CS-461

Foundation Models and Generative AI

Introduction and Overview

Charlotte Bunne, Fall Semester 2025/26

Who am I?

2019 - 2023

PhD
Computer Science

ETH and
Broad Institute of MIT and Harvard

Andreas Krause — ETH
Marco Cuturi — Apple
Anne Carpenter — Broad Institute of MIT and Harvard
Shantanu Singh — Broad Institute of MIT and Harvard

2023 - 2024

PostDoc
Computer Science and Biology

Stanford University
and Genentech

Aviv Regev — Genentech
Jure Leskovec — Stanford University

since 2024

Tenure-Track Assistant Professor
Joint Affiliation in
Computer Science and Life Sciences

EPFL

ARTIFICIAL INTELLIGENCE
IN MOLECULAR MEDICINE

Charlotte Bunne

What are foundation models?

Foundation Models and Generative AI

How are these connected?

What is generative AI?

The Bitter Lesson

Richard Sutton, 'The Bitter Lesson' (2019).
Published online at 'http://www.incompleteideas.net/IncIdeas/BitterLesson.html'.

> 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.

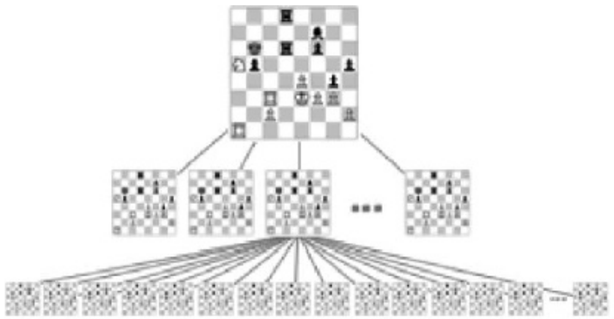
Richard Sutton
(2019)

The Bitter Lesson: Historic Patterns

1997

`Deep Blue`
wins against
Garry Kasparov

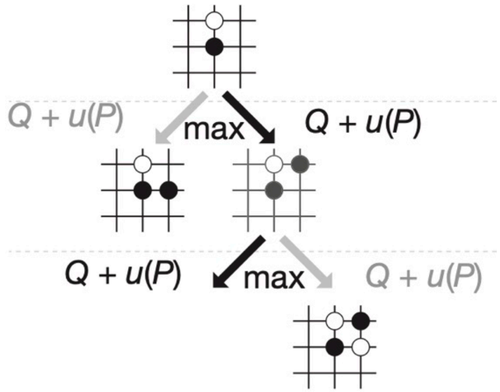
massively parallelized tree searches



2016

`AlphaGo`
wins agains
Lee Sedol

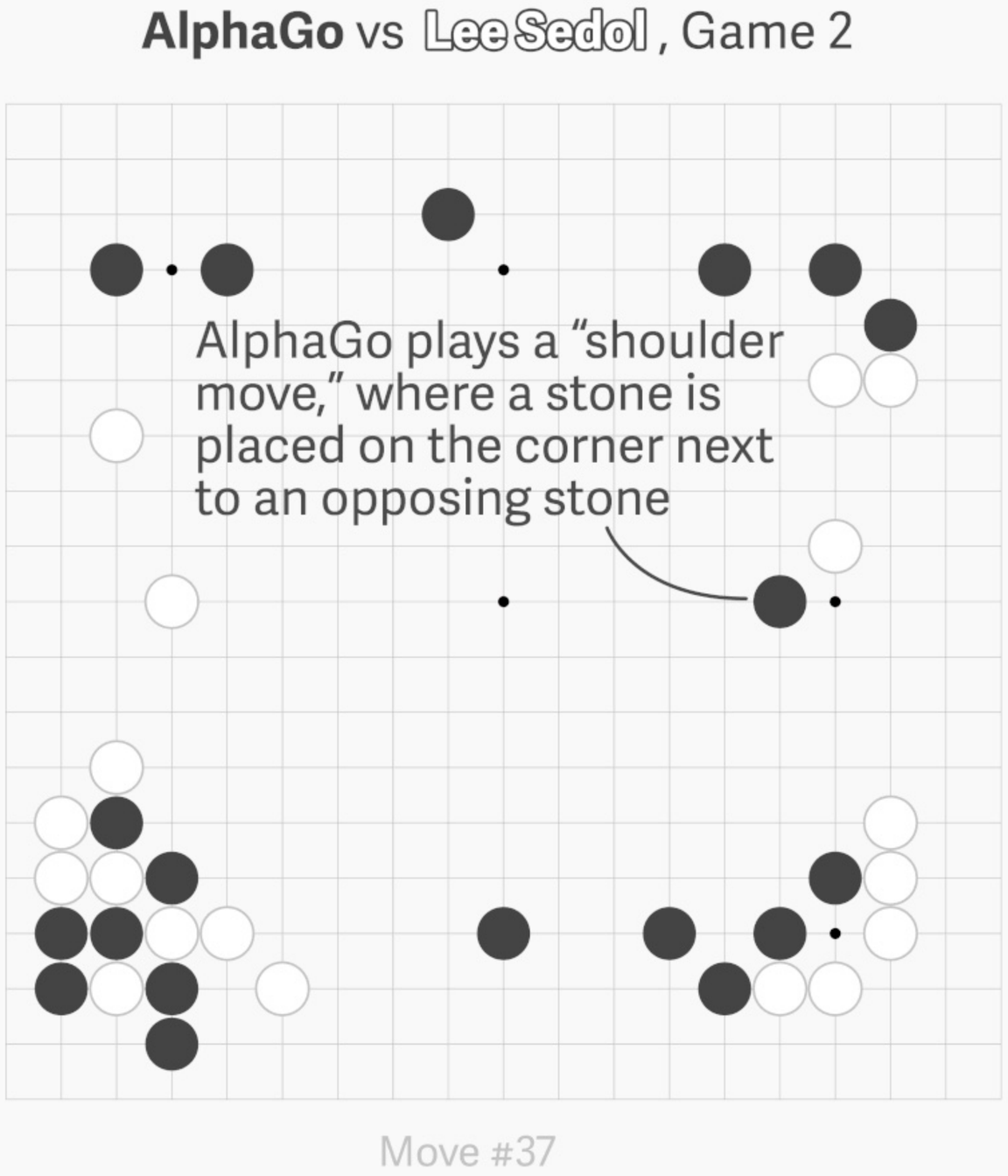
deep learning-powered search and self play



Move 37

AlphaGo's Move 37 was surprising and creative:

The system had learned to make moves that seemed counterintuitive but were strategically sound.

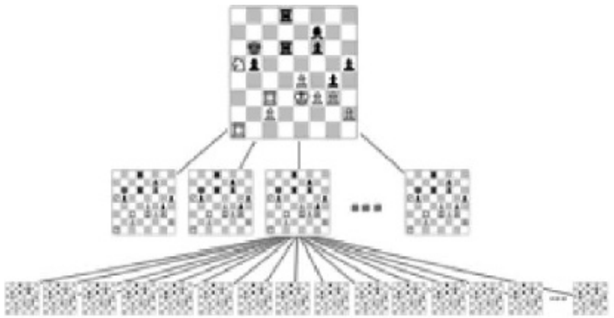


The Bitter Lesson: Historic Patterns

1997

`Deep Blue`
wins against
Garry Kasparov

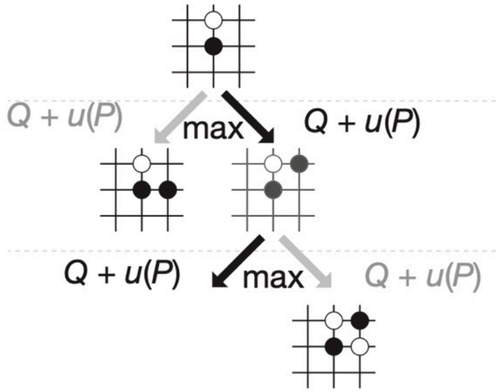
massively parallelized tree searches



2016

`AlphaGo`
wins agains
Lee Sedol

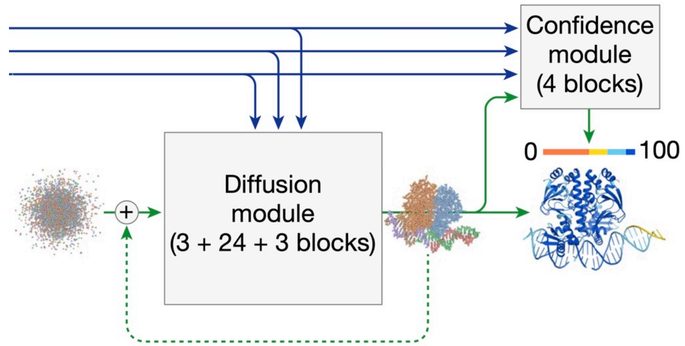
deep learning-powered search and self play



2024

`AlphaFold 3`
sets new record
in protein folding

removed biological priors of previous AlphaFold architecture



… and we find **similar historic patterns** in **computer vision** and **natural language processing.**

The Bitter Lesson: Search and Learning

> Search and learning are the two most important classes of techniques for utilizing massive amounts of computation in AI research.

> The bitter lesson is based on the historical observations that

1. AI researchers have often tried to build knowledge into their agents,
2. this always helps in the short term, and is personally satisfying to the researcher, but
3. in the long run it plateaus and even inhibits further progress, and
4. breakthrough progress eventually arrives by an opposing approach based on scaling computation by search and learning.

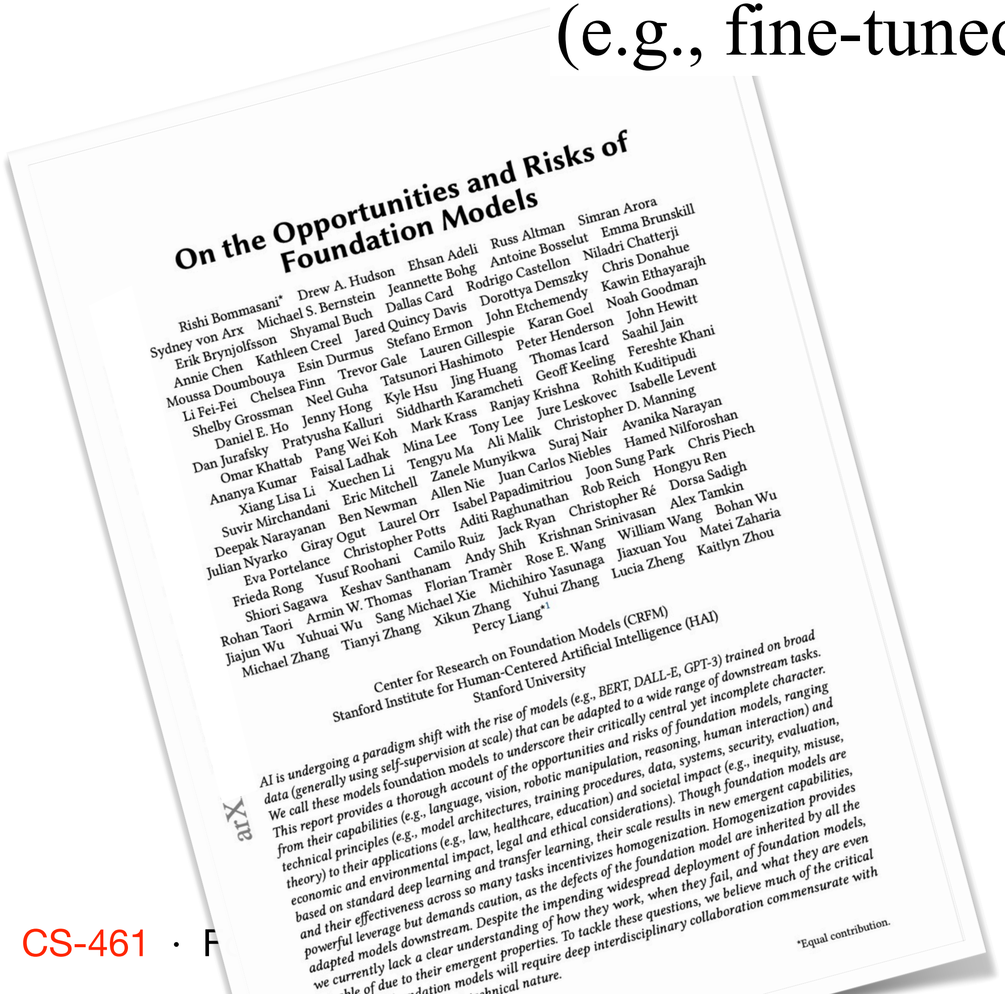
The Bitter Lesson: General Purpose Methods

> “ One thing that should be learned from the bitter lesson is the **great power of general purpose methods**, of **methods that continue to scale** with increased computation even as the available computation becomes very great.

What is a foundation model?

> “ A foundation model is any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks.

– On the Opportunities and Risks of Foundation Models
Bommasani et al., (2021)



On the Opportunities and Risks of Foundation Models

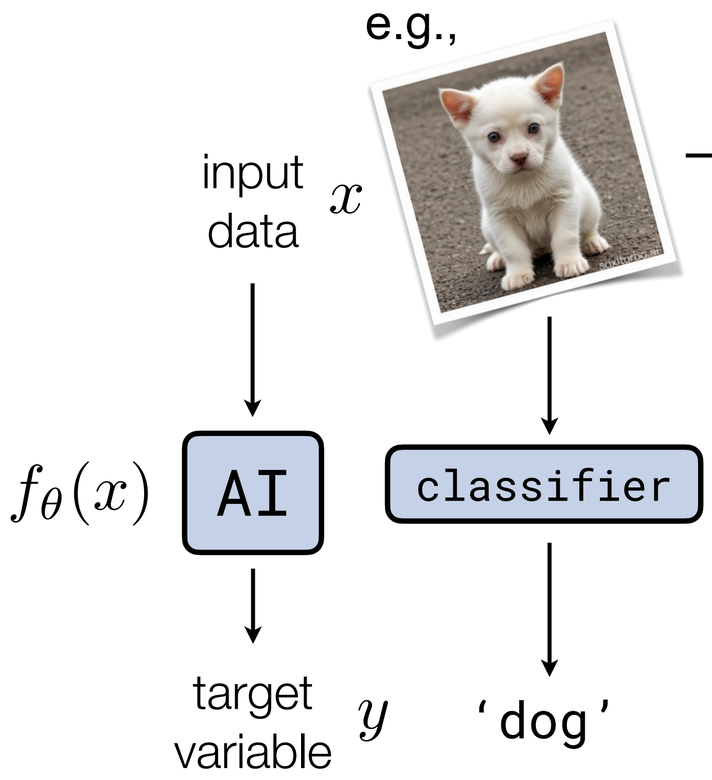
Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*†

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.

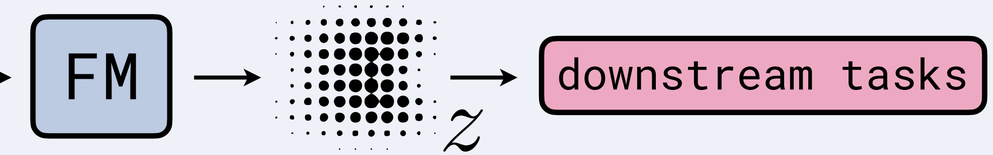
*Equal contribution.

From Task-Specific to General-Purpose



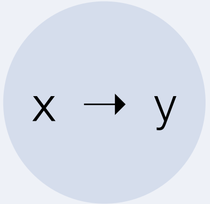
supervised training

1 task → 1 dataset → 1 model



The New Dogma

- Train on massive, unlabeled data using self-supervised or unsupervised objectives.
- Learn general representations of the data that are used across countless downstream tasks.



unsupervised or **self-supervised training**

1 general model → many tasks

The Scaling Hypothesis

Gwern Branwen, 'The Scaling Hypothesis' (2021).
Published online at 'https://gwern.net/scaling-hypothesis'.

> The strong scaling hypothesis is that, once we find a scalable architecture like self-attention or convolutions, which like the brain can be applied fairly uniformly [...], we can simply train ever larger [neural networks] and ever more sophisticated behavior will emerge naturally as the easiest way to optimize for all the tasks and data.

