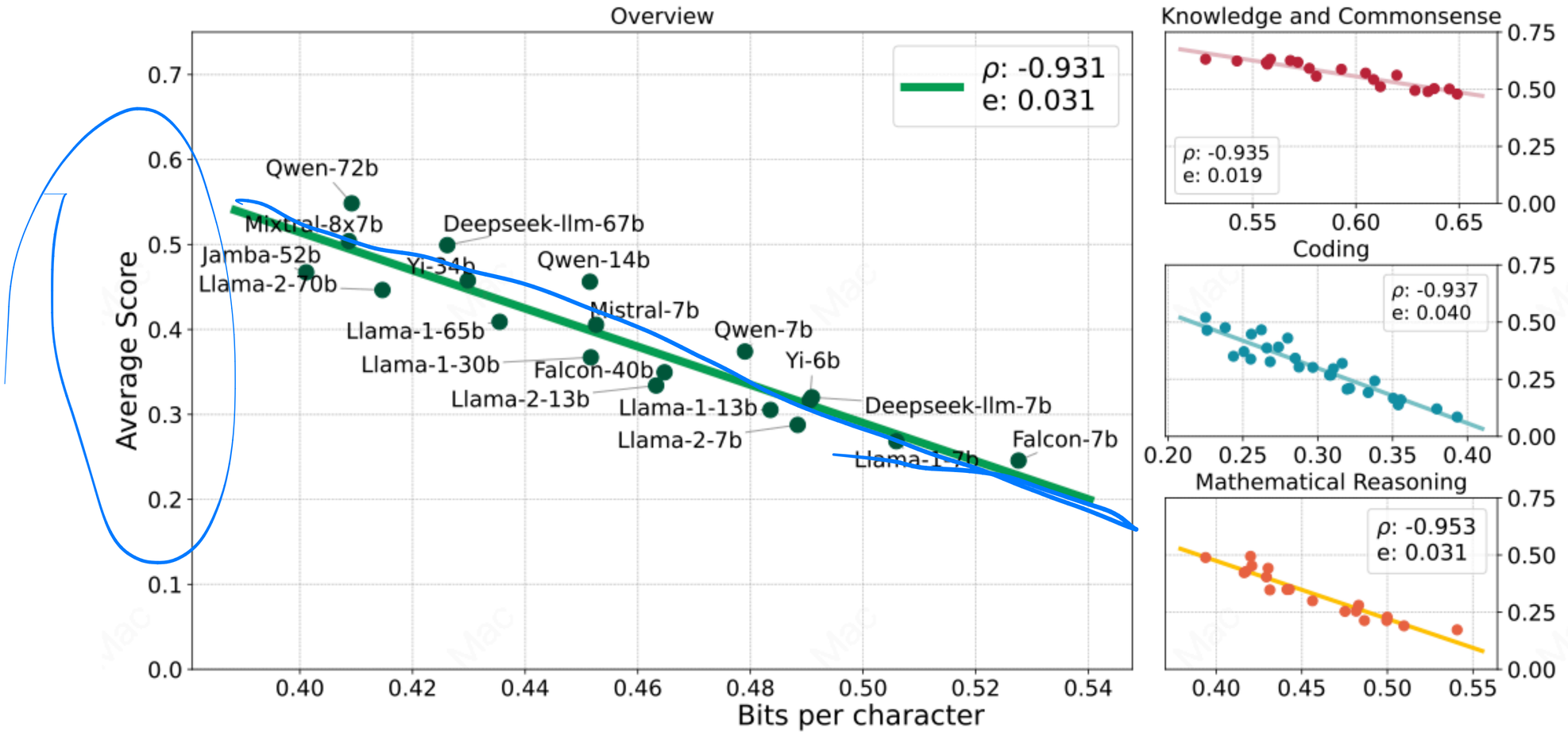
Scaling Laws for Prediction



Downstream task performance can be predicted relatively well

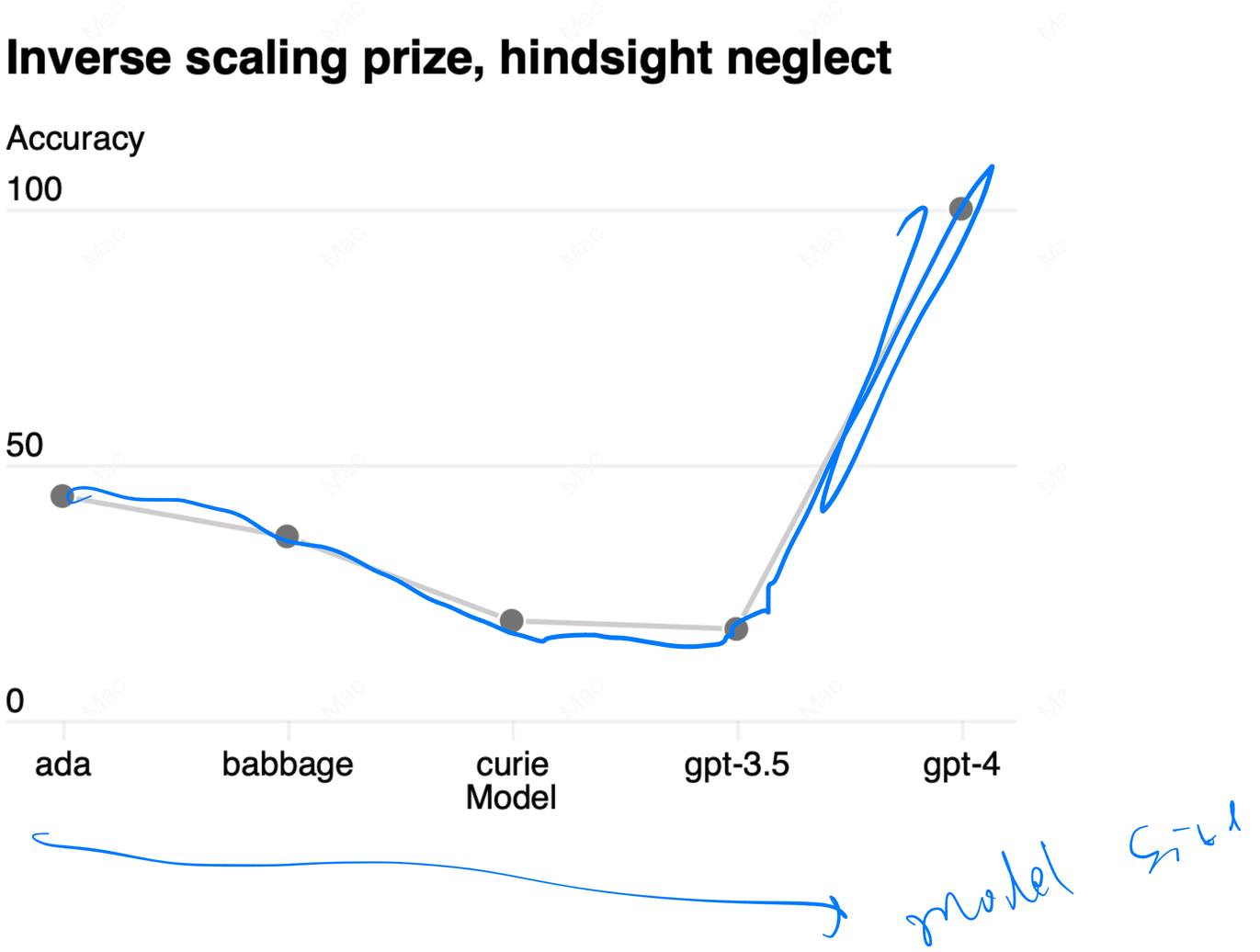
OpenAI. GPT-4 Technical Report. 2024

Close Relationship between Pretraining Loss and Downstream Performance

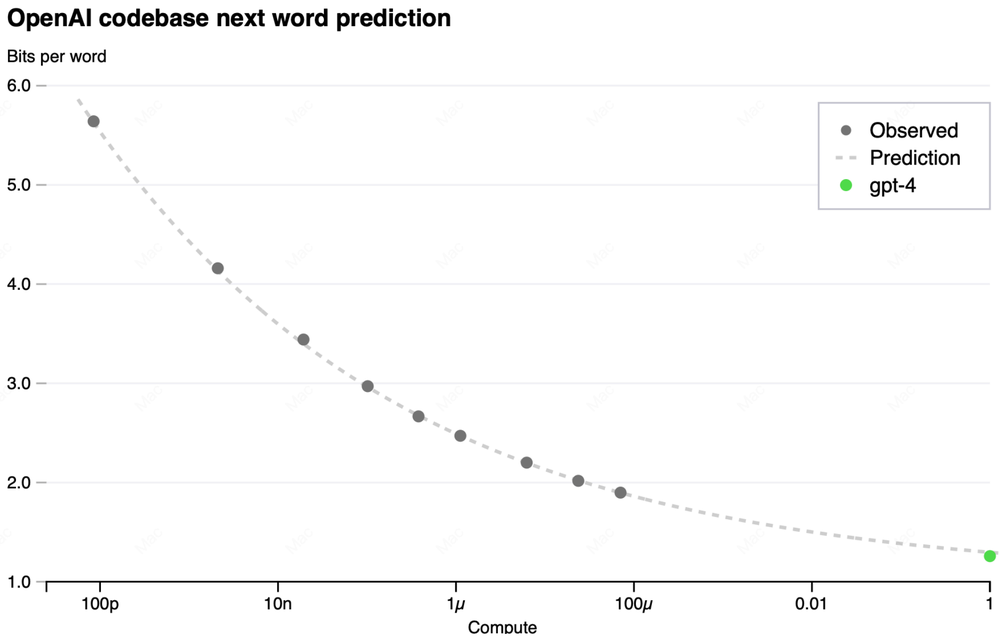


