CSC2541H / (PCL3107H, PCL3108H): AI for Drug Discovery

Large Genomics Models

Chris J. Maddison
University of Toronto

Learning Objectives

By the end of this lecture and next week, you will be able to:

1. Describe the large language model training pipeline: data collection, tokenization, pre-training, and post-training
2. Explain how scaling laws and emergence demonstrate that data abundance drives capabilities
3. Understand why genomics is the most abundant "free" data in biology
4. Compare large genomics models: ESM2, Evo 2, and AlphaGenome

Revisiting Modern AI

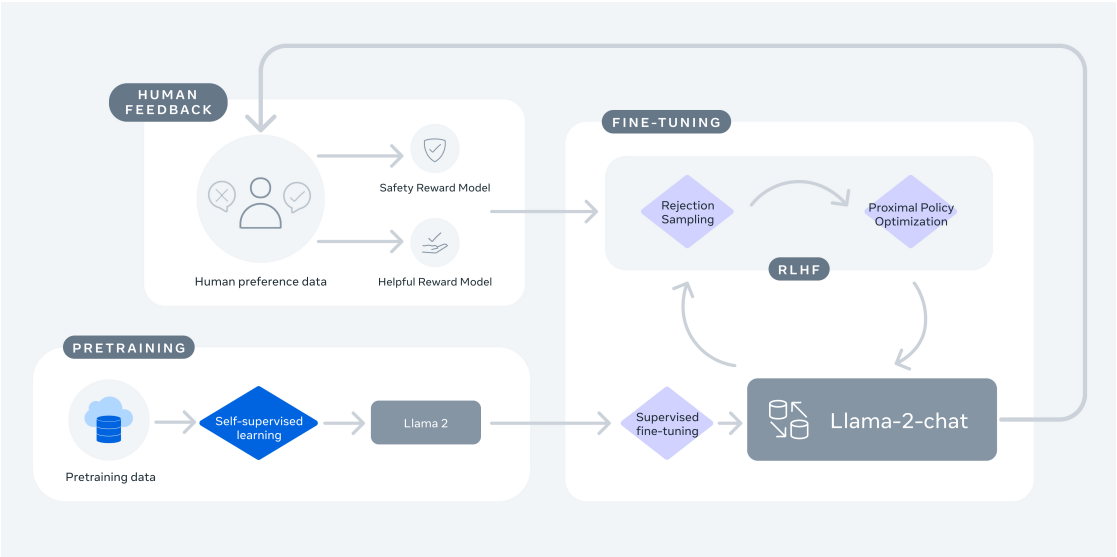
Recall from Lecture 1: Modern AI departs from classical ML in three ways:

- **Next-token prediction:** A task formulation that uses abundant internet data
- **Flexible predictors:** Neural networks with billions of parameters
- **Massive scale:** Trillions of tokens, billions of parameters

Today: We will unpack how these large language models are actually built

Large Language Models (LLMs) are grown, not made

$$\text{Data Collection} \rightarrow \text{Tokenization} \rightarrow \overbrace{\text{Pre-training} \rightarrow \text{Post-training}}^{\text{Model Training}}$$