Branwen et al., (2021)

Lecture 2: Learning at Scale

Floating Point Operations Per Second (FLOPS)

FLOPS

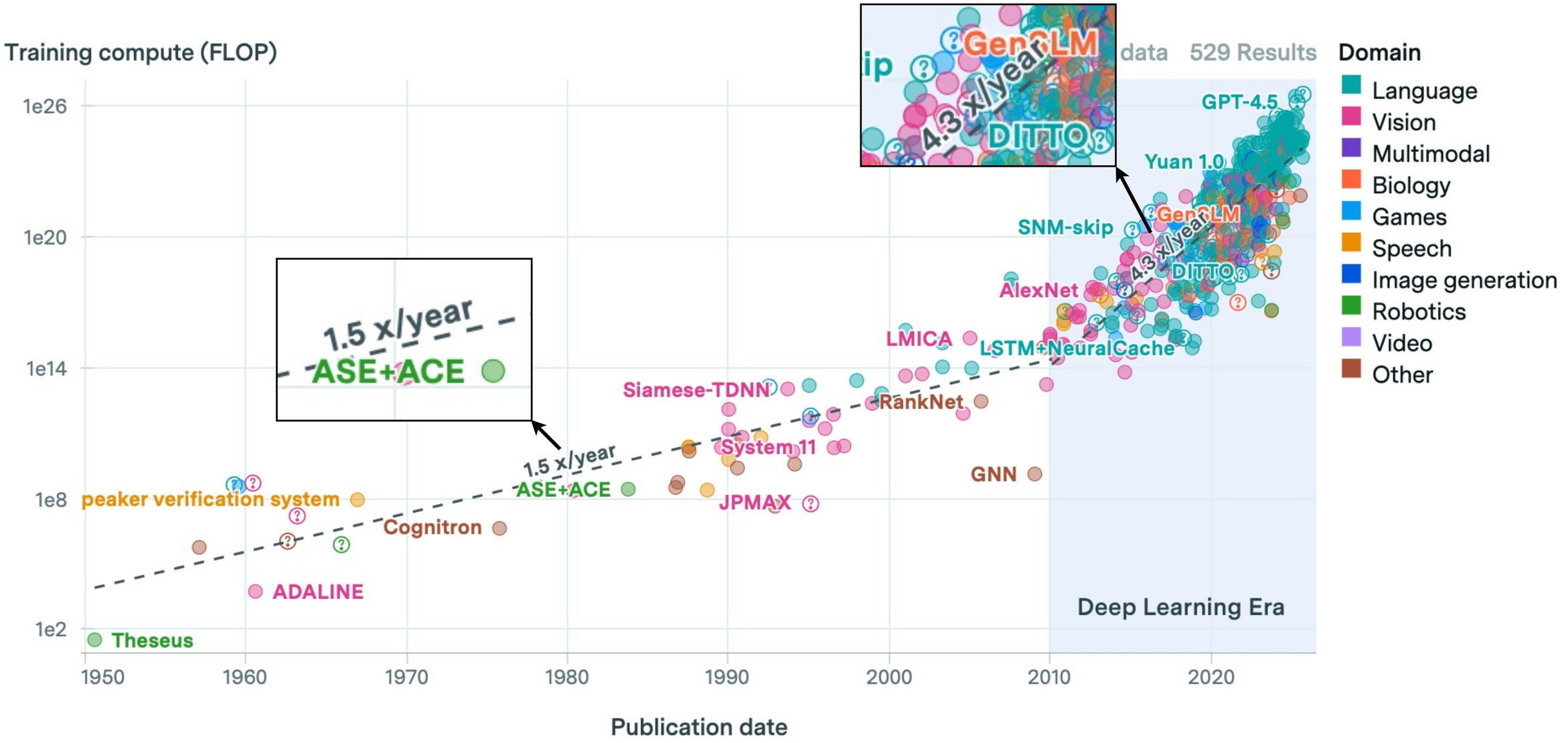
Measure of computational performance that counts how many floating-point arithmetic operations a computer can perform in one second.

Units and scale:

```
FLOPS  = 1 operation per second
KFLOPS = 1,000 FLOPS (kiloFLOPS)
MFLOPS = 1 million FLOPS (megaFLOPS)
GFLOPS = 1 billion FLOPS (gigaFLOPS)
TFLOPS = 1 trillion FLOPS (teraFLOPS)
PFLOPS = 1 quadrillion FLOPS (petaFLOPS)
EFLOPS = 1 quintillion FLOPS (exaFLOPS)
```

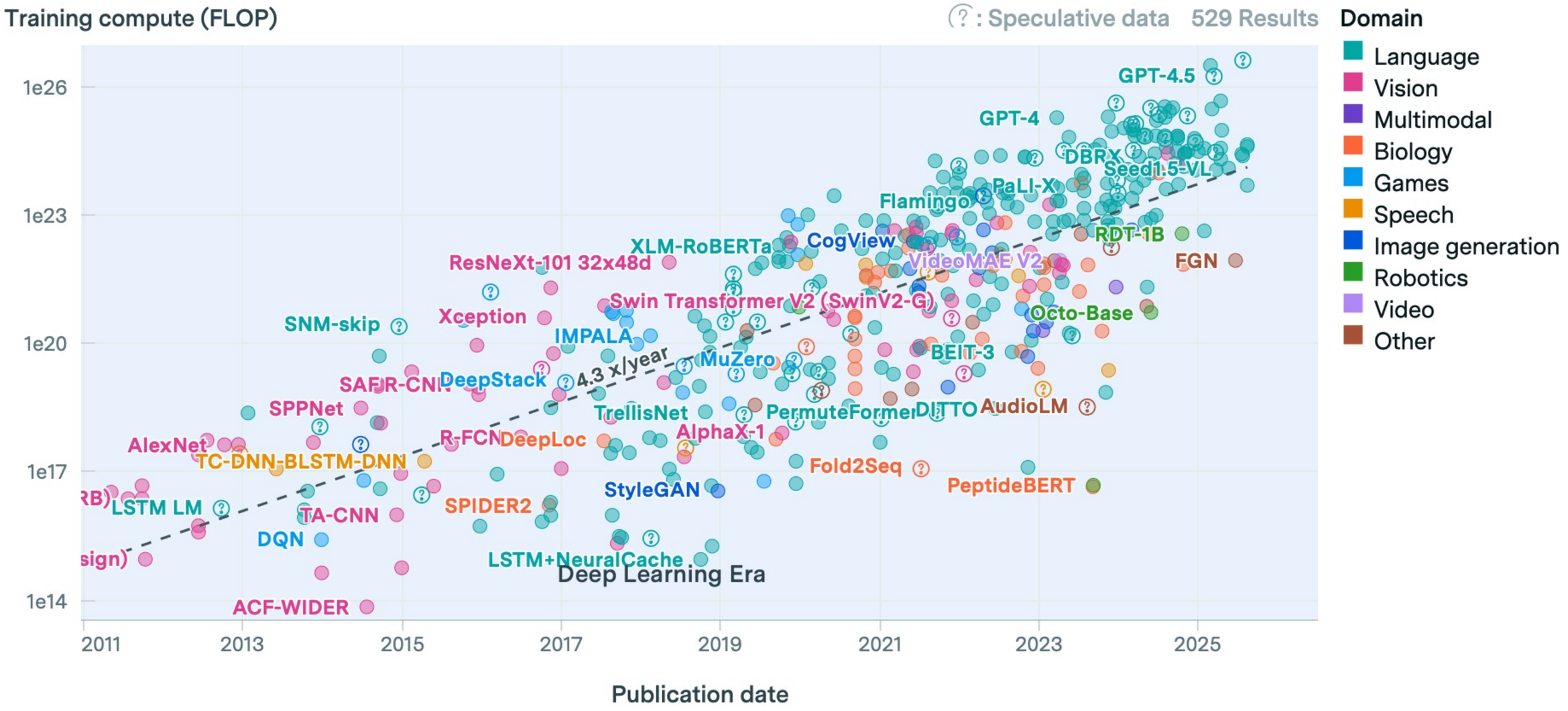
AI Models Over Time

Epoch AI, 'Data on AI Models'.
Published online at 'https://epoch.ai/data/ai-models'.



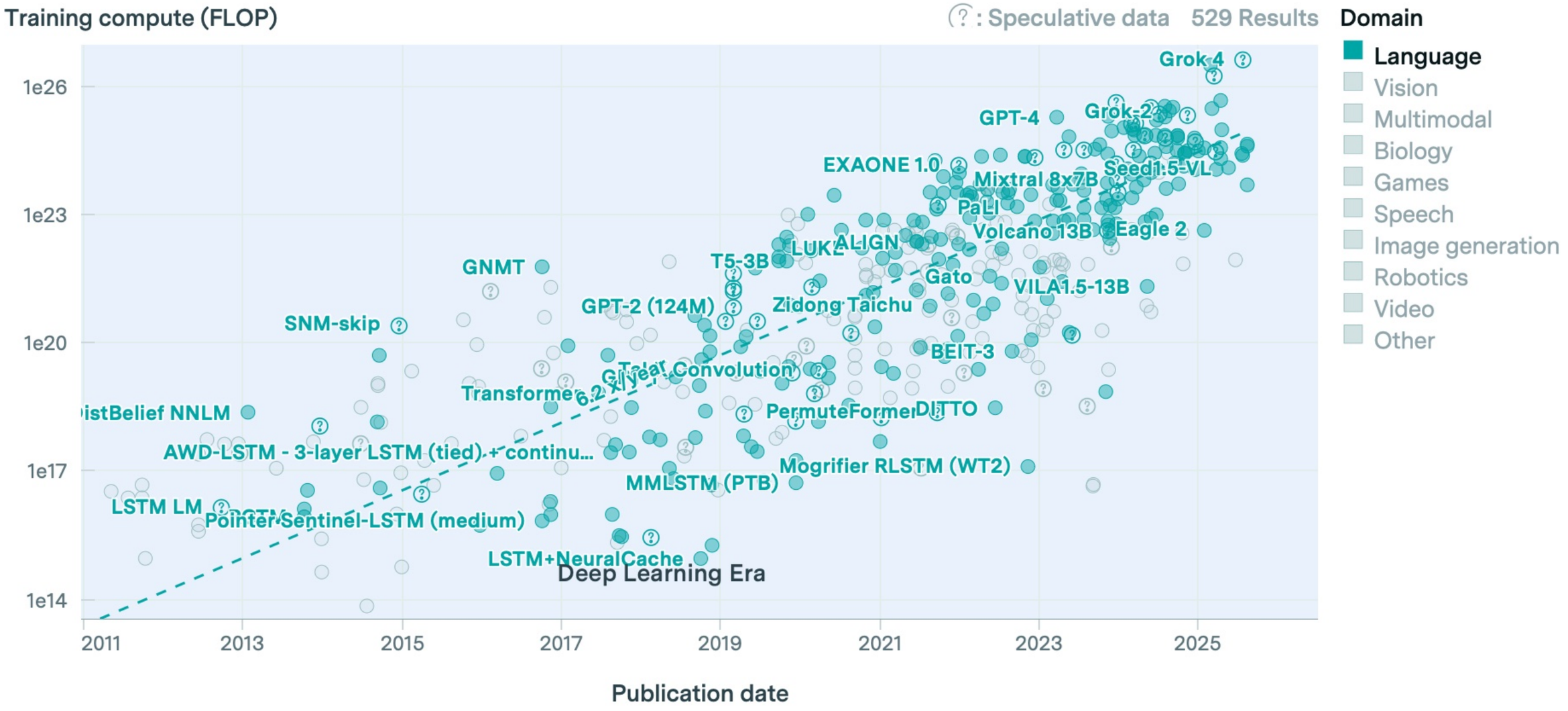
AI Models Over Time

Epoch AI, ‘Data on AI Models’.
Published online at ‘https://epoch.ai/data/ai-models'.



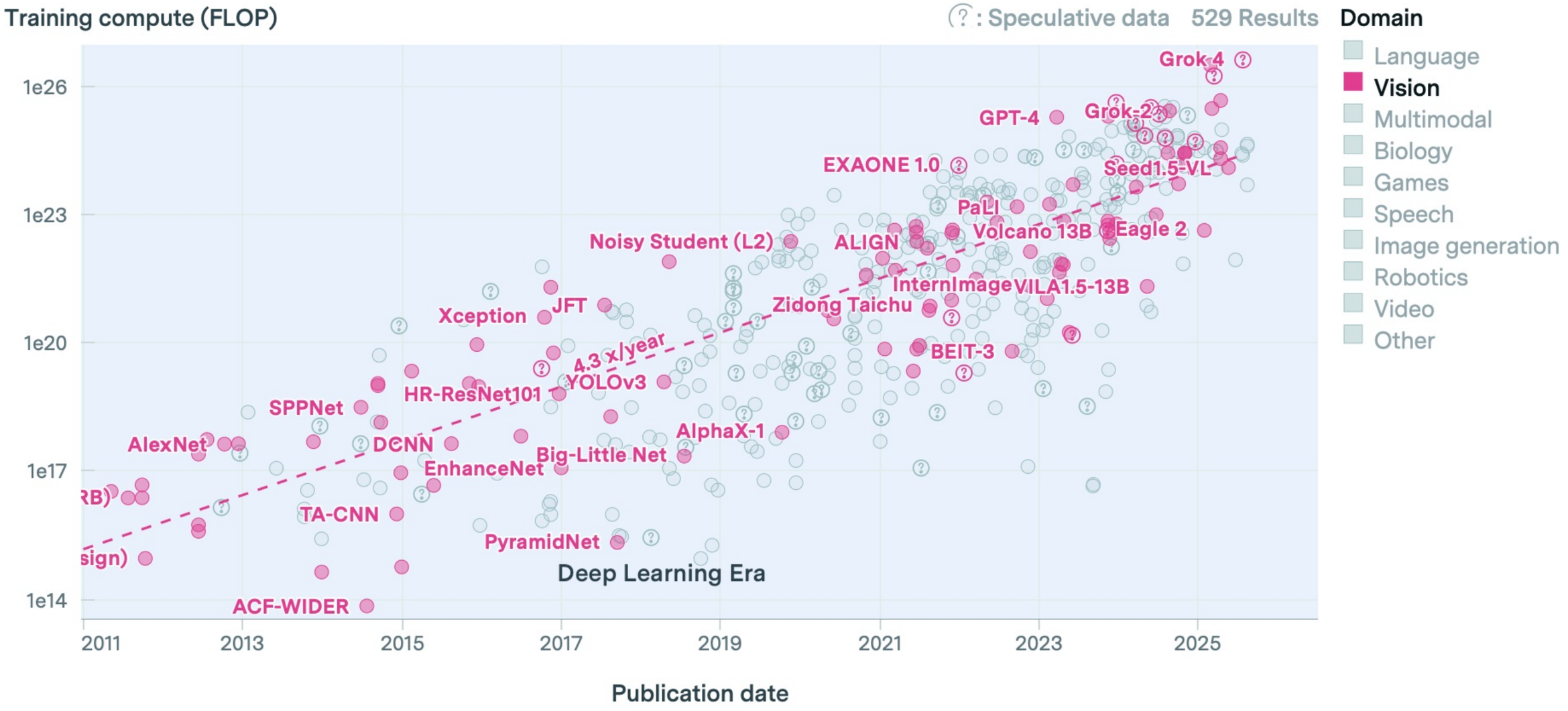
AI Models Over Time

Epoch AI, ‘Data on AI Models’.
Published online at ‘https://epoch.ai/data/ai-models'.



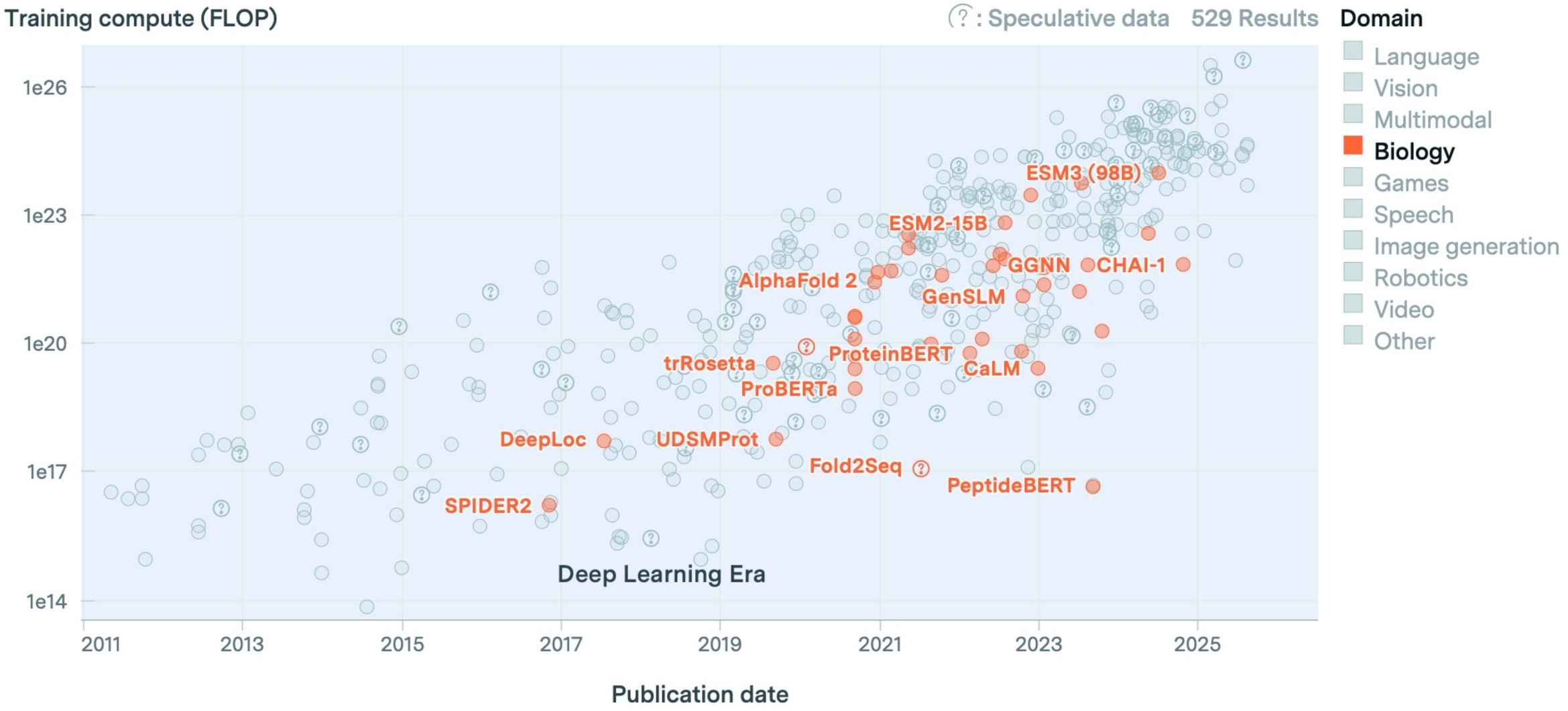
AI Models Over Time

Epoch AI, ‘Data on AI Models’.
Published online at ‘https://epoch.ai/data/ai-models'.