Huang et al. Compression Represents Intelligence Linearly. 2024

Certain Abilities Remain Hard to Predict

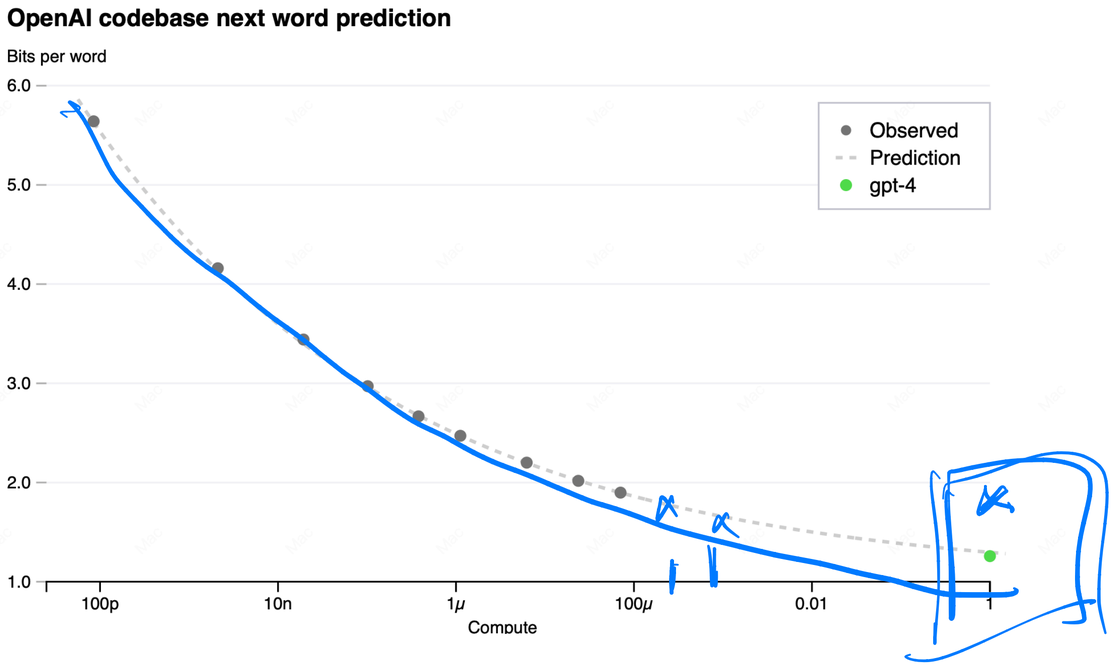


Scaling Laws for Prediction



OpenAI. GPT-4 Technical Report. 2024

Scaling Laws for Prediction



Scaling law is very useful as it can be used to check whether our large-scale run is as expected in the middle

3 months 1M USD

OpenAI. GPT-4 Technical Report. 2024

Scaling Laws for Prediction

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

Scaling Laws for Prediction

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

If everything is destined before training, then what we is pertaining studying?