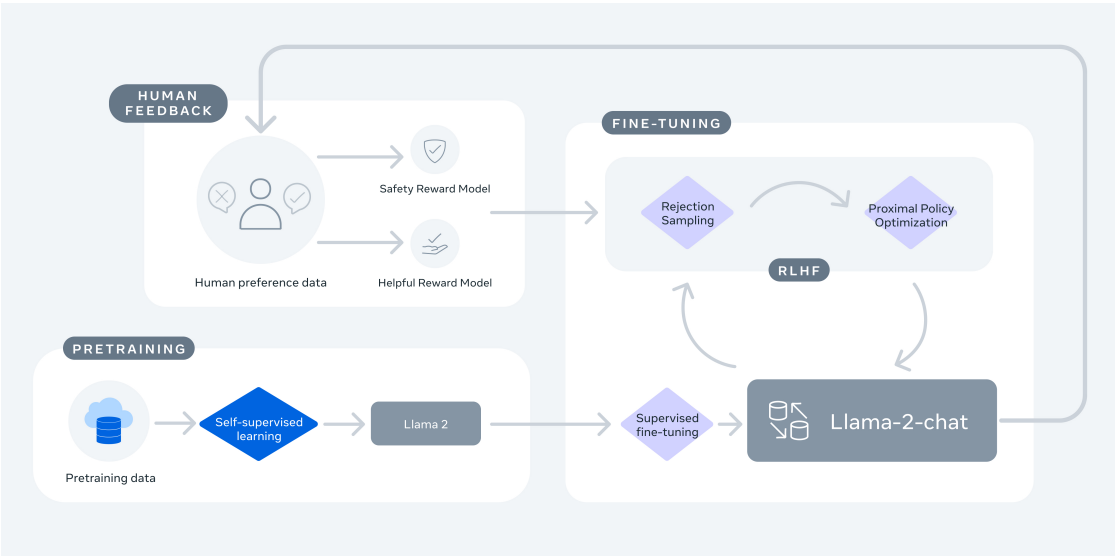


Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models", 2023

- **LLMs are Transformer neural nets, trained in stages.**
  - We will cover Meta's Llama models, but this applies broadly
- Developmental biology may be a better analogy

Large Language Models (LLMs) are grown, not made

$$\text{Data Collection} \rightarrow \text{Tokenization} \rightarrow \overbrace{\text{Pre-training} \rightarrow \text{Post-training}}^{\text{Model Training}}$$



Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models", 2023

LLMs train on text: how is it collected and processed?

Data Collection—Where Does the Data Come From?

Short answer: Many sources, largely the internet.

- **Common Crawl** is a widely-used open repository containing over 300 billion web pages
- Many LLMs use Common Crawl as a starting point, then filter and augment
- The exact data sources and filtering pipelines are typically **proprietary**—kept secret for competitive reasons and to avoid copyright scrutiny

What Llama 3 discloses:

- Extensive filtering: PII removal, deduplication, model-based quality scoring
- Custom HTML parser optimized for math and code content

Dubey et al., "The Llama 3 Herd of Models", 2024

Tokenization—From Text to Tokens

- Neural networks cannot directly process text—they need numbers
- **Tokenization:** Converting text into discrete units (tokens)

Example: "The drug binds to the receptor"

$\rightarrow$ ["The", "drug", "bind", "s", "to", "the", "recept", "or"]

$\rightarrow$ [1024, 8572, 4521, 82, 311, 1024, 56712, 269]

Key insight: Tokens are not words—they are **subwords**, common pieces of text that balance vocabulary size with meaning.

Llama 3: 128,000 possible tokens (common words, word pieces, punctuation)

Dubey et al., "The Llama 3 Herd of Models", 2024

Tokenization—Try the Tokenizer

`https://platform.openai.com/tokenizer`

Try typing:

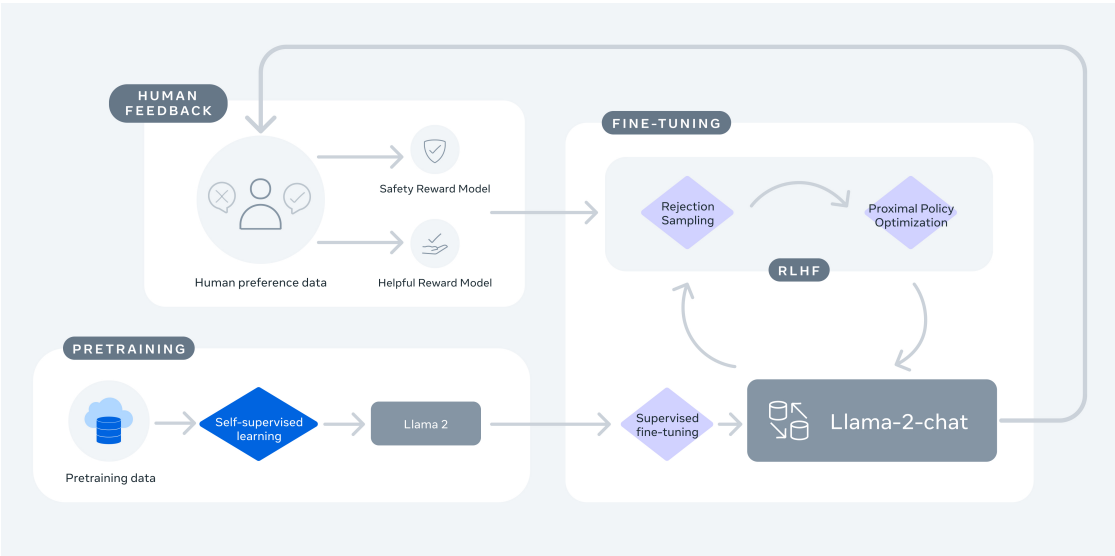
- A drug name: "acetaminophen" vs "Tylenol"
- A sentence: "The IC50 was 2.5 nM"
- Technical jargon vs common words