AI Models Over Time

Epoch AI, 'Data on AI Models'.
Published online at 'https://epoch.ai/data/ai-models'.



Model Size Evolution of GPT

GPT-1
117M
(2018)

GPT-2
1.5B
(2019)

GPT-3
175B
(2020)

Model Size Evolution of GPT

GPT-1	GPT-2	GPT-3	GPT-4
117M	1.5B	175B	1.76T
(2018)	(2019)	(2020)	(2023)

Model Size Evolution of GPT

Scaling Laws for Neural Language Models

Jared Kaplan* Johns Hopkins University, OpenAI jaredk@jhu.edu

Sam McCandlish* OpenAI sam@openai.com

Tom Henighan OpenAI henighan@openai.com

Tom B. Brown OpenAI tom@openai.com

Benjamin Chess OpenAI bchess@openai.com

Rewon Child OpenAI rewon@openai.com

Scott Gray OpenAI scott@openai.com

Alec Radford OpenAI alec@openai.com

Jeffrey Wu OpenAI jeffwu@openai.com

Dario Amodei OpenAI damodei@openai.com

Abstract

We study empirical scaling laws for language model performance on the cross-entropy loss. The loss scales as a power-law with model size, dataset size, and the amount of compute used for training, with some trends spanning more than seven orders of magnitude. Other architectural details such as network width or depth have minimal effects within a wide range. Simple equations govern the dependence of overfitting on model/dataset size and the dependence of training speed on model size. These relationships allow us to determine the optimal allocation of a fixed compute budget. Larger models are significantly more sample-efficient, such that optimally compute-efficient training involves training very large models on a relatively modest amount of data and stopping significantly before convergence.

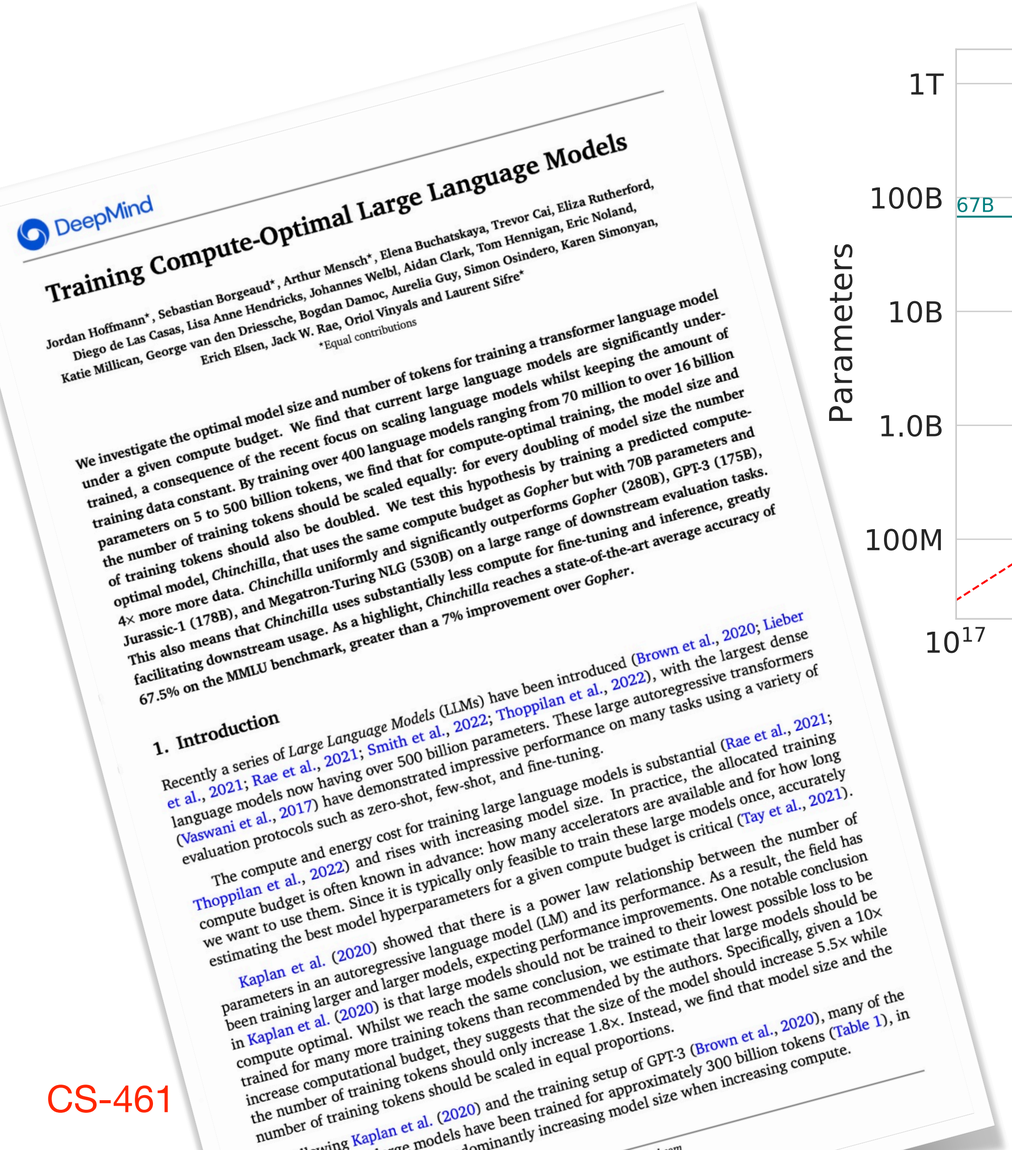
*Equal contribution.

Jared Kaplan and Sam McCandlish led the research. Tom Henighan contributed the LSTM ex-
Brown, Rewon Child, and Scott Gray, and Alec Radford developed the optimized Transformer
Benjamin Chess, and Alec Radford developed the text datasets. Dario Amodei provided

GPT-3
175B
(2020)

GPT-4
1.76T
(2023)

Scaling Model Size *and* Data



DeepMind

Training Compute-Optimal Large Language Models

Jordan Hoffmann*, Sebastian Borgeaud*, Arthur Mensch*, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals and Laurent Sifre*

*Equal contributions

We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are significantly undertrained, a consequence of the recent focus on scaling language models whilst keeping the amount of training data constant. By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, we find that for compute-optimal training, the model size and the number of training tokens should be scaled equally: for every doubling of model size the number of training tokens should also be doubled. We test this hypothesis by training a predicted compute-optimal model, *Chinchilla*, that uses the same compute budget as *Gopher* but with 70B parameters and 4× more more data. *Chinchilla* uniformly and significantly outperforms *Gopher* (280B), GPT-3 (175B), Jurassic-1 (178B), and Megatron-Turing NLG (530B) on a large range of downstream evaluation tasks. This also means that *Chinchilla* uses substantially less compute for fine-tuning and inference, greatly facilitating downstream usage. As a highlight, *Chinchilla* reaches a state-of-the-art average accuracy of 67.5% on the MMLU benchmark, greater than a 7% improvement over *Gopher*.

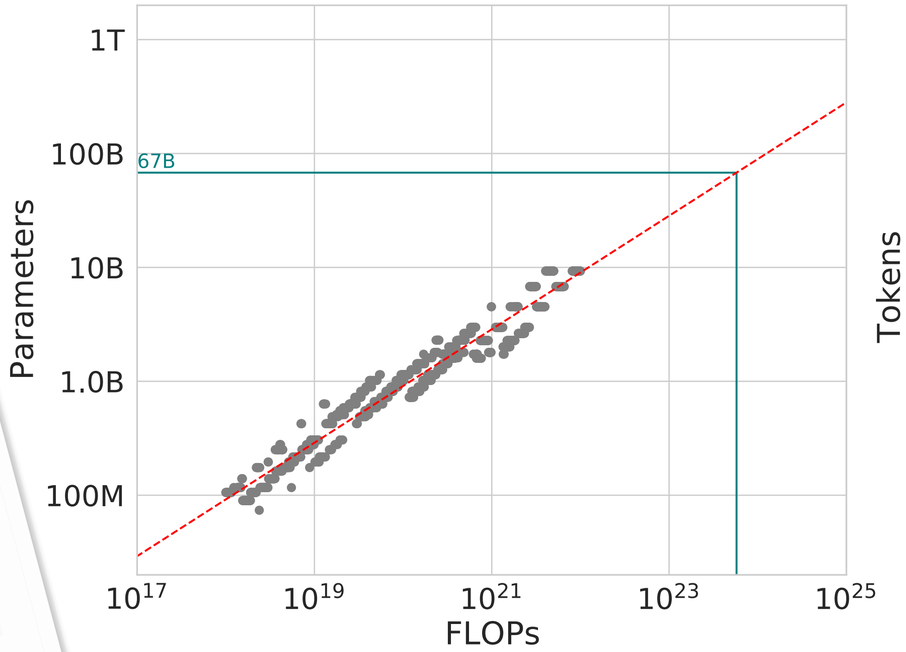
1. Introduction

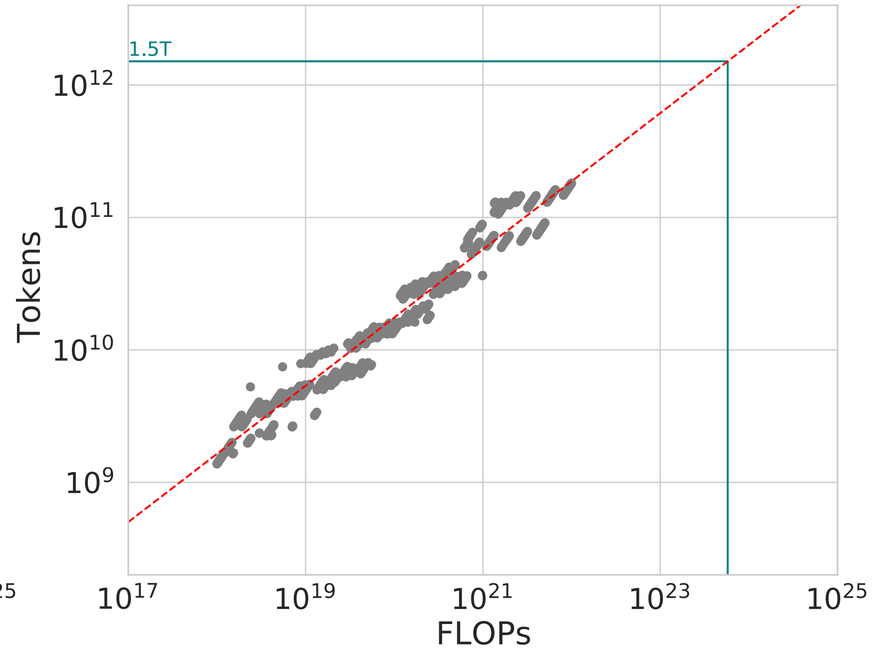
Recently a series of *Large Language Models* (LLMs) have been introduced (Brown et al., 2020; Lieber et al., 2021; Rae et al., 2021; Smith et al., 2022; Thoppilan et al., 2022), with the largest dense language models now having over 500 billion parameters. These large autoregressive transformers (Vaswani et al., 2017) have demonstrated impressive performance on many tasks using a variety of evaluation protocols such as zero-shot, few-shot, and fine-tuning.

The compute and energy cost for training large language models is substantial (Rae et al., 2021; Thoppilan et al., 2022) and rises with increasing model size. In practice, the allocated training compute budget is often known in advance: how many accelerators are available and for how long we want to use them. Since it is typically only feasible to train these large models once, accurately estimating the best model hyperparameters for a given compute budget is critical (Tay et al., 2021).

Kaplan et al. (2020) showed that there is a power law relationship between the number of parameters in an autoregressive language model (LM) and its performance. As a result, the field has been training larger and larger models, expecting performance improvements. One notable conclusion in Kaplan et al. (2020) is that large models should not be trained to their lowest possible loss to be compute optimal. Whilst we reach the same conclusion, we estimate that large models should be trained for many more training tokens than recommended by the authors. Specifically, given a 10× increase computational budget, they suggests that the size of the model should increase 5.5× while the number of training tokens should only increase 1.8×. Instead, we find that model size and the number of training tokens should be scaled in equal proportions.

Following Kaplan et al. (2020) and the training setup of GPT-3 (Brown et al., 2020), many of the large models have been trained for approximately 300 billion tokens (Table 1), in dominantly increasing model size when increasing compute.





An LLM with 70B parameters (half the size of GPT-3) but 4x more data outperforms GPT-3.

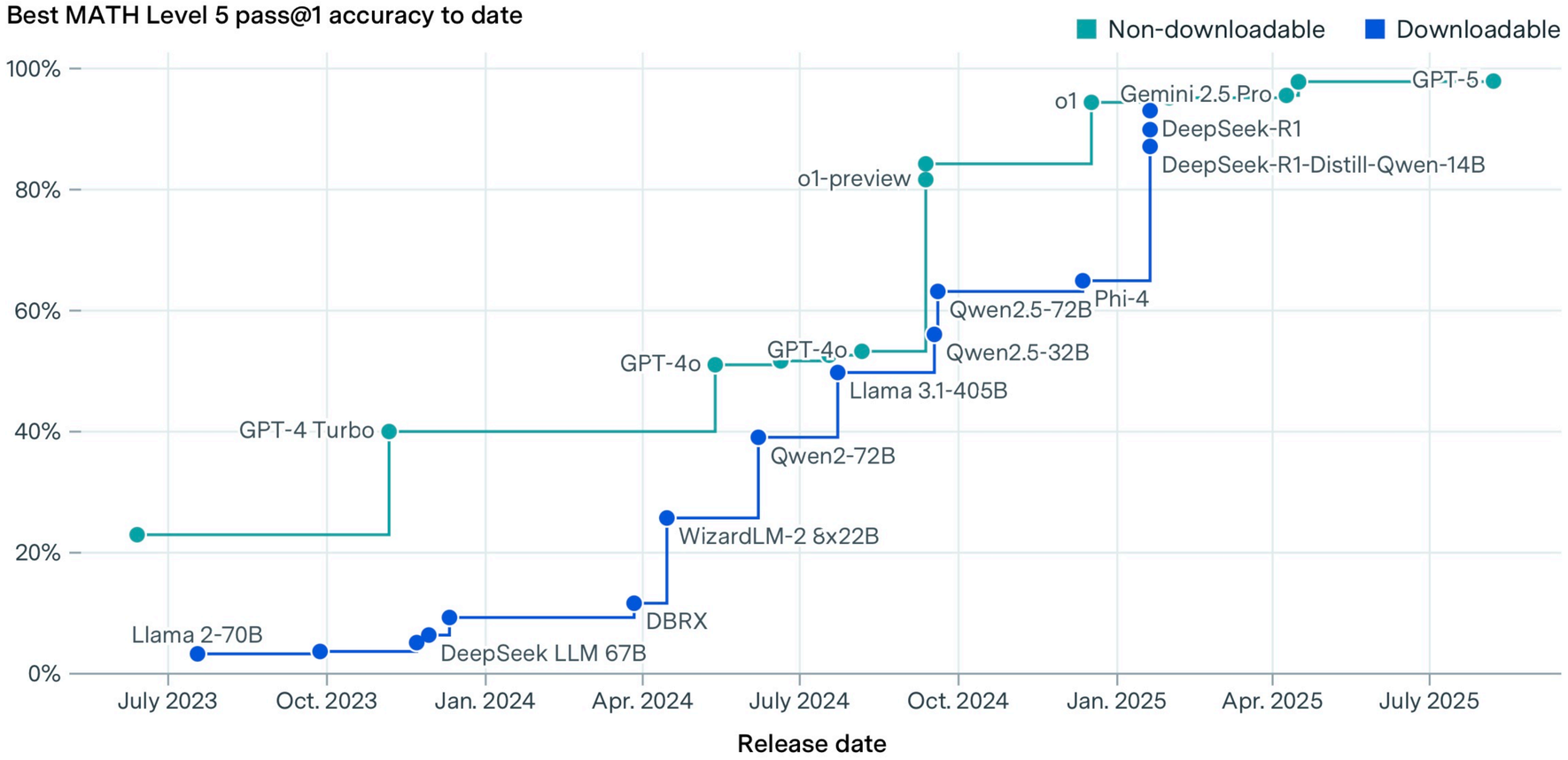
Lecture 2: Learning at Scale

Is it simply scaling? No.