Scaling Laws for Prediction

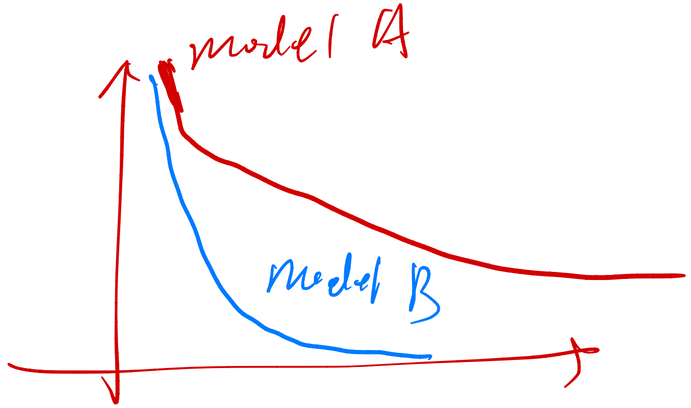
$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$

If everything is destined before training, then what we is pertaining studying?

Is this a dead end and just requiring too many resources?

Scaling Laws for Prediction

$$L(N, D) = \left[\left(\frac{N_c}{N} \right)^{\frac{\alpha_N}{\alpha_D}} + \frac{D_c}{D} \right]^{\alpha_D}$$



If everything is destined before training, then what we is pertaining studying?

Is this a dead end and just requiring too many resources?

Even though the form is power law, but the coefficient can be different with different architectures and data quality. We want to find high data quality and better model arch so that it scales more efficiently

Scaling Laws for Resource Allocation

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

Chinchilla law

Scaling Laws for Resource Allocation

Given a fixed FLOPs budget, how should one trade-off model size and the number of training tokens?

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

Scaling Laws for Resource Allocation

Given a fixed FLOPs budget, how should one trade-off model size and the number of training tokens?

In practice, we are often FLOPs bounded, like we can only use 1000 H100s for 3 months.

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

Scaling Laws for Resource Allocation

Given a fixed FLOPs budget, how should one trade-off model size and the number of training tokens?

In practice, we are often FLOPs bounded, like we can only use 1000 H100s for 3 months.

But this situation is changing, in the future we may be data bounded

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

Scaling Laws for Resource Allocation

Given a fixed FLOPs budget, how should one trade-off model size and the number of training tokens?

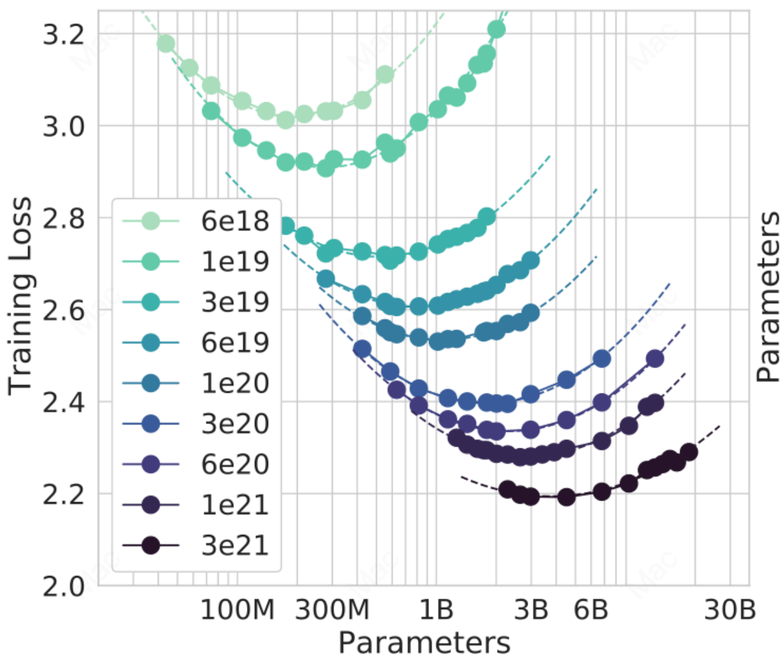
In practice, we are often FLOPs bounded, like we can only use 1000 H100s for 3 months.