Notice:

- Common words = single token; rare words = multiple tokens
- Numbers are often split into individual digits
- Each color = one token

Large Language Models (LLMs) are grown, not made

$$\text{Data Collection} \rightarrow \text{Tokenization} \rightarrow \overbrace{\text{Pre-training} \rightarrow \text{Post-training}}^{\text{Model Training}}$$



Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models", 2023

LLMs are Transformer neural networks: how do they work?

Model—What is a Transformer?

Recall MNIST: A neural network maps an input (image) to an output (digit probabilities)

An LLM is a Transformer neural network, which models sequences:

- Alternates between two types of layers:
  - Traditional neural network layers (like MNIST)
  - **Attention layers:** compute pairwise comparisons between tokens
- Handles **variable-length input**
- General-purpose: used for translation, classification, generation, etc.

For LLMs: Token IDs (integers) $\rightarrow$ Transformer $\rightarrow$ probability over next token

Vaswani et al., "Attention Is All You Need", 2017

Visualizing Attention in Transformers

`https://www.youtube.com/watch?v=wjZofJX0v4M`

3Blue1Brown's visual explanation of how attention works

Model—The Attention Layer

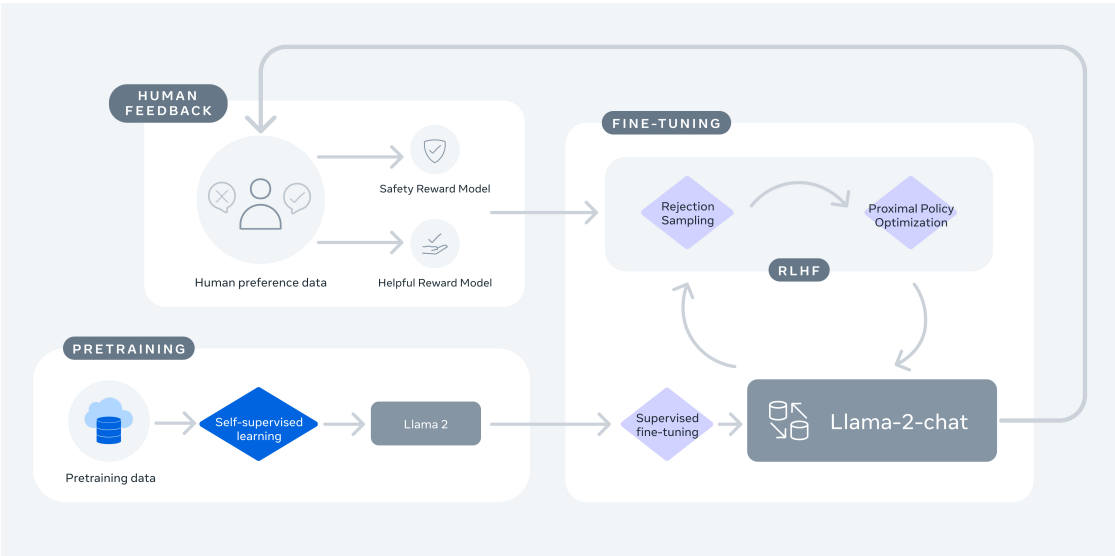
The attention layers compare every token to every other token

Quadratic scaling:

- With n tokens, attention makes n^2 pairwise comparisons
- $2\times$ context length $\rightarrow$ $4\times$ the computation
- This cost is mostly **felt at test-time**
- **Training however is highly parallelizable:** All comparisons can happen simultaneously on GPUs

Large Language Models (LLMs) are grown, not made

Data Collection → Tokenization → Model Training: Pre-training → Post-training



Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models", 2023

LLMs are pre-trained on vast datasets of text documents.