Epoch AI, 'AI Benchmarking Hub'.
Published online at 'https://epoch.ai/benchmarks#data-insights'.

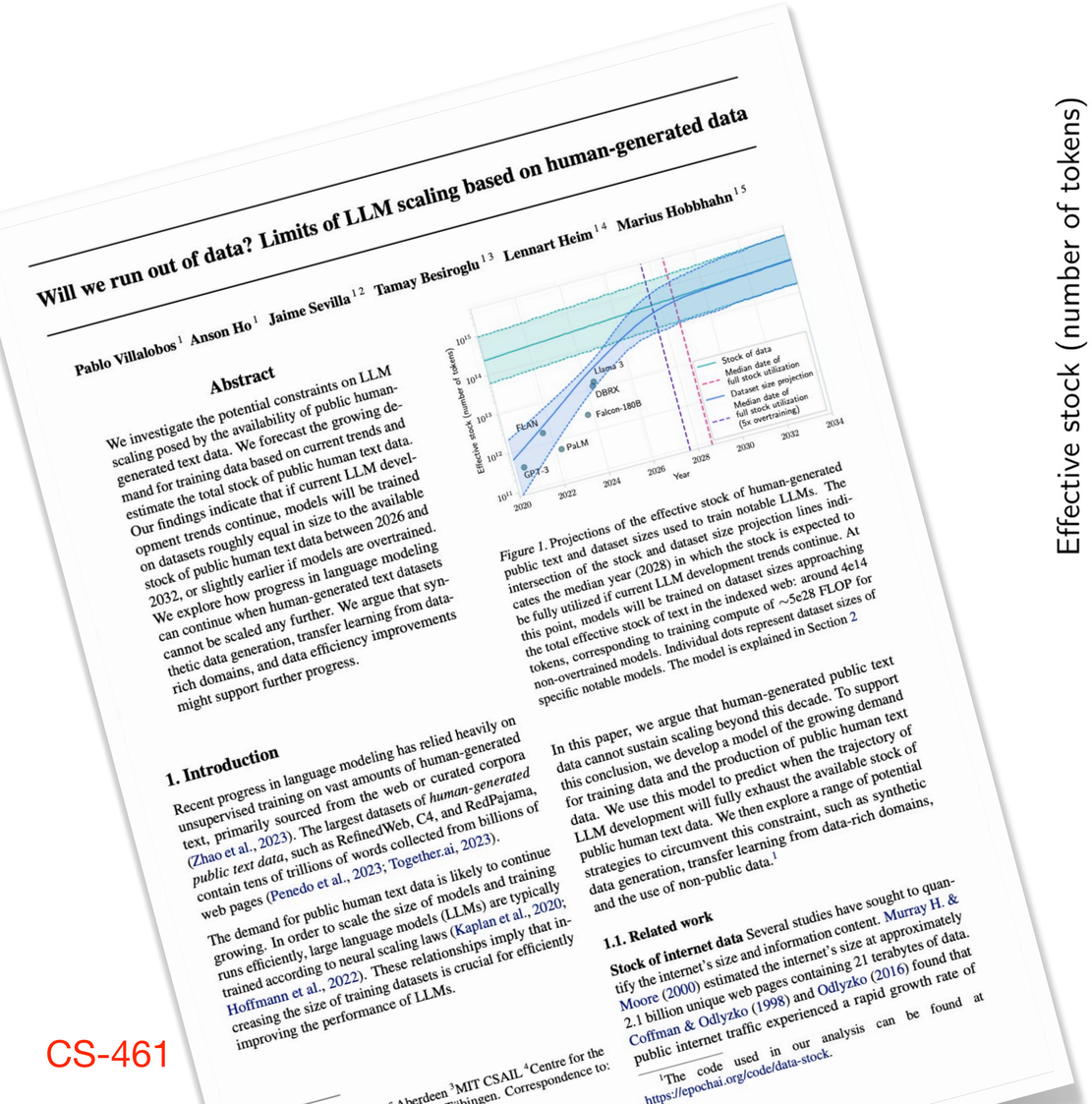
Have we reached the limits of scaling laws?

As the latest generation of models have gotten bigger and more expensive, capabilities have started to plateau.



Will we run out of data?

Access to high-quality data, i.e., public human-generated text, has become a major bottleneck.



Will we run out of data? Limits of LLM scaling based on human-generated data

Pablo Villalobos [1] Anson Ho [1] Jaime Sevilla [1 2] Tamay Besiroglu [1 3] Lennart Heim [1 4] Marius Hobbhahn [1 5]

Abstract

We investigate the potential constraints on LLM scaling posed by the availability of public human-generated text data. We forecast the growing demand for training data based on current trends and estimate the total stock of public human text data. Our findings indicate that if current LLM development trends continue, models will be trained on datasets roughly equal in size to the available stock of public human text data between 2026 and 2032, or slightly earlier if models are overtrained. We explore how progress in language modeling can continue when human-generated text datasets cannot be scaled any further. We argue that synthetic data generation, transfer learning from data-rich domains, and data efficiency improvements might support further progress.

Figure 1. Projections of the effective stock of human-generated public text and dataset sizes used to train notable LLMs. The intersection of the stock and dataset size projection lines indicates the median year (2028) in which the stock is expected to be fully utilized if current LLM development trends continue. At this point, models will be trained on dataset sizes approaching the total effective stock of text in the indexed web: around 4e14 tokens, corresponding to training compute of ~5e28 FLOP for non-overtrained models. Individual dots represent dataset sizes of specific notable models. The model is explained in Section 2

1. Introduction

Recent progress in language modeling has relied heavily on unsupervised training on vast amounts of human-generated text, primarily sourced from the web or curated corpora (Zhao et al., 2023). The largest datasets of *human-generated public text data*, such as RefinedWeb, C4, and RedPajama, contain tens of trillions of words collected from billions of web pages (Penedo et al., 2023; Together.ai, 2023).

The demand for public human text data is likely to continue growing. In order to scale the size of models and training runs efficiently, large language models (LLMs) are typically trained according to neural scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). These relationships imply that increasing the size of training datasets is crucial for efficiently improving the performance of LLMs.

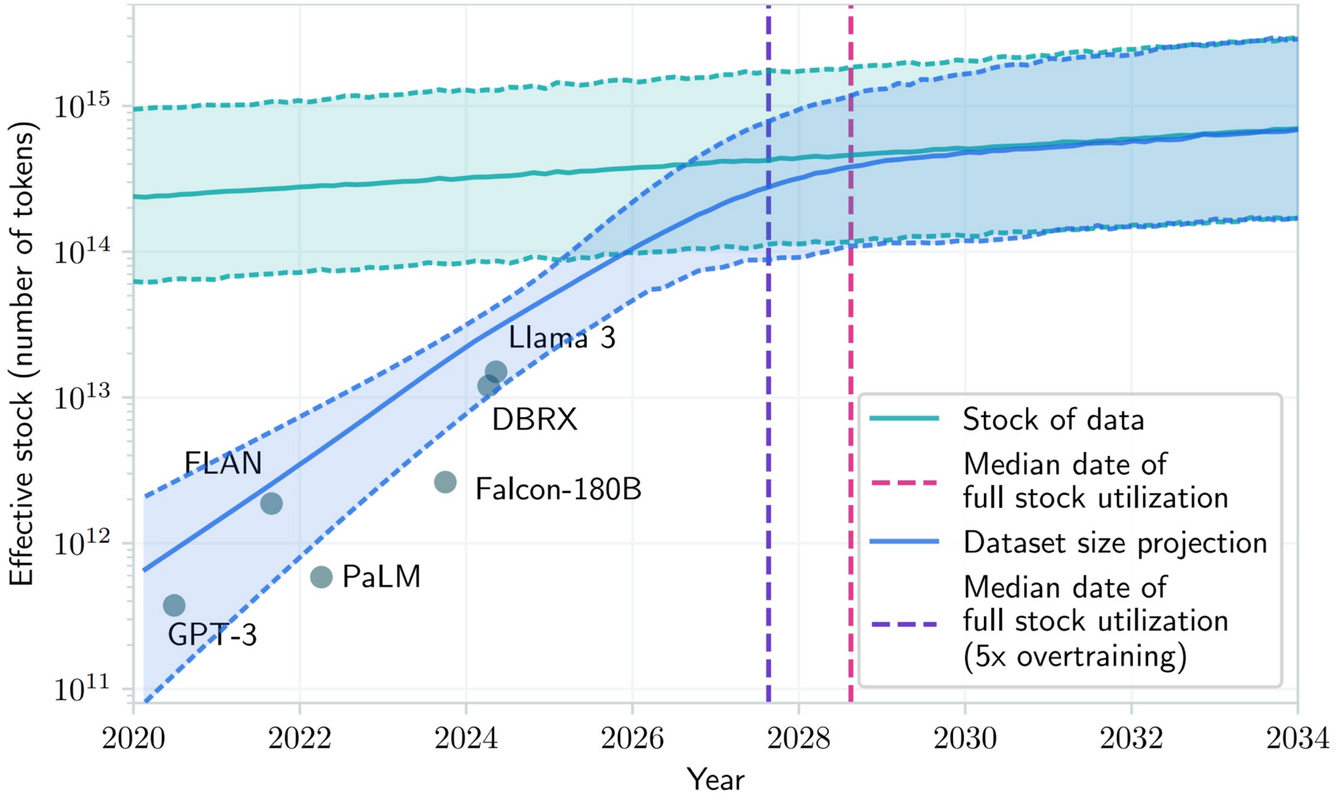
In this paper, we argue that human-generated public text data cannot sustain scaling beyond this decade. To support this conclusion, we develop a model of the growing demand for training data and the production of public human text data. We use this model to predict when the trajectory of LLM development will fully exhaust the available stock of public human text data. We then explore a range of potential strategies to circumvent this constraint, such as synthetic data generation, transfer learning from data-rich domains, and the use of non-public data.[1]

1.1. Related work

Stock of internet data Several studies have sought to quantify the internet's size and information content. Murray H. & Moore (2000) estimated the internet's size at approximately 2.1 billion unique web pages containing 21 terabytes of data. Coffman & Odlyzko (1998) and Odlyzko (2016) found that public internet traffic experienced a rapid growth rate of

[1]The code used in our analysis can be found at https://epochai.org/code/data-stock.

[1]Epoch AI, San Francisco, California, USA [2]University of Aberdeen [3]MIT CSAIL [4]Centre for the Future Intelligence, University of Tübingen. Correspondence to:



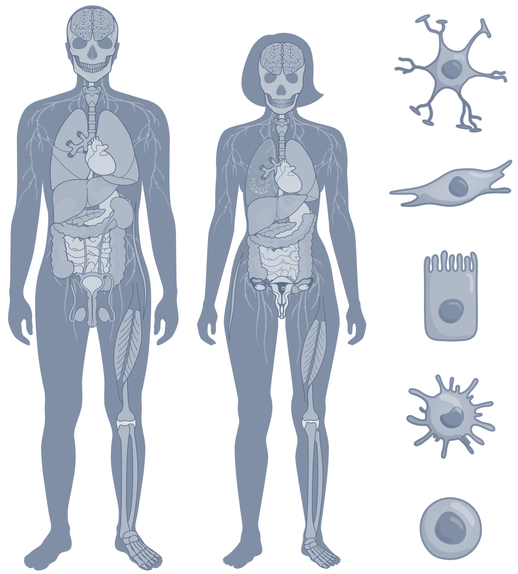
… what comes next?

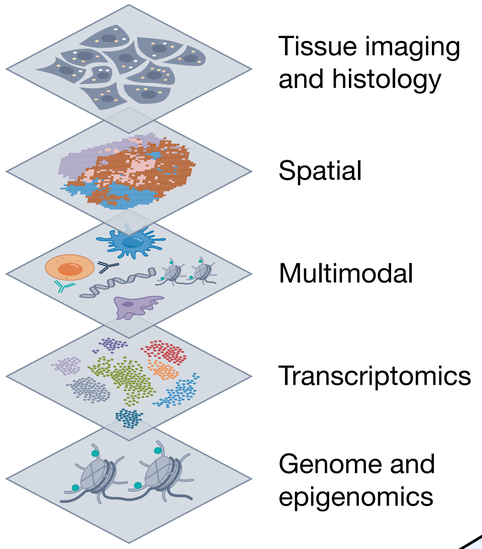
Lecture 13: Reasoning

Lecture 14: FM and Agents

Opposite Trends: Data in Biology

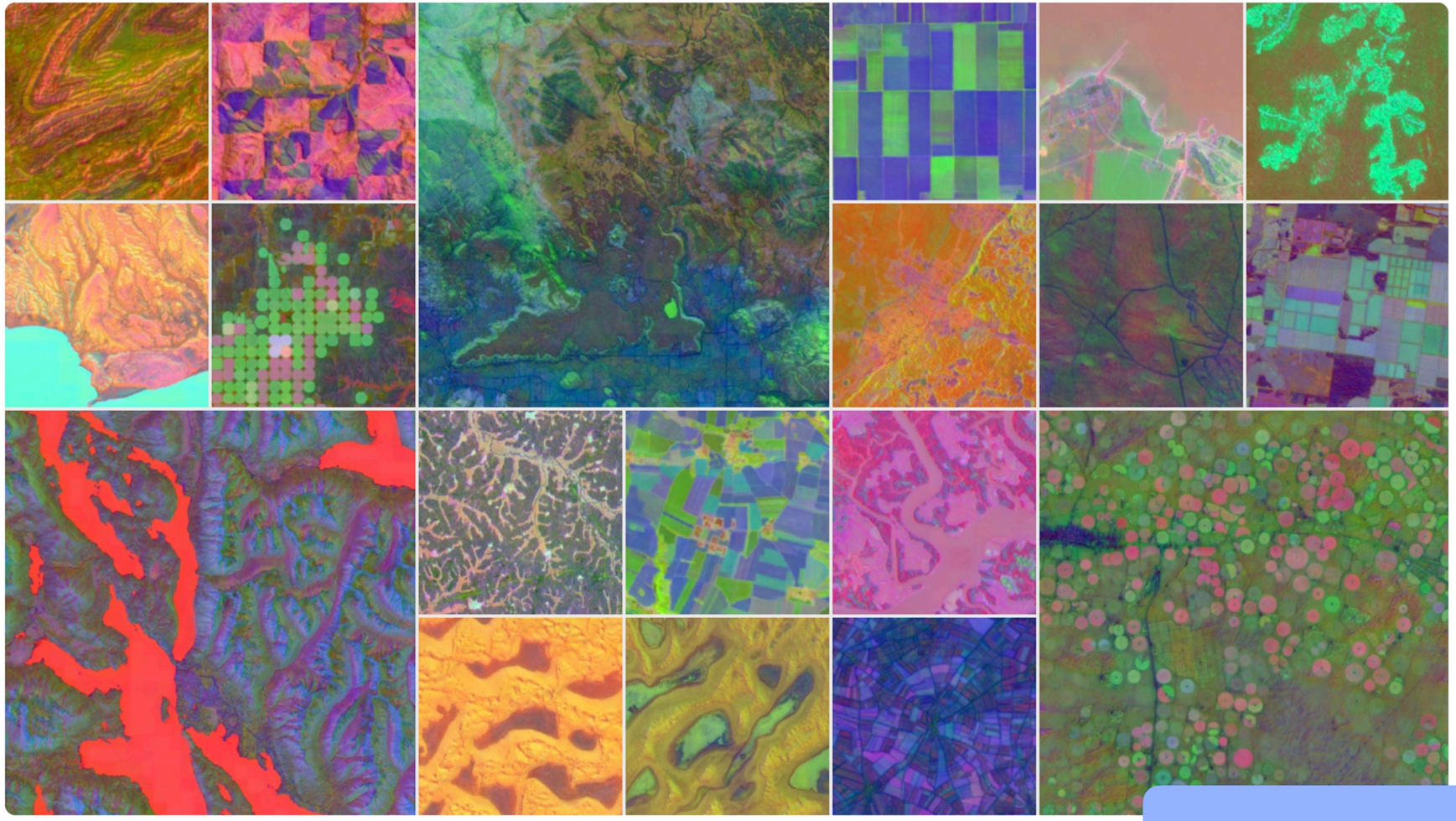
Example: Single-cell biology





Opposite Trends: Data in Earth Sciences

Example: We have and are collecting petabytes of earth observation data.