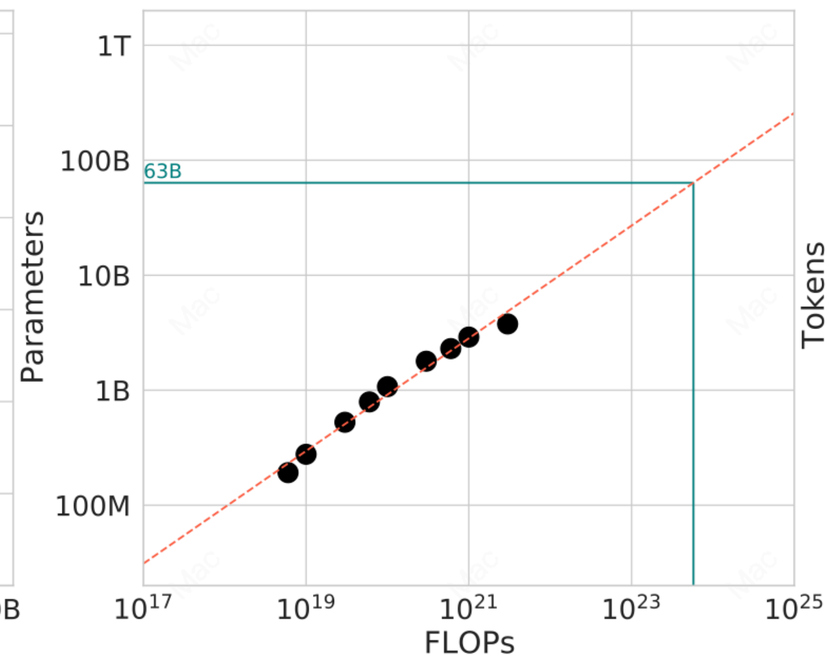
But this situation is changing, in the future we may be data bounded

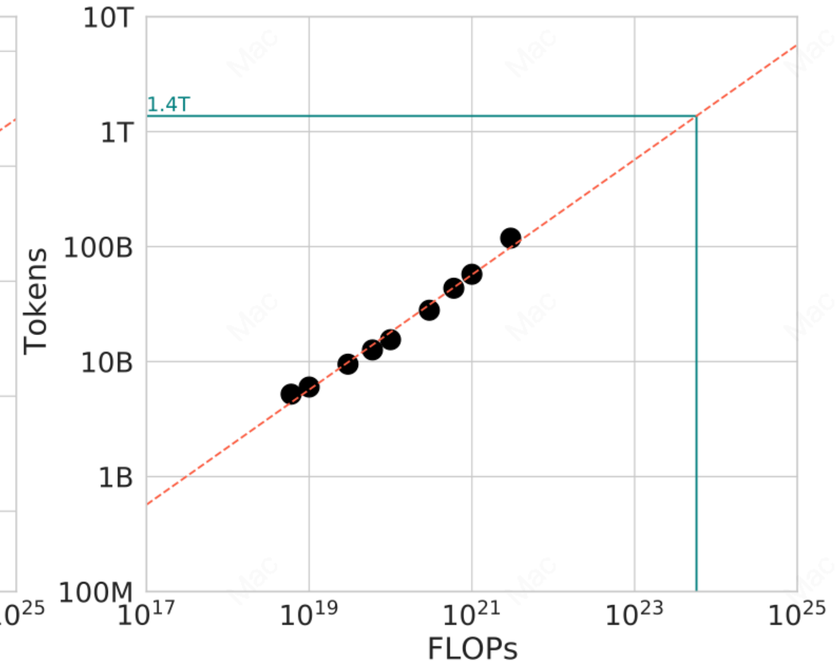
$$N_{opt}(C), D_{opt}(C) = \underset{N,D \text{ s.t. } \mathrm{FLOPs}(N,D)=C}{\operatorname{argmin}} L(N, D).$$