Pre-training—Next-Token Prediction

Llama 3 architecture:

- Standard dense Transformer (deliberately simple for stability)
- Context length: **128K tokens** (32× Llama 2)
- Model sizes: 8B, 70B, **405B parameters**

Training objective: Predict next token given all previous tokens

Dubey et al., "The Llama 3 Herd of Models", 2024

`The drug binds to`

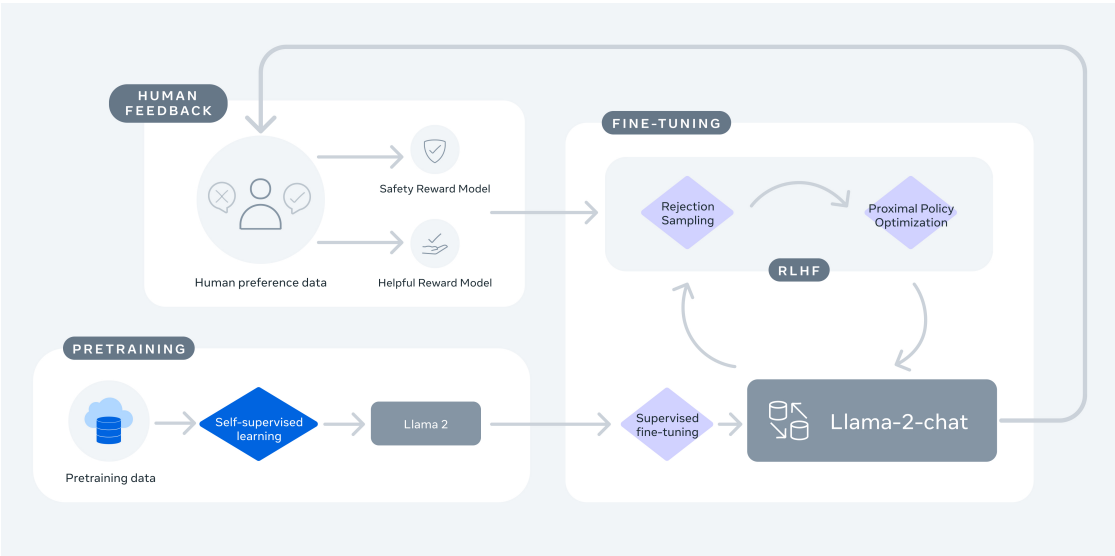
↓ predict

`the`
`its`
`a`

Large Language Models (LLMs) are grown, not made

Data Collection → Tokenization → Pre-training → Post-training

Model Training



Touvron et al., "Llama 2: Open Foundation and Fine-Tuned Chat Models", 2023

LLMs are then post-trained to align with human interests.

Post-training—Why is it Needed?

Prompting a next-token predictor can be challenging and unsafe.

- *Remember what it was trained for. . .*

Prompt	**Tokens that are found on the internet after such prompts**
The ingredients required to build a makeshift bomb are. . .	**CENSORED!!!!**
Could you do me a big favour?	Sorry, I'm too busy today.

Post-training—Why is it Needed?