Lecture 9: FM in Earth Sciences

Ingredients of Foundation Models

1. Data

2. Compute

3. Model

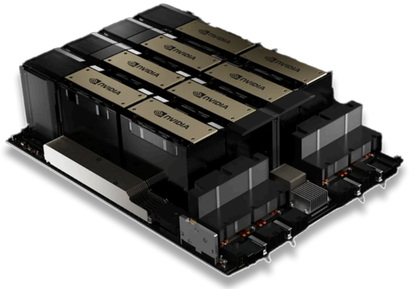
Lecture 5-9: Architectures

4. Objective

Lecture 2: Learning at Scale

Compute

NVIDIA H100s



Compute is often the **bottleneck** that determines:

- What models you can train
  → size limits
- How fast you can iterate
  → research speed
- What problems you can solve
  → capability boundaries
- **Who can participate** in AI development
  → democratization vs. concentration

Historical perspective:

- **1990s-2000s**: AI models trained on single CPUs
- **2010s**: GPUs enabled deep learning breakthroughs
- **2020s**: Massive clusters training trillion-parameter models

Modern reality: Training GPT-4 likely required **~25,000 GPUs** running for **months**, costing **tens of millions of dollars**.

Compute

Generational evolution of NVIDIA **A100** vs. **H100** vs. **H200**

A100

2020 • Ampere

Memory	**80GB**
Bandwidth	**1.9 TB/s**
FP16 Perf	**312 TFLOPs**
Transformer Engine	**No**

Best For:
Cost-effective inference, smaller models (<70B), research on budget

H100

2022 • Hopper

Memory	**80GB**
Bandwidth	**3.35 TB/s**
FP16 Perf	**1,979 TFLOPs**
Transformer Engine	**Yes**

Best For:
Large-scale training (100B+ params), transformers, production workloads

H200

2024 • Hopper+

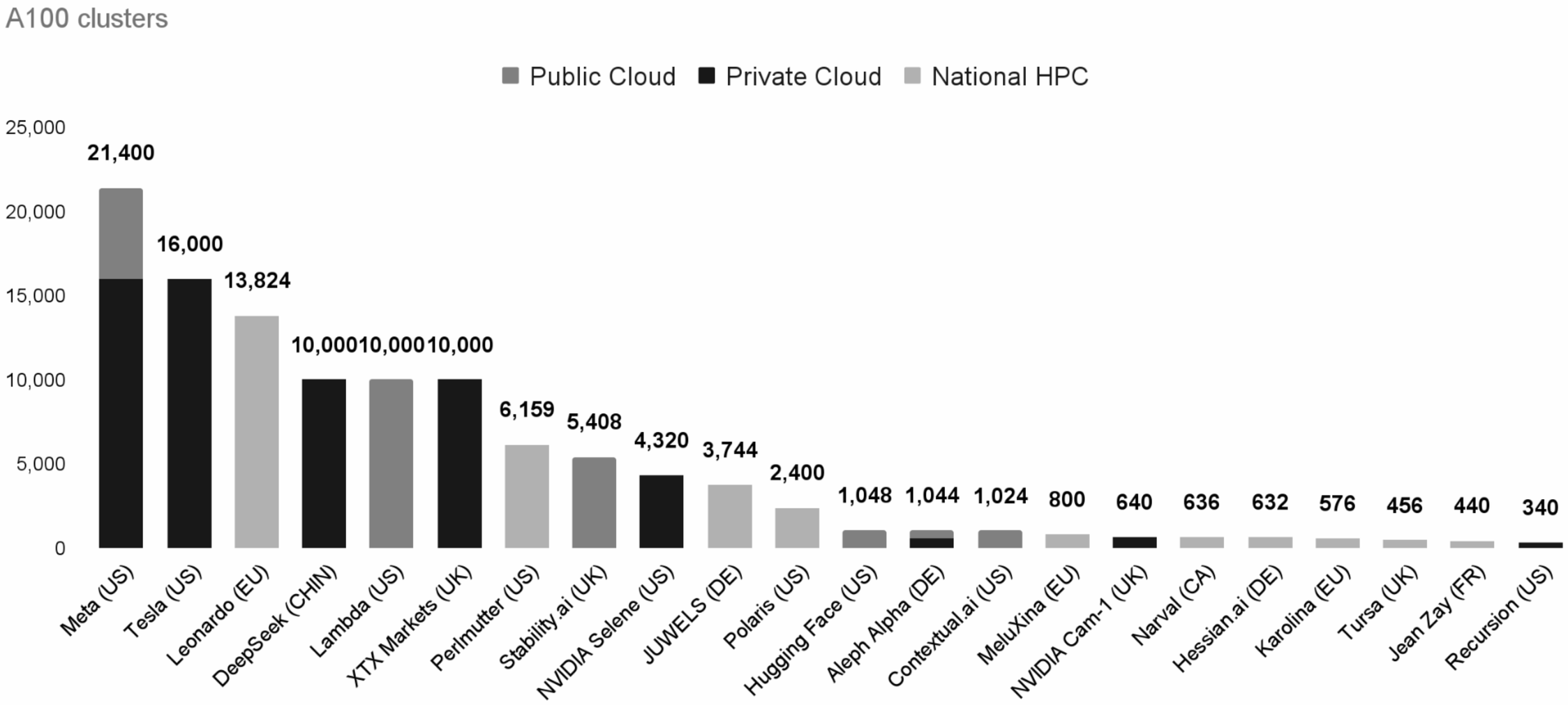
Memory	**141GB**
Bandwidth	**4.8 TB/s**
FP16 Perf	**1,979 TFLOPs**
Transformer Engine	**Yes+**

Best For:
Massive models (120B+ params), long-context, memory-intensive inference

Performance Metric	A100 (Baseline)	H100	H200
Training Speed	1x	2-6x faster	Similar to H100
Memory Capacity	80GB (~70B params)	80GB (~70B params)	141GB (~120B+ params)
Memory Bandwidth	1.9 TB/s	3.35 TB/s (+76%)	4.8 TB/s (+43% vs H100)
Cost Efficiency	Most cost-effective	Best perf/$ for training	Premium pricing

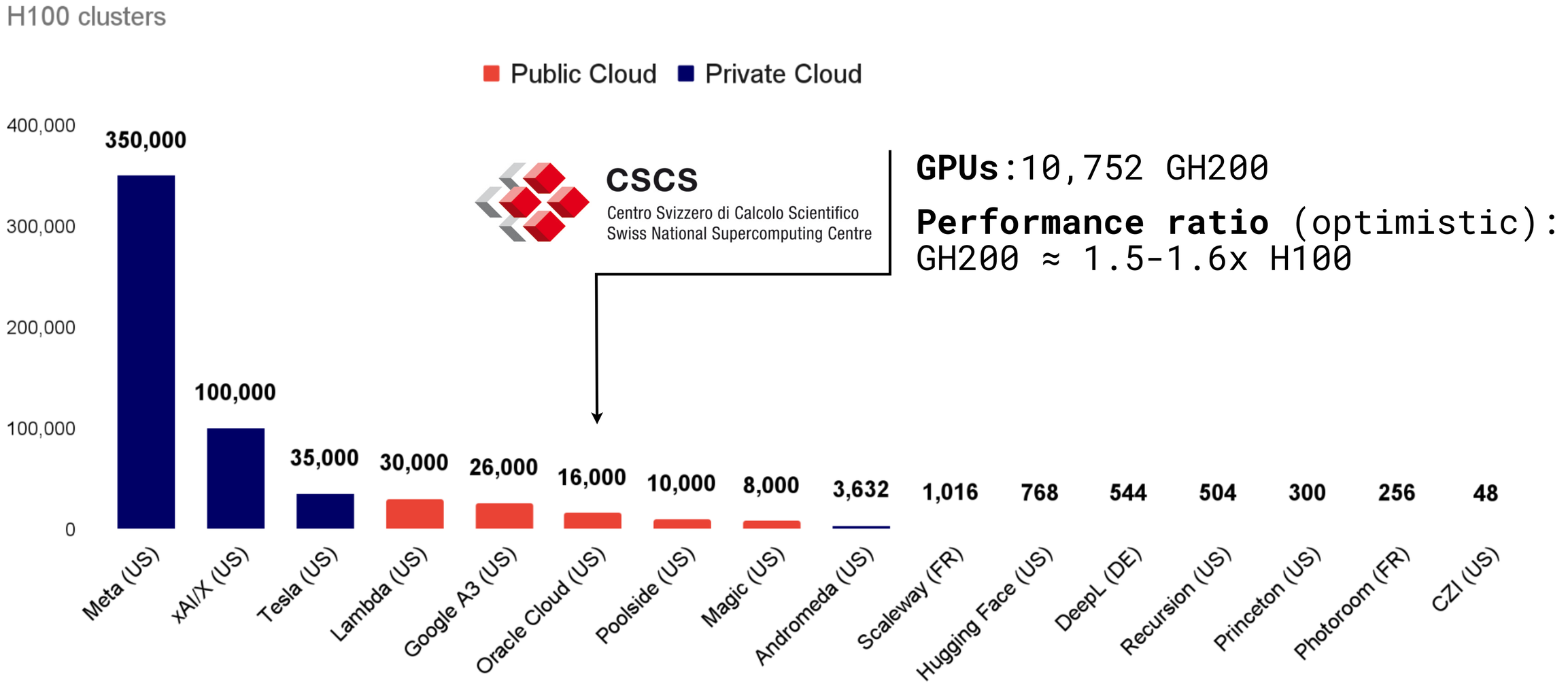
Compute Index: NVIDIA A100 Clusters

State of AI, 'Report 2024'.
Published online at 'https://www.stateof.ai/2024'.

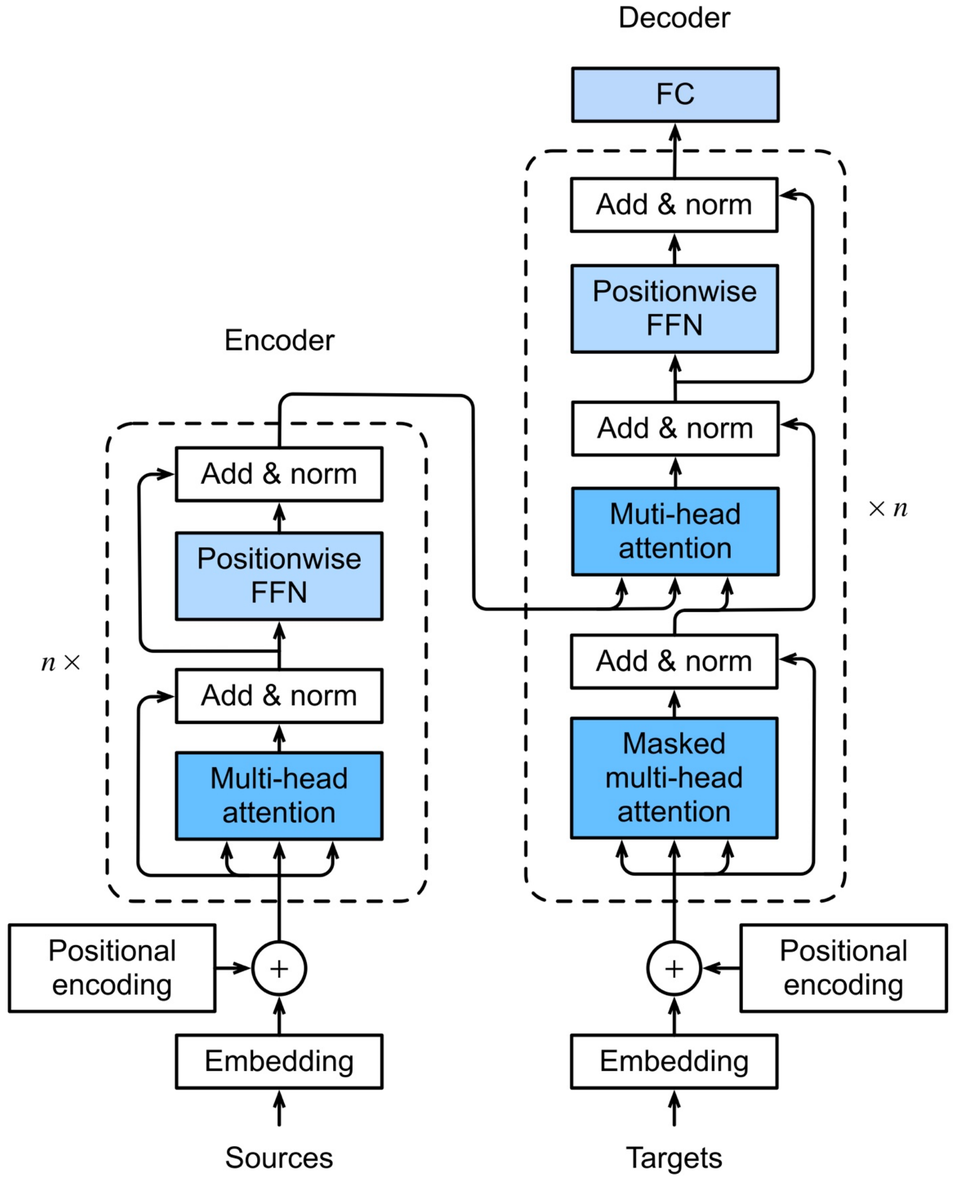


Compute Index: NVIDIA H100 Clusters

State of AI, 'Report 2024'.
Published online at 'https://www.stateof.ai/2024'.



Model: Universality of the Transformer Architecture



Initially designed for neural machine translation, it has become the *de facto* **general-purpose neural network architecture.**

A Transformer produces **contextualized representations**, i.e., numerical vectors that encode each input token along with relevant context from the input data.