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

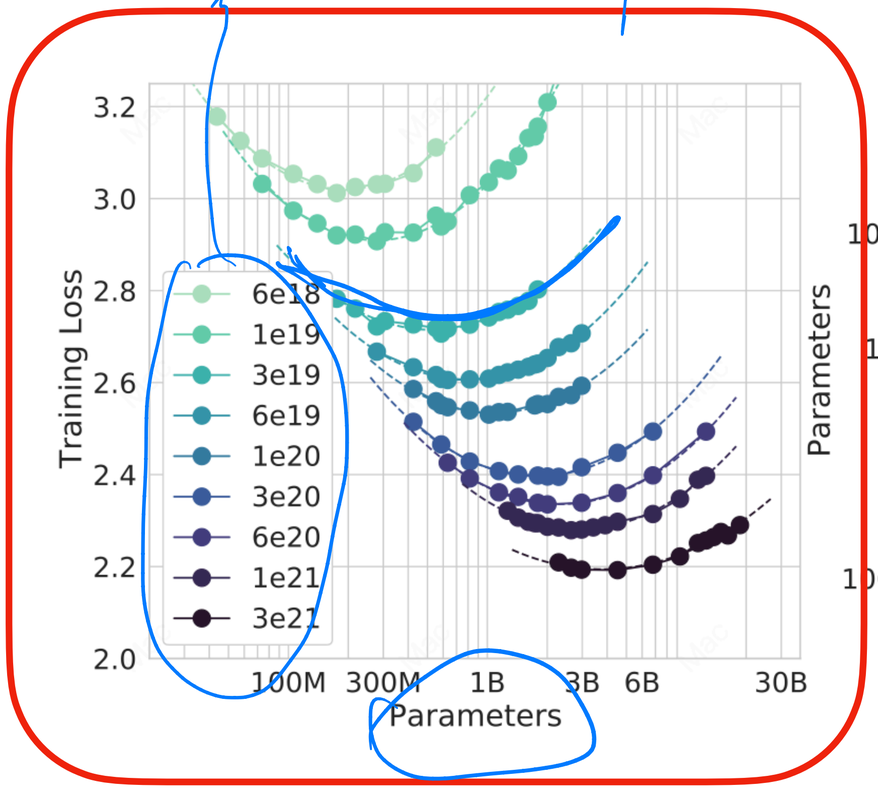
Scaling Laws for Resource Allocation

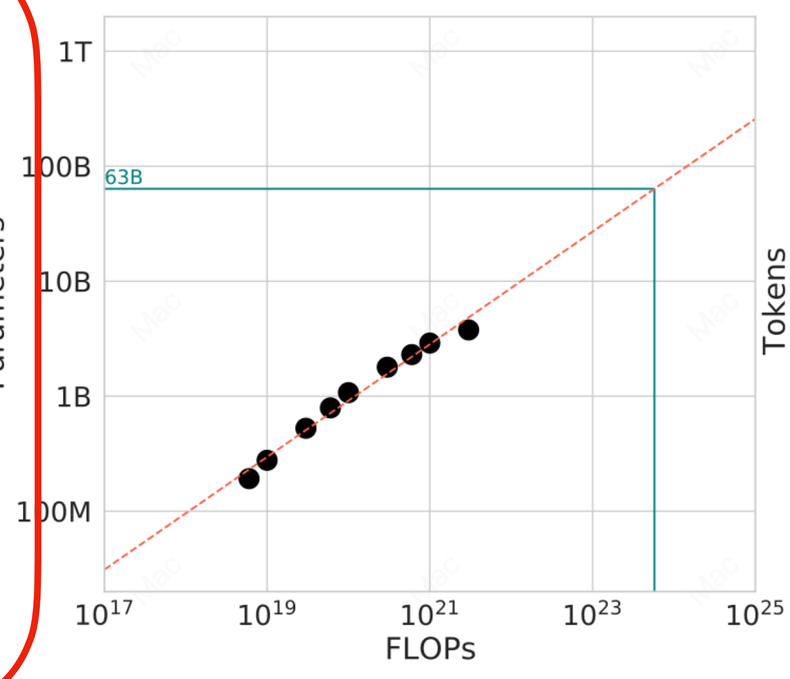


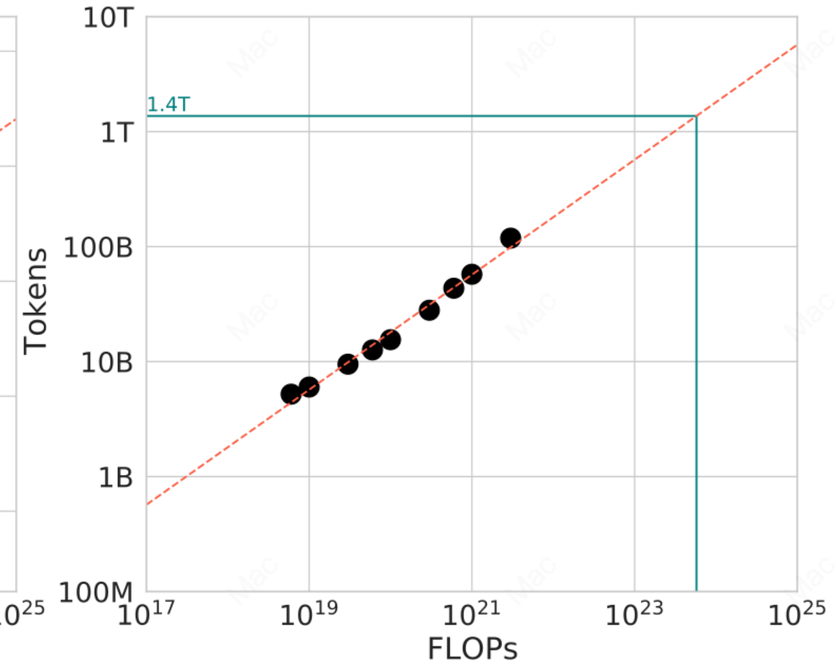




Scaling Laws for Resource Allocation

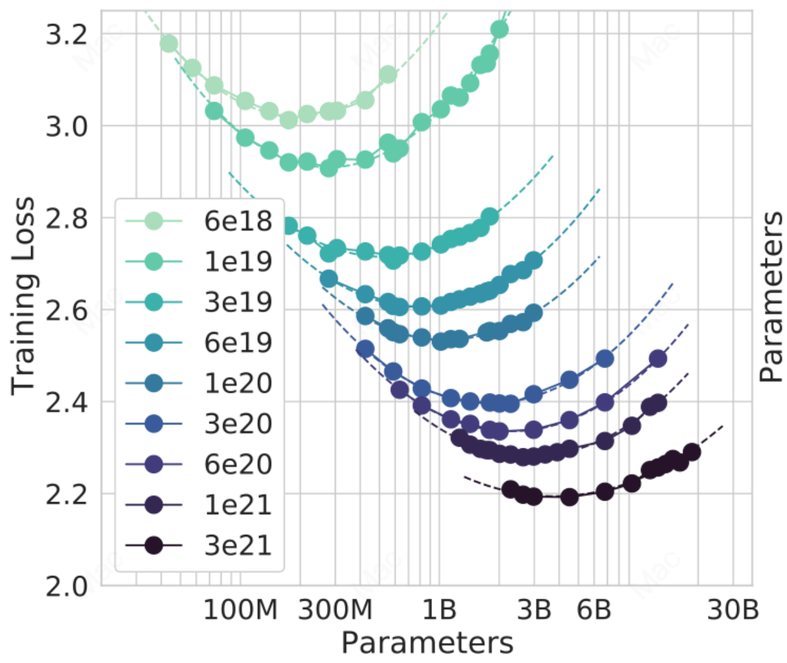