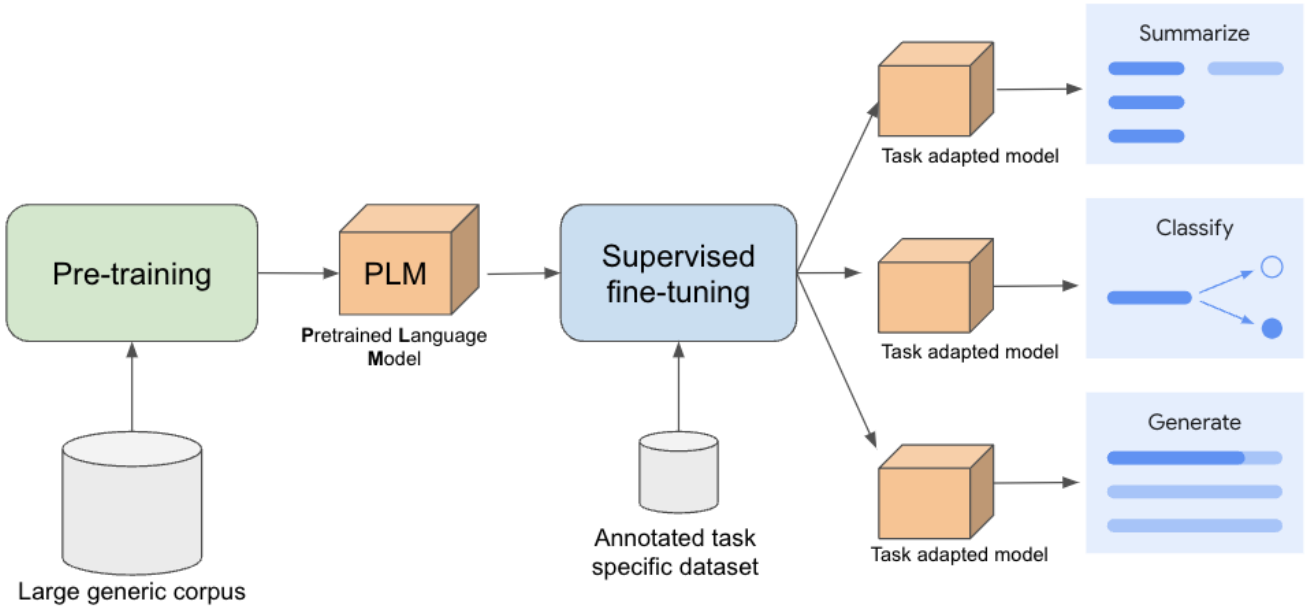
- Pre-training produces a "completion engine" that predicts likely continuations
- But users want an **assistant** that follows instructions and is helpful/safe
- **Post-training** aligns the model with human preferences

Llama 3's approach (simpler than Llama 2):

1. **Learning from ideal responses:** Learn from ideal human and synthetic examples
2. **Human preference data:** Learn from preference pairs

Dubey et al., "The Llama 3 Herd of Models", 2024

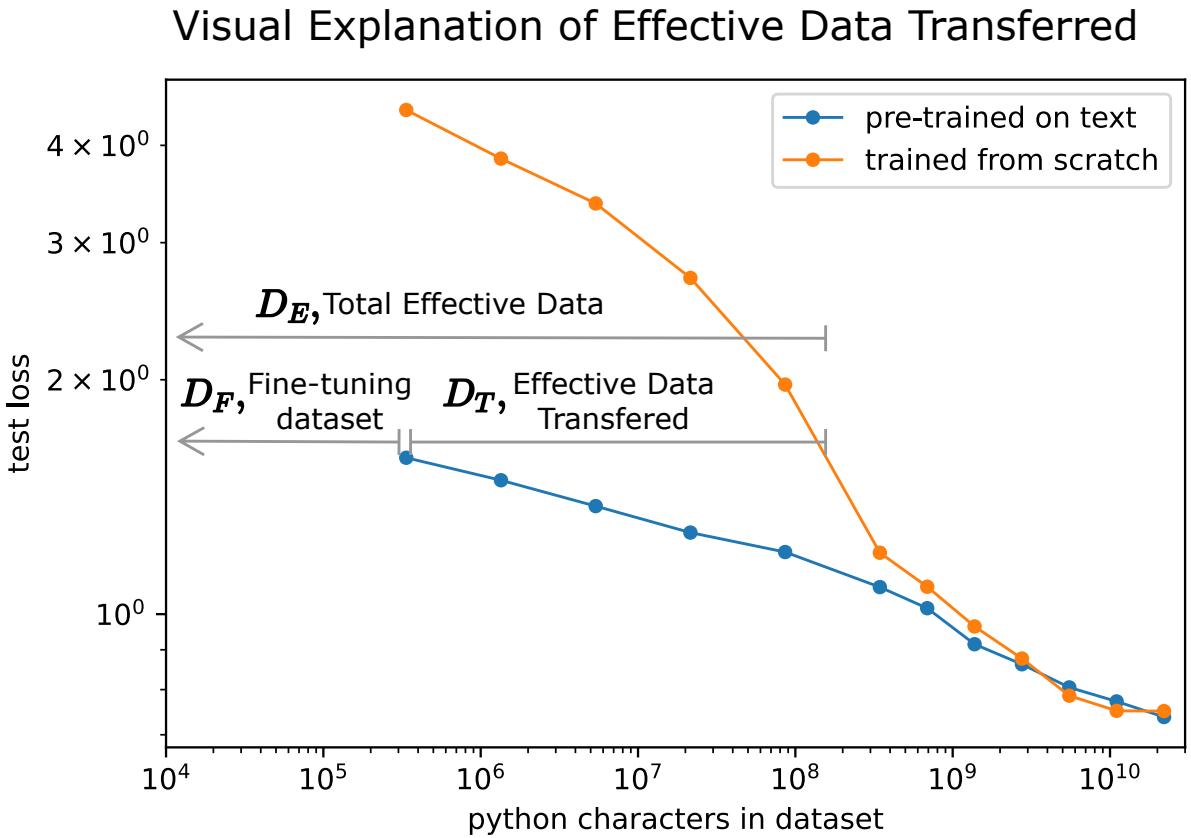
Post-training—Learning from ideal responses