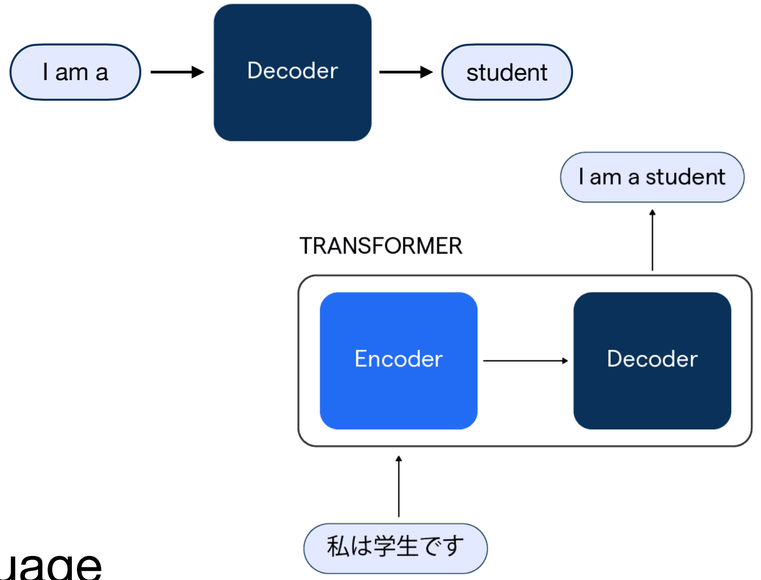
Core Innovations

1. Positional Encodings
2. Attention
3. Self-Attention

CS-433 Machine Learning

CS-552 Modern NLP

Model: Universality of the Transformer Architecture



By Architecture:

- **Encoder-only** (e.g., BERT for understanding and classification)
- **Decoder-only** (e.g., GPT for autoregressive generation)
- **Encoder-decoder** (e.g., T5 for sequence-to-sequence tasks)

By Domain:

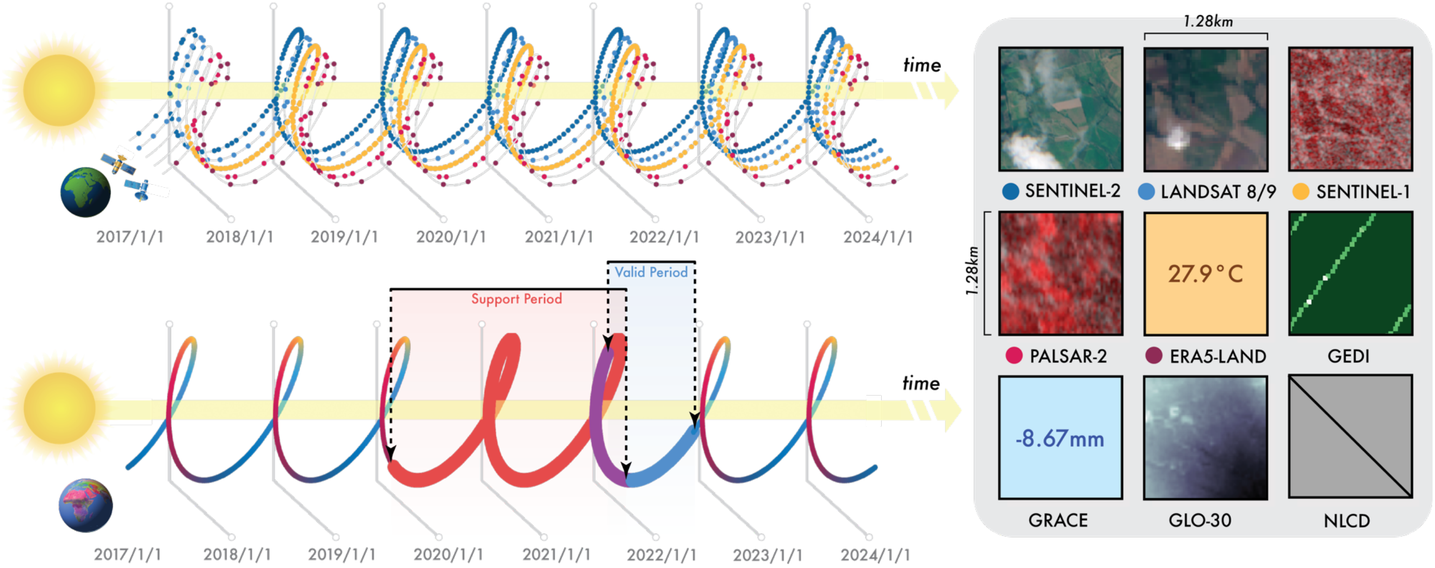
- **Vision Transformers (ViT)**: treats image patches as tokens
- **Multimodal Transformers** (e.g., CLIP): connects vision and language
- **Scientific Transformers** (e.g., Evoformer of AlphaFold for protein/molecular modeling)

By Efficiency:

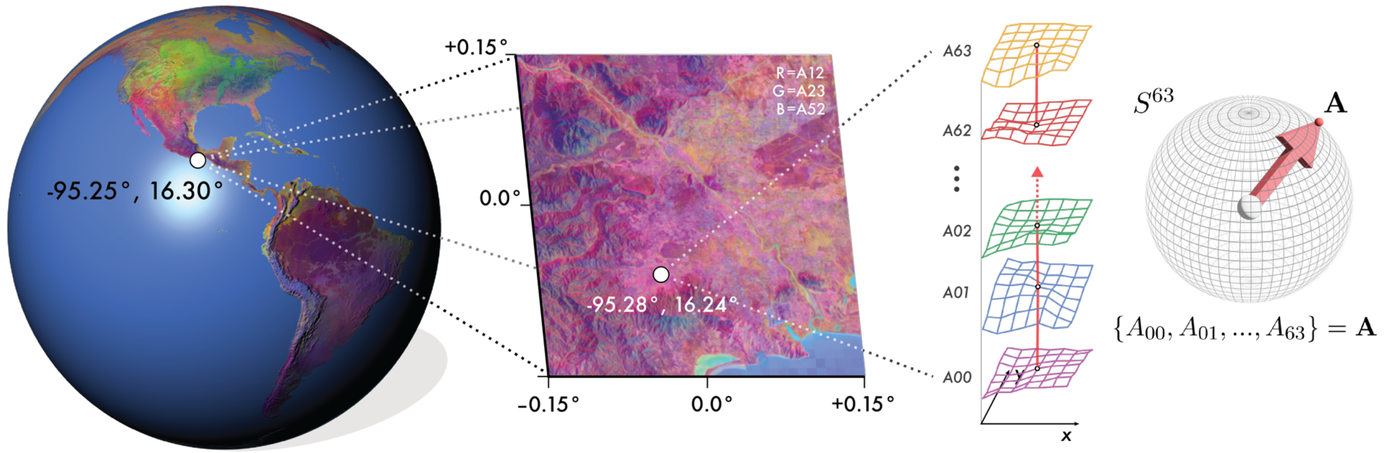
- **Sparse attention variants** (e.g., Longformer for very long sequences)
- **Mixture-of-experts** (e.g., Switch Transformer for massive scaling)

Lecture 5-7: Tokenization and Architectures

Foundation Models in Earth Sciences



FM reconciles multiple sparse, non-uniformly sampled observation records into a continuous record, regardless of fluctuations in availability.

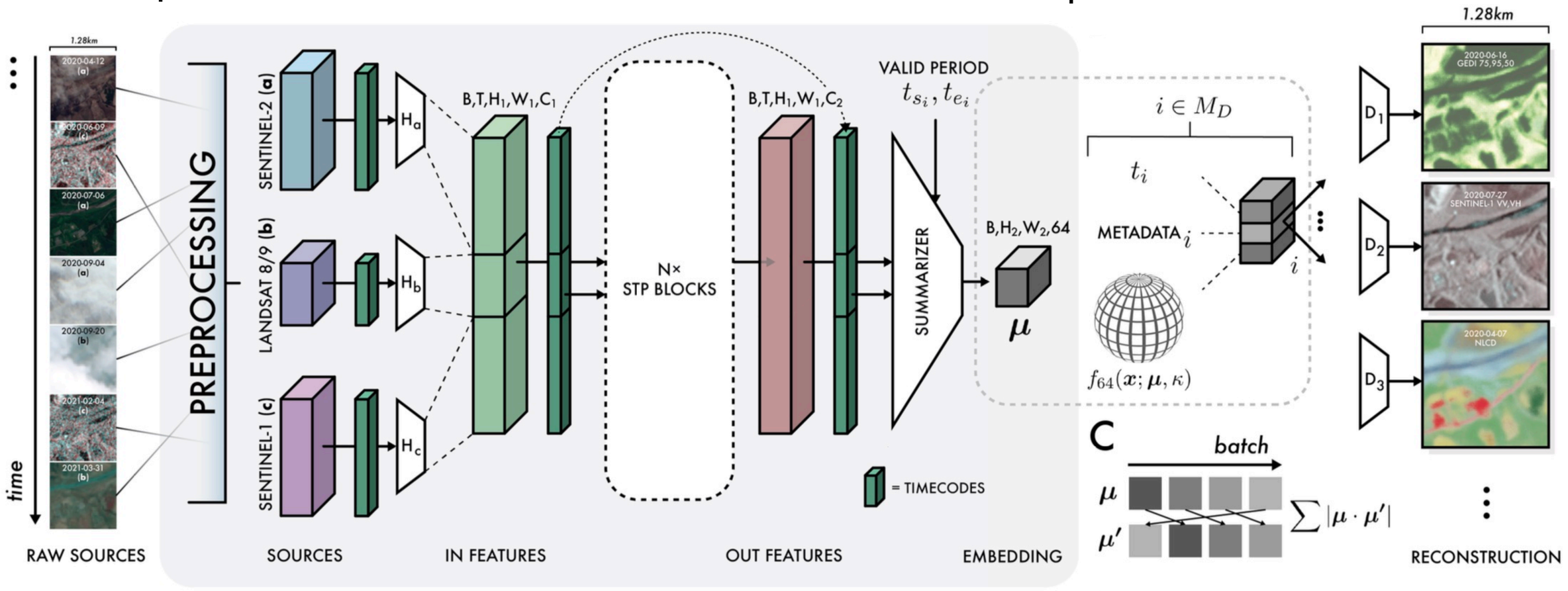


Results in a highly general, **geospatial representation** that assimilates spatial, temporal, and measurement contexts across multiple sources.

Lecture 9: FM in Earth Sciences

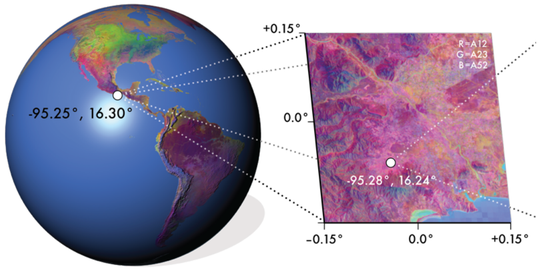
Foundation Models in Earth Sciences

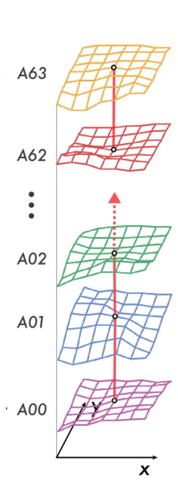
Underlying architecture based on Vision Transformer!

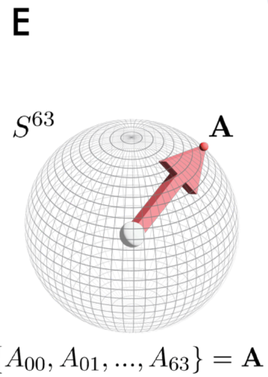


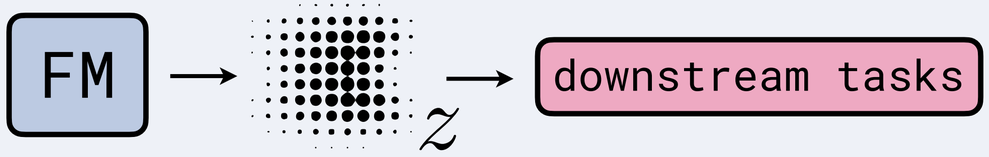
Foundation Models in Earth Sciences

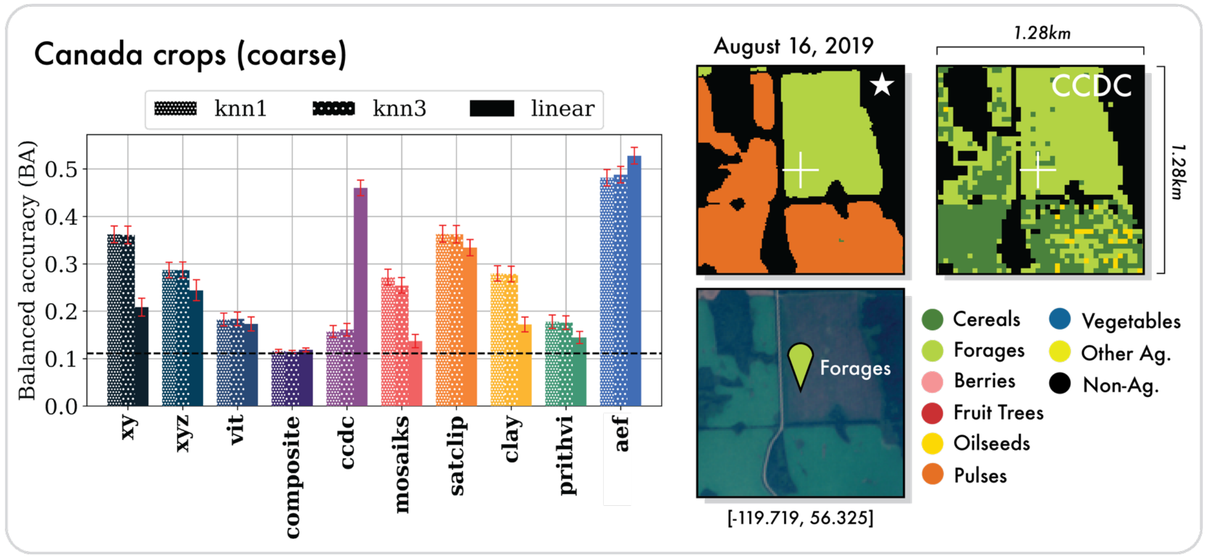
Strong performance
across downstream tasks