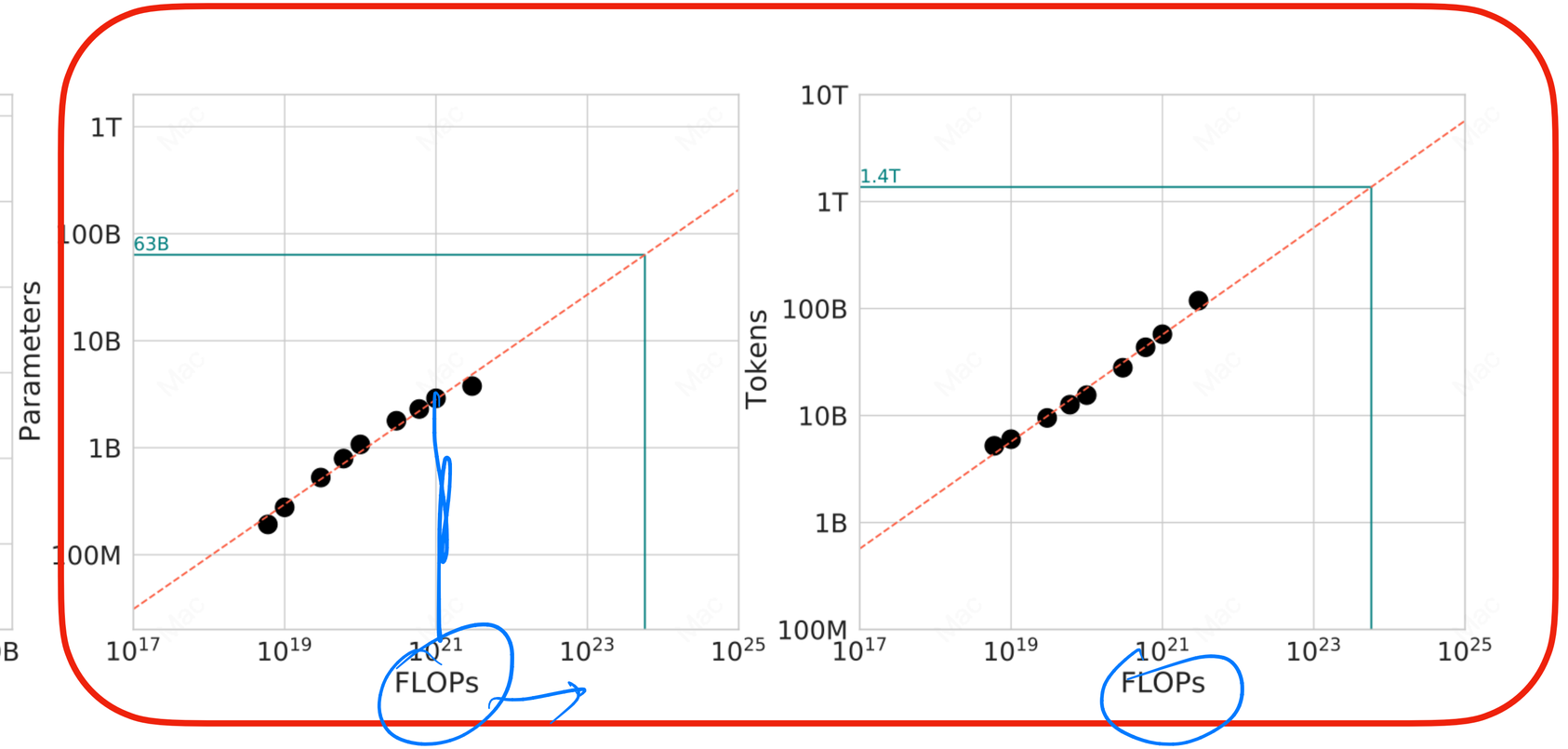




Varying model parameters for the same FLOPs, there is an optimal model size, larger is not always better (because the model is trained with less data)

Scaling Laws for Resource Allocation





The optimal model size and training token size, they should increase TOGETHER as we have more compute

Scaling Laws for Resource Allocation

Parameters	FLOPs	FLOPs (in *Gopher* unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

Chinchilla Optimal Allocation

Hoffmann et al. Training Compute-Optimal Large Language Models. 2022

Thank You!