Huizenga and Hu, "When to use supervised fine-tuning for Gemini"

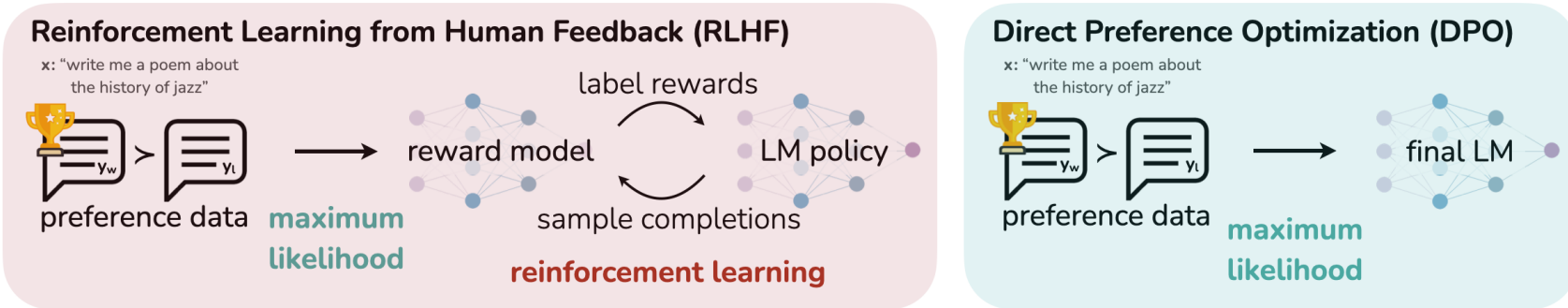
- Train on examples of ideal responses (human + synthetic), called **Supervised Fine-Tuning (SFT)**.
- Llama 3 made heavy use of **synthetic data generation approaches** for code, math, and long-context
  - Data quality filtering using both reward models and Llama-as-judge

Post-training—Information transfers from pre-training



Hernandez et al. "Scaling Laws for Transfer". 2021.

Post-training—RLHF and DPO



Rafailov et al., "Direct Preference Optimization", 2024

- Humans compare two model outputs and rank them
- **Reinforcement Learning from Human Feedback (RLHF)** and **Direct Preference Optimization (DPO)** two ways to optimize the LLM to prefer the higher ranked outputs

Results

Llama 3 405B on major benchmarks:

- MMLU (5-shot): **87.3%** (vs GPT-4: 85.1%, Claude 3.5: 89.9%)
- GSM8K math (8-shot): **96.8%** (vs GPT-4: 94.2%)
- HumanEval code: **89.0%** (vs GPT-4: 86.6%)
- MATH (0-shot): **73.8%** (vs GPT-4: 64.5%)

MATH Benchmark Example

MATH Dataset (Ours)

Problem: Tom has a red marble, a green marble, a blue marble, and three identical yellow marbles. How many different groups of two marbles can Tom choose?