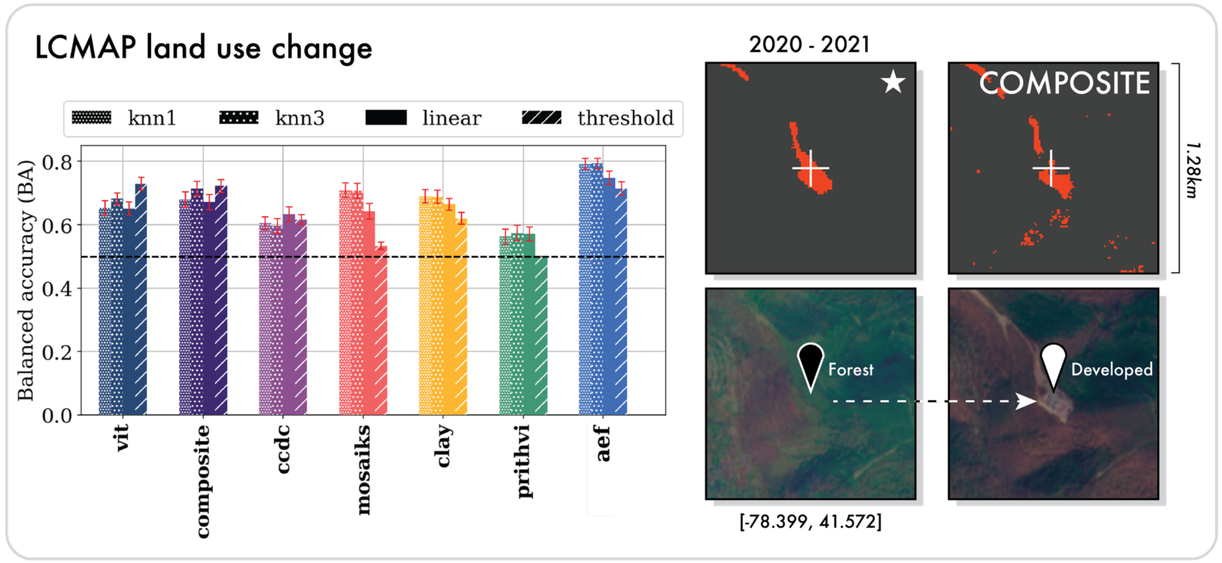






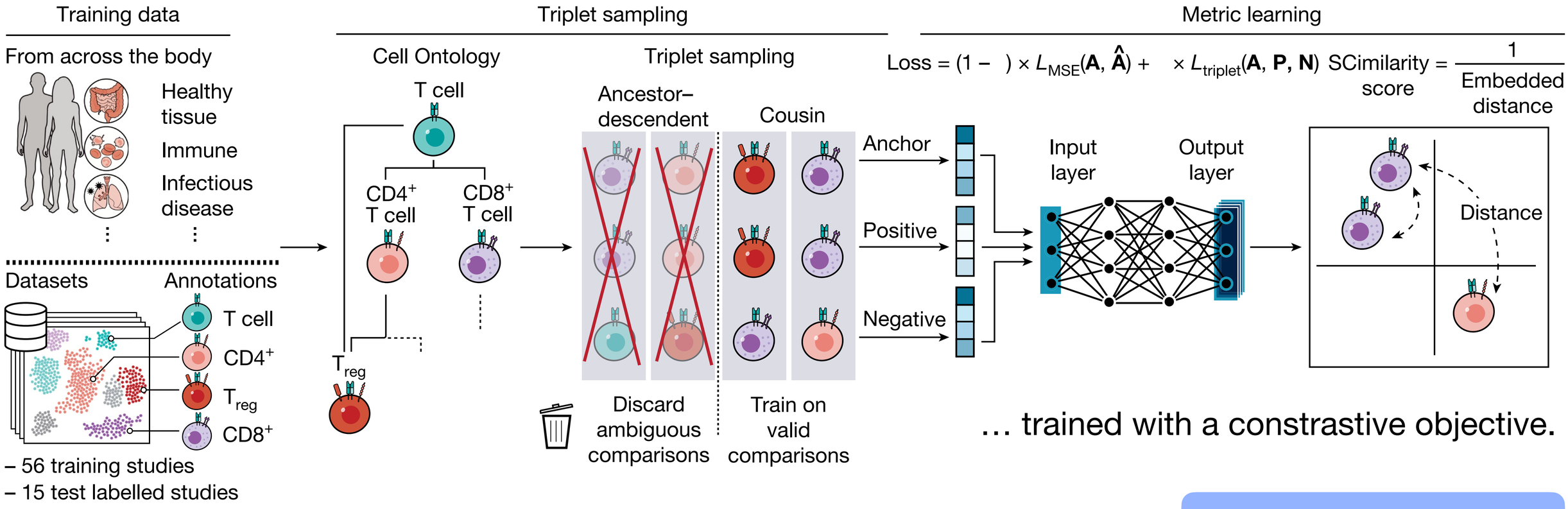






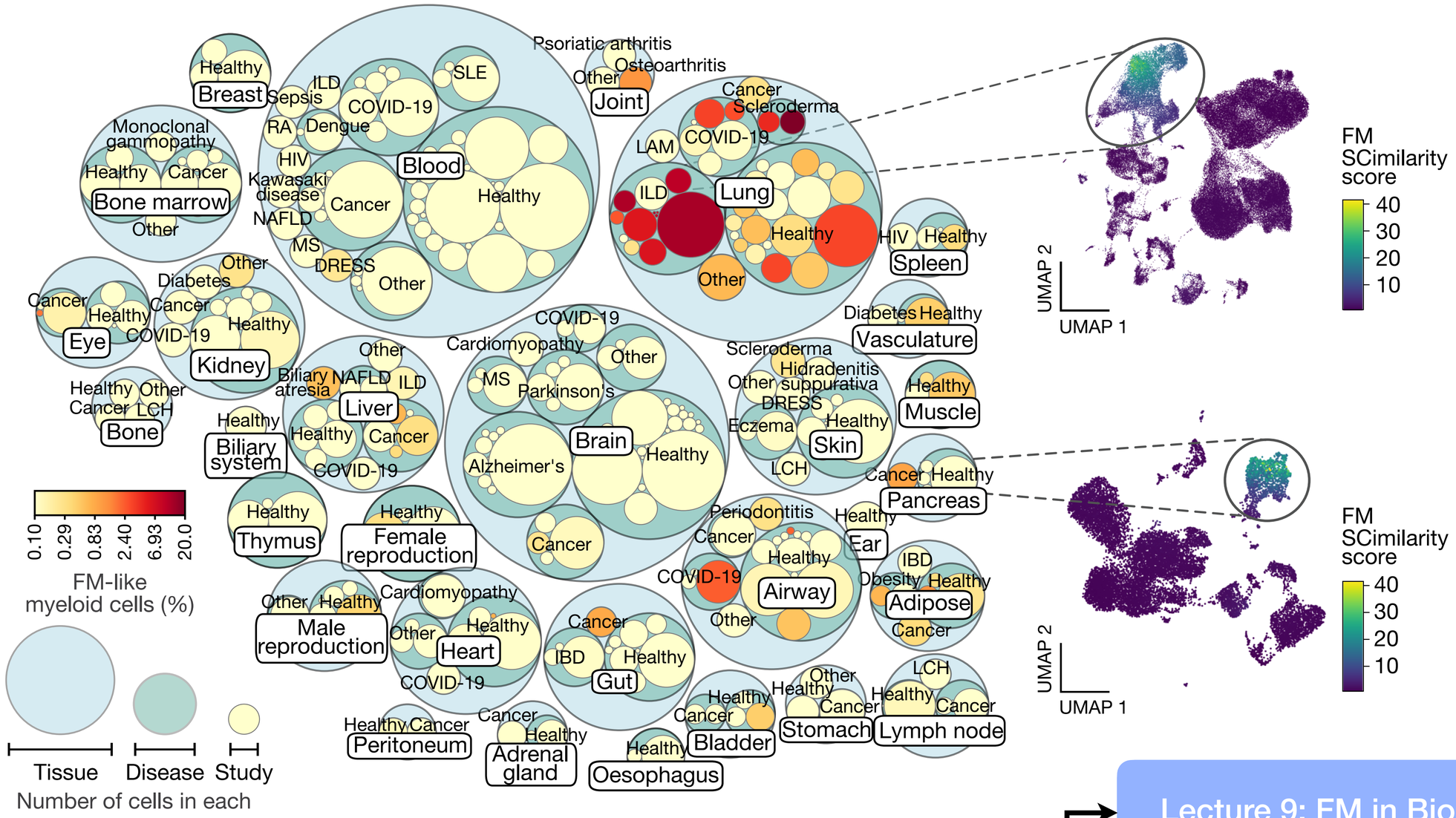
Foundation Models in Biology

A foundation model for single-cell profiles trained on 7,886,247 single-cell profiles from 56 studies that enables researchers to query for similar cellular states across the human body.



Lecture 9: FM in Biology

Foundation Models in Biology



Lecture 9: FM in Biology

Why Generative AI?

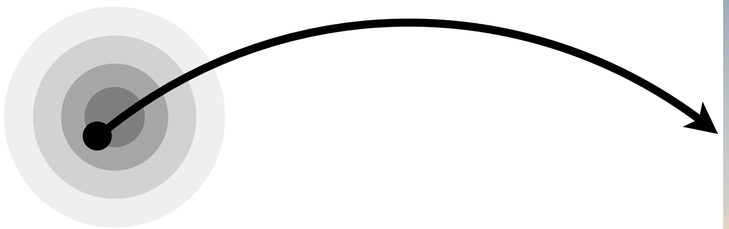
What is Generative AI?

Generative AI is a broad term that can be used for any AI system whose primary function is to generate content.

This is in contrast to AI systems that perform other functions, such as classifying data, grouping data, or choosing actions.

Lecture 3-5: Generative Models

Generative AI: Image Generation



$\mathcal{N}(0, I)$

+ condition

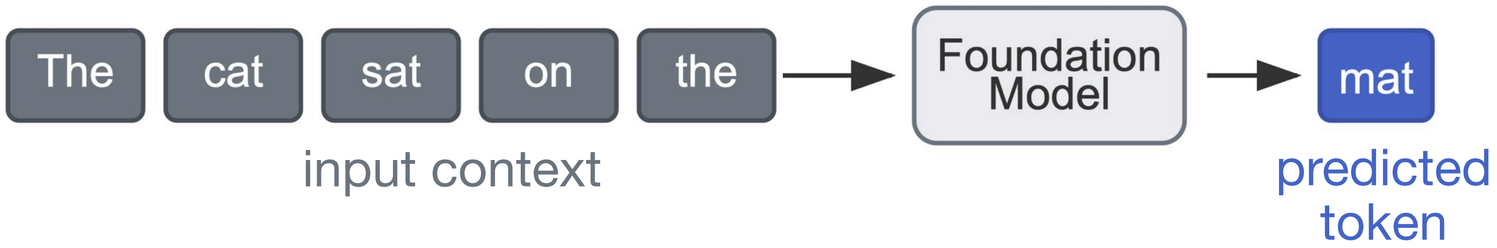
Generative AI and Foundation Models

Learning through Generation

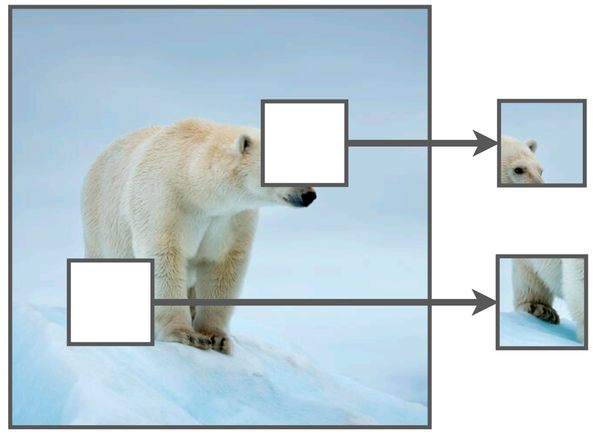
- Generative objectives provide rich training signals
- Forces models to understand deep patterns, not just surface features

Examples:

GPT : autoregressive next-token prediction



MAE : masking

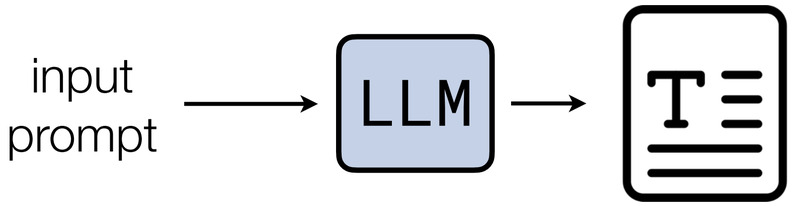


Lecture 2: Learning at Scale

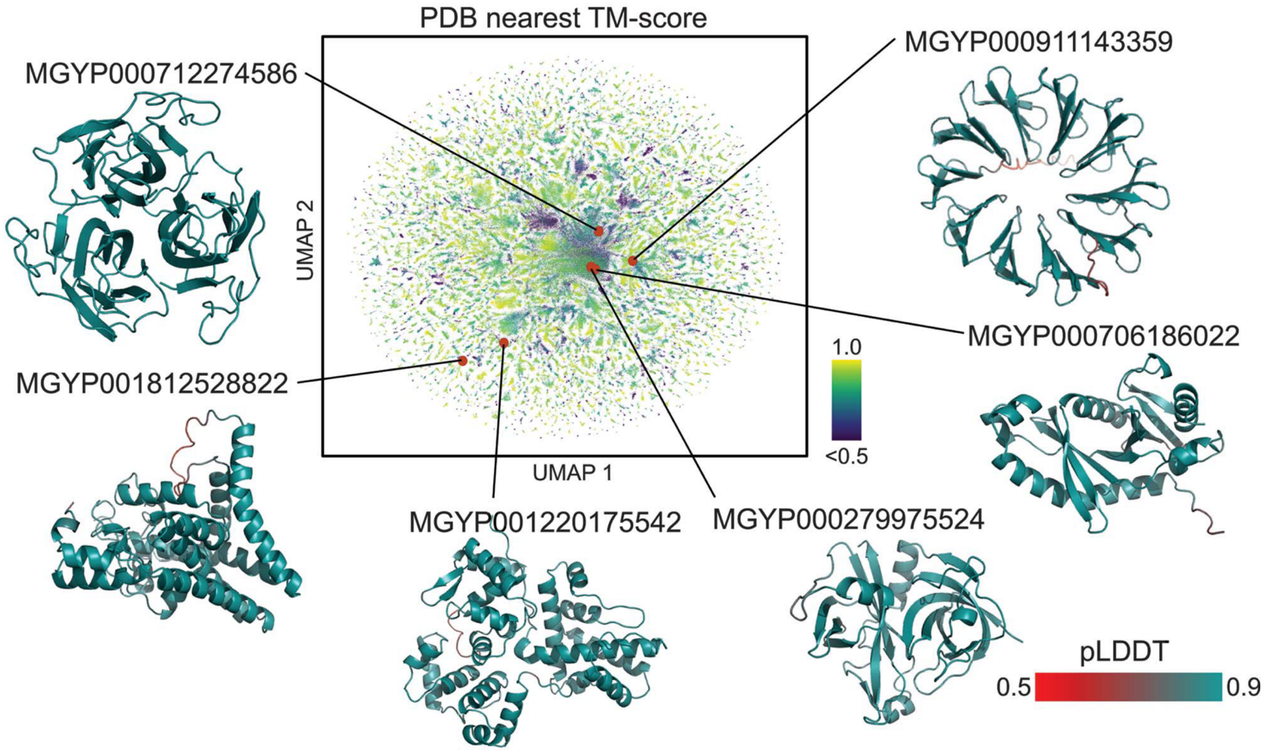
Generative AI and Foundation Models

Foundation Models as Generative Systems

Large language models are generative models



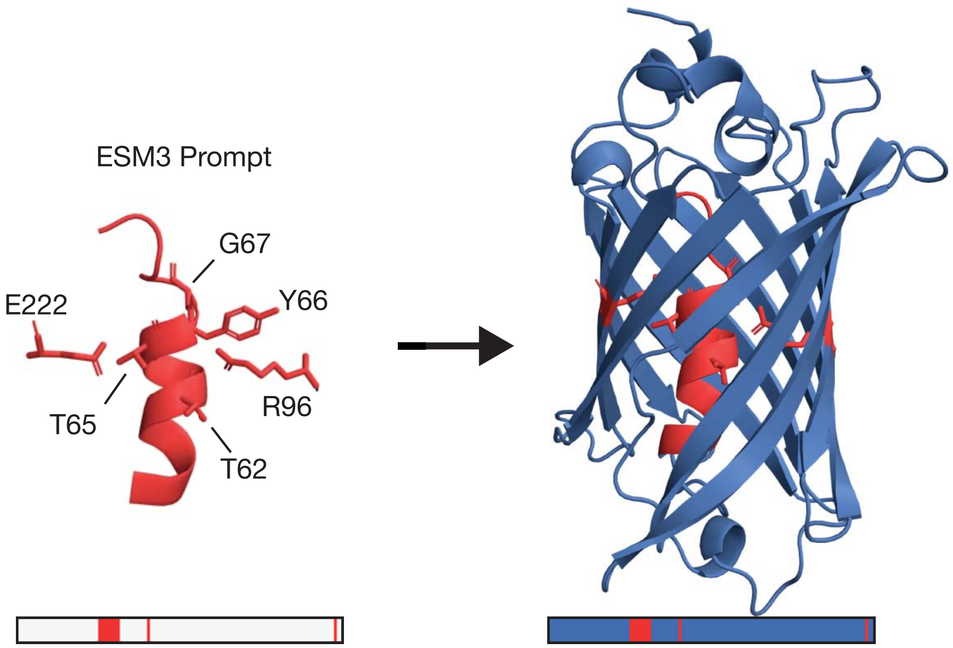
Protein sequence foundation models allow generating *new* proteins

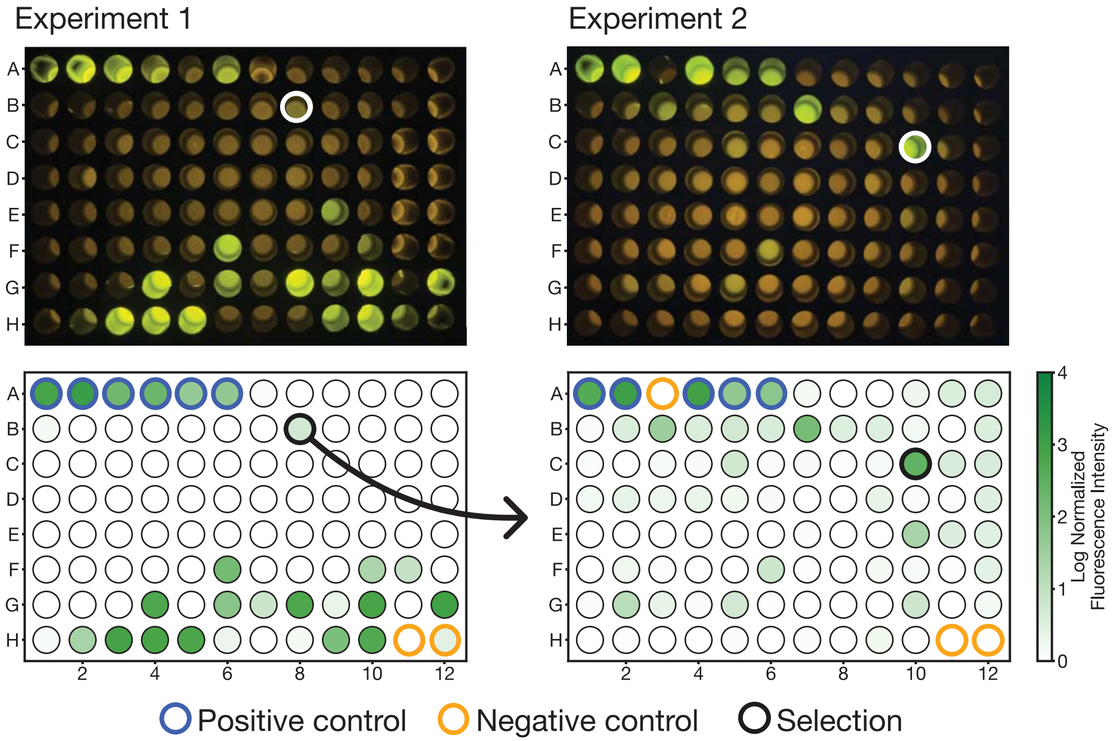


Generative AI and Foundation Models

Foundation Models as Generative Systems

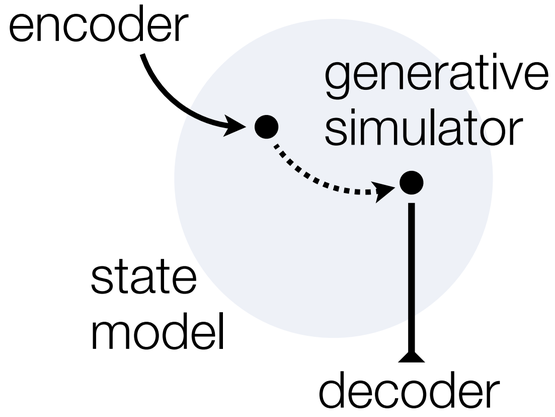
Protein sequence foundation models
allow generating *new* proteins





Generative AI and Foundation Models

Toward (Generative) **World Models**



WORLD MODEL

A world model is an AI system that can simulate and predict how the world might change given current conditions and potential actions or events.