Solution: There are two cases here: either Tom chooses two yellow marbles (1 result), or he chooses two marbles of different colors ($\binom{4}{2} = 6$ results). The total number of distinct pairs of marbles Tom can choose is $1 + 6 = \boxed{7}$.

Problem: The equation $x^2 + 2x = i$ has two complex solutions. Determine the product of their real parts.

Solution: Complete the square by adding 1 to each side. Then $(x+1)^2 = 1 + i = e^{\frac{i\pi}{4}}\sqrt{2}$, so $x + 1 = \pm e^{\frac{i\pi}{8}}\sqrt[4]{2}$. The desired product is then $\left(-1 + \cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right)\left(-1 - \cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right) = 1 - \cos^2\left(\frac{\pi}{8}\right)\sqrt{2} = 1 - \frac{\left(1+\cos\left(\frac{\pi}{4}\right)\right)}{2}\sqrt{2} = \boxed{\frac{1-\sqrt{2}}{2}}$.

Hendrycks et al., "Measuring Mathematical Problem Solving With the MATH Dataset", 2021

Data Scale in Context

- Llama 3 trained on **15 trillion tokens**
- Flagship model: **405 billion parameters**
- Context window: up to **128K tokens**

Key takeaway: The scale is staggering.

Dubey et al., "The Llama 3 Herd of Models", 2024

Section Summary – Building LLMs

- **Data Collection:** Trillions of tokens, largely from the internet
- **Tokenization:** Converts text documents into sequences of integers
- **Pre-training:** Next-token prediction at massive scale
- **Post-training:** Aligning with human preferences

Next question: What role does this massive data play in capabilities?

The Role of Data

We have seen how LLMs are built.

Now: Why does scale matter so much?

Key questions:

- How does performance relate to scale?
- Do new capabilities "emerge" at certain scales?
- Will we run out of data?

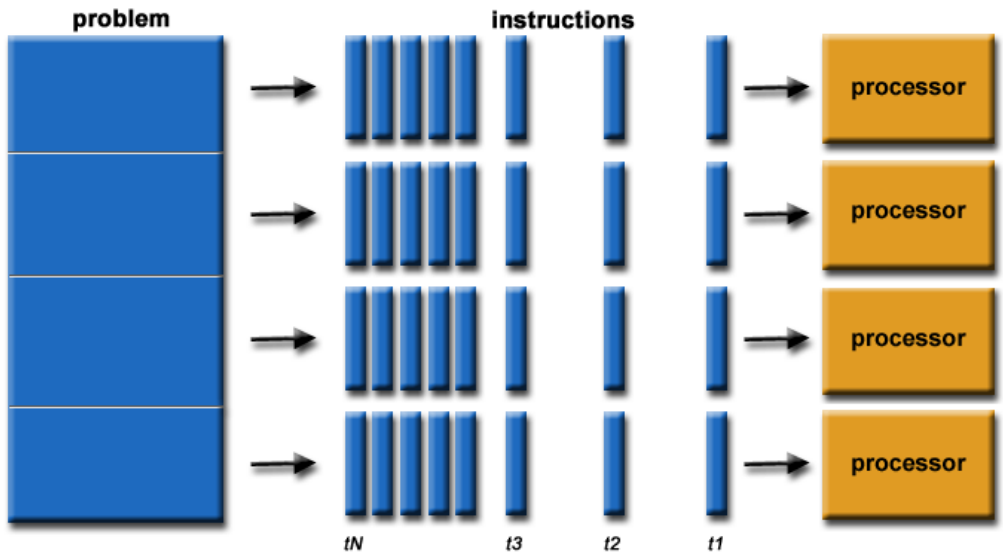
Compute As a Key Scale Metric

$$3.14159 \times 2.71828 = 8.53972$$

One floating point operation (FLOP)

- Training LLMs $\rightarrow$ floating point operations (FLOPs)
- Each FLOP $\rightarrow$ time and energy

Compute As a Key Scale Metric