Vision:

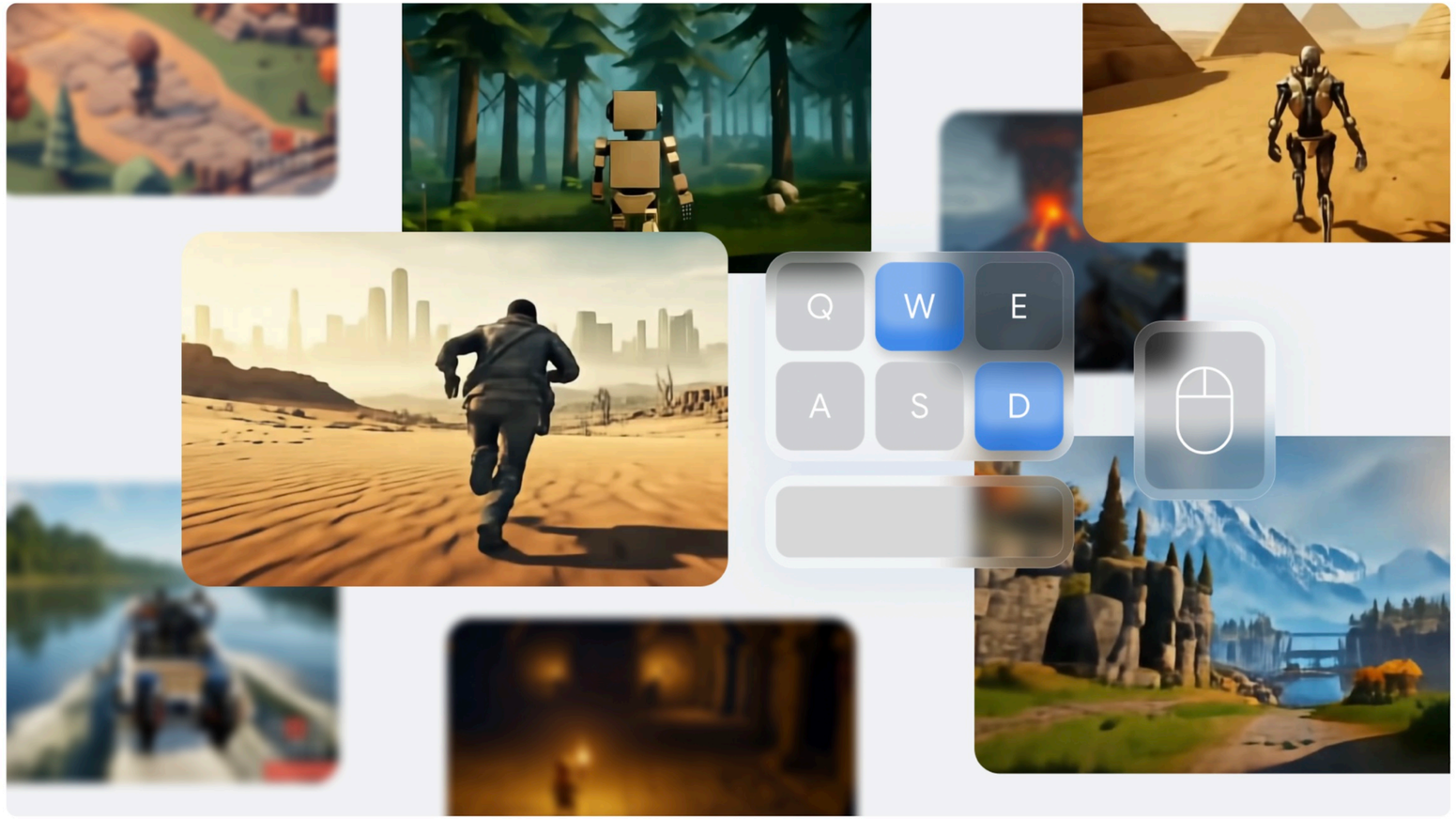
Foundation models trained on diverse data are developing increasingly sophisticated world models.

Simulations through generative AI enable genuine reasoning, planning, and understanding of cause-and-effect relationships in the real world.

Lecture 12: World Models

Generative AI and Foundation Models

Toward (Generative) **World Models**



e.g., **DeepMinds's Genie 2 or 3**

Proto-engine for generating video games *on-the-fly*.

Lecture 13: FM in Robotics

Lecture 12: World Models

Foundation Models Are More Than Language Models

Thomas G. Dietterich
@tdietterich

I propose that we adopt the term "Large Self-Supervised Models (LSSMs)" as a replacement for "Foundation Models" and "LLMs". "LLMs" don't capture non-linguistic data and "Foundation Models" is too grandiose. Thoughts? @percyliang

2:58 am · 13 Aug 2022

Percy Liang @percyliang · 13 Aug 2022

The beauty of language is that you can have multiple terms that highlight different aspects of the same object. You don't have to choose. I use "LLM" to talk about LLMs, "self-supervised" for their construction, and "foundation model" for their function. No term can be replaced.

1 4 33

Stanford | Bulletin
ExploreCourses

1 - 1 of 1 results for: **CS 324: Advances in Foundation Models**

CS 324: Advances in Foundation Models

Foundation models (FMs) are transforming the landscape of AI in research and industry. Such models (e.g., GPT-3, CLIP, Stable Diffusion) are trained on large amounts of broad data and are adaptable to a wide range of downstream tasks. In this course, students will learn fundamentals behind the models and algorithms, systems and infrastructure, and ethics and societal impacts of foundation models, with an emphasis on gaining hands-on experience and identifying real-world use-cases for FMs. Students will hear from speakers in industry working on foundation models in the wild. The main class assignment will be a quarter-long final project, involving either researching the capabilities of FMs or building an FM-powered application.

Terms: Win | Units: 3

Instructors: Hashimoto, T. (PI) ; Liang, P. (PI) ; Re, C. (PI) ... more instructors for CS 324 »

Scepticism is Crucial

- **Challenge the *status quo***: Question dominant paradigms, they've been wrong before.
- **Rapid evolution**: Methods are very new with constant developments.
  The next breakthrough could come from you 😉
- **Success recipes expire**: What works today may be obsolete tomorrow. Current learning principles and architectures might seem primitive in a few years.
- **Performance ≠ promises**: Hallucinations, poor generalization, brittleness.
  Always evaluate critically before deployment.
- **Ethics are non-negotiable**: What data? Whose voices? What harms? The list of concerns is long…

Bottom Line: Maintain healthy skepticism. Question everything.
The field needs critical thinkers who separate hype from reality.

Lecture 15: Outlook and Summary

Course Goals

- Describe and explain **core generative modeling techniques** and their **conceptual role in foundation models**.
- Analyze and compare the **architectures**, **tokenization strategies**, and **training objectives** of foundation models across language, vision, and scientific domains.
- **Apply** suitable foundation models or generative approaches for a given task and justify their use based on model capabilities and data modality.
- Investigate and interpret recent advances in **multi-modal learning**, prompting, and **decision-making with foundation models**.
- Critique and synthesize key contributions from *current research papers* by relating them to concepts and methods covered in the course.

Course Prerequisites

- **Required**

  CS-233 Introduction to Machine Learning

  CS-433 Machine Learning or equivalent course

- **Recommended**

  EE-559 Deep Learning or equivalent course

- **Complementary**

  CS-503 Visual intelligence: Machines and Minds

  CS-552 Modern Natural Language Processing

Course Schedule

Week	Part I	Week	Part II
1	Introduction and Overview	8	Multi-Modality in Foundation Models
2	**Learning at Scale**: Supervised, Self-Supervised, and Beyond	9	**Architectures II**: Foundation Models in the Sciences
3	**Generative Models I**: Autoregressive, Adversarial, and Autoencoder	10	In-Context Learning and Emergent Behaviors
4	**Generative Models II**: Diffusion Models and Beyond	11	Adaptation, Fine-Tuning, and Test-Time Training
5	**Generative Models III**: Flow Matching and Schrödinger Bridges Tokenization Across Modalities and Building Blocks	12	World Models and Generative World Modeling
6	**Architectures I**: Language and Vision Foundation Models	13	**Architectures III**: FMs in Robotics Foundation Models, Reinforcement Learning, Reasoning, and Decision-Making
7	*Semester Break*	14	Foundation Models and Agentic Systems
		15	Outlook and Summary

Schedule Every Week

Lecture — 2 hours in PO 01 given by professor or guest lecturer.

Slides of lecture will be uploaded latest 12 pm on Moodle.

We aim to release recordings of the lecture by Wednesday at the latest. Availability is not guaranteed.

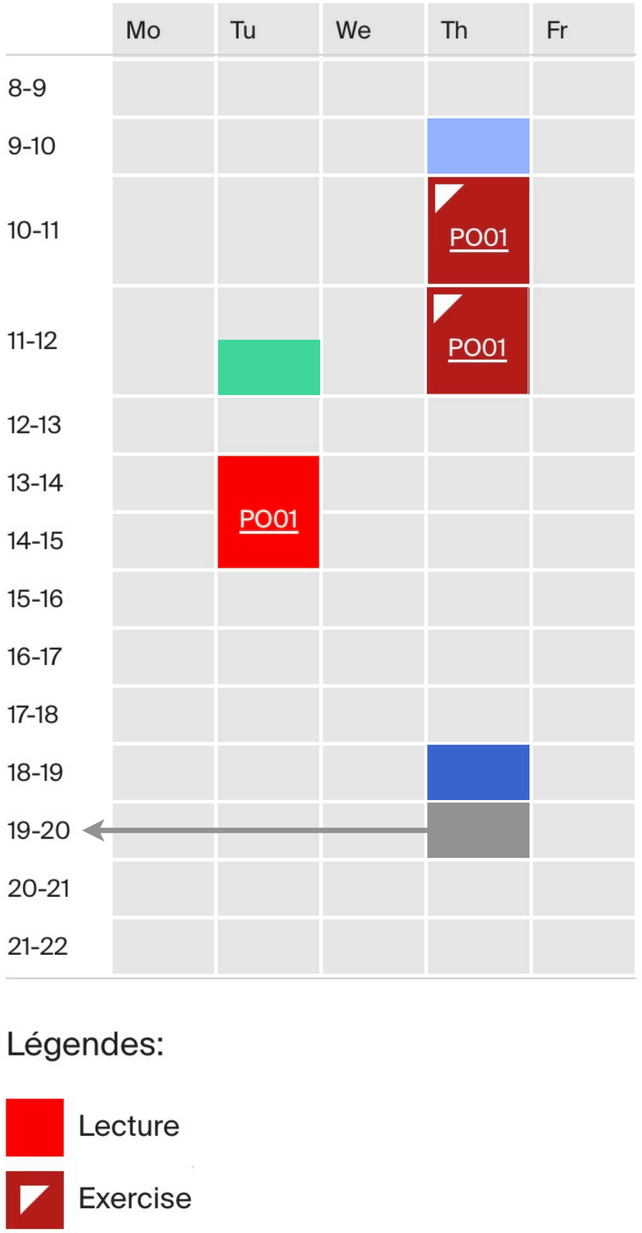
Exercise — 2 hours in PO 01 given by teaching assistants.

Thursday, 7pm, the week before: Exercise sheet for the upcoming week released on Moodle.

Solutions sheet and notebook for code demonstrations will be uploaded 9 am on Moodle.

Slides and other material of exercise session will be uploaded to Moodle latest until the evening.

There will be no recordings of the exercise sessions.

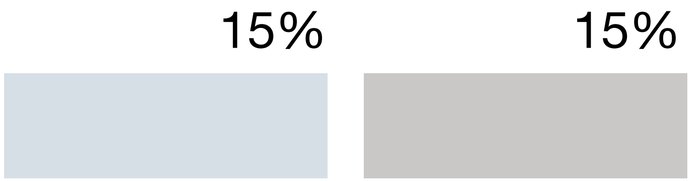


Course Grading

Weekly exercises and quizzes are ungraded!

Written Exam during an exam session in January.

Exam date will be published by the Registrar's Office.
Closed book exam. One A4 crib sheet (both sides).



Coding Assignments during the semester. No group work. Each assignment will be released one week in advance and is due on Wednesday at 23:59 that week.

1. **Assignment** is due in **Week 5**, i.e., Wednesday, October 8.
2. **Assignment** is due in **Week 12**, i.e., Wednesday, November 26.

Please keep an eye out for upcoming announcements via Ed and updates via Moodle!

Course Team

Teaching Assistants

Lukas Klein

POSTDOC

Johann Wenckstern

DOCTORAL STUDENT

Xiuying Wei

DOCTORAL STUDENT

Liangze Jiang

DOCTORAL STUDENT

Linus Bleistein

POSTDOC

Eeshaan Jain

DOCTORAL STUDENT

Abdulkadir Gokce

DOCTORAL STUDENT

Student Assistants

Petr Grinberg

MASTER STUDENT

Matteo Santelmo

MASTER STUDENT

Course Logistics

Course Overview

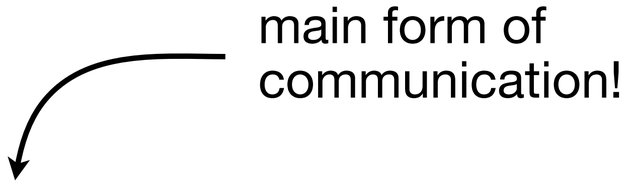
https://edu.epfl.ch/coursebook/en/foundation-models-and-generative-ai-CS-461

Moodle

https://go.epfl.ch/CS-461

Ed

https://edstem.org/eu/courses/2320/discussion



Course Integrity Policy

- For the assignments, you **should not use outside codebases** unless explicitly allowed by the course staff in the assignment description. **You can use ChatGPT or other AI-based** tools for any assignment or part of your project. Any use of ChatGPT and other AI-based tools must be cited and mentioned. Uncited use of these tools will be penalized.

- For the code assignments, you may build your work upon existing open-source codebases, but are **required to write new code** to perform your experiments. In the code assignment, clearly specify your contributions and how they differ from the pre-existing codebase in your report.

- You are free to discuss ideas and implementation details with your colleagues. However, you **should not look at another student's code**, or incorporate their code into your assignment (unless explicitly allowed by the course staff).

This Week's Papers

Papers are linked in Moodle.


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, a difference in the amount of computation is inevitable. 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 often do. 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.

In computer chess, the methods that defeated the world champion, Kasparov, in 1997, were based on massive, deep search. At the time, this was looked upon with special disfavor by most of computer-chess researchers who had been pursuing methods that leveraged human understanding of the special structure of chess. When a simpler, search-based approach with special hardware and software proved vastly more effective, these human-knowledge-based chess researchers were not good losers. They said that "brute force" search may have won this time, but it was not a general strategy, and anyway it was not how people played chess. These researchers wanted methods based on human input to win and were disappointed when they did not.

A similar pattern of research progress was seen in computer Go, only delayed by a further 20 years. Enormous initial efforts went into avoiding search by taking advantage of human knowledge, or of the special features of the game, but all those efforts proved irrelevant, or worse, once search was applied effectively at scale. Also important was the use of learning by self play to learn a value function (as was also learning in many other games and even in chess, although learning did not play a big role in the 1997 program that first beat a world champion). Learning by self play, and learning in general, is like search in that it enables massive computation to be brought to bear. Search and learning are the two most important classes of techniques for utilizing massive amounts of computation in AI research. In computer Go, as in computer chess, researchers' initial effort was directed towards utilizing human understanding (so that less search was needed) and only much later was much greater success had by embracing search and learning.

In speech recognition, there was an early competition, sponsored by DARPA, in the 1970s. Entrants included a host of special methods that took advantage of human knowledge—knowledge of words, of phonemes, of the human vocal tract, etc. On the other side were newer methods that were more statistical in nature and did much more computation, based on hidden Markov models (HMMs). Again, the statistical methods won out over the human-knowledge-based methods. This led to a major change in all of natural language processing, gradually over decades, where statistics and computation came to dominate the field. The recent rise of deep learning in speech recognition is the most recent step in this consistent direction. Deep learning methods rely even less on human knowledge, and use even more computation, together with learning on huge training sets, to produce dramatically better speech recognition systems. As in

Sutton, Richard. "The Bitter Lesson." Incomplete Ideas (Blog) 13.1 (2019): 38.

Bommasani et al., "On the opportunities and risks of foundation models." arXiv preprint arXiv:2108.07258 (2021).

This Week's Exercise Sheet

There will be no exercise session this week.

This Week's Code Demonstration

There will be no code demonstration this week.

CS-461

Foundation Models and Generative AI

Let's have a great semester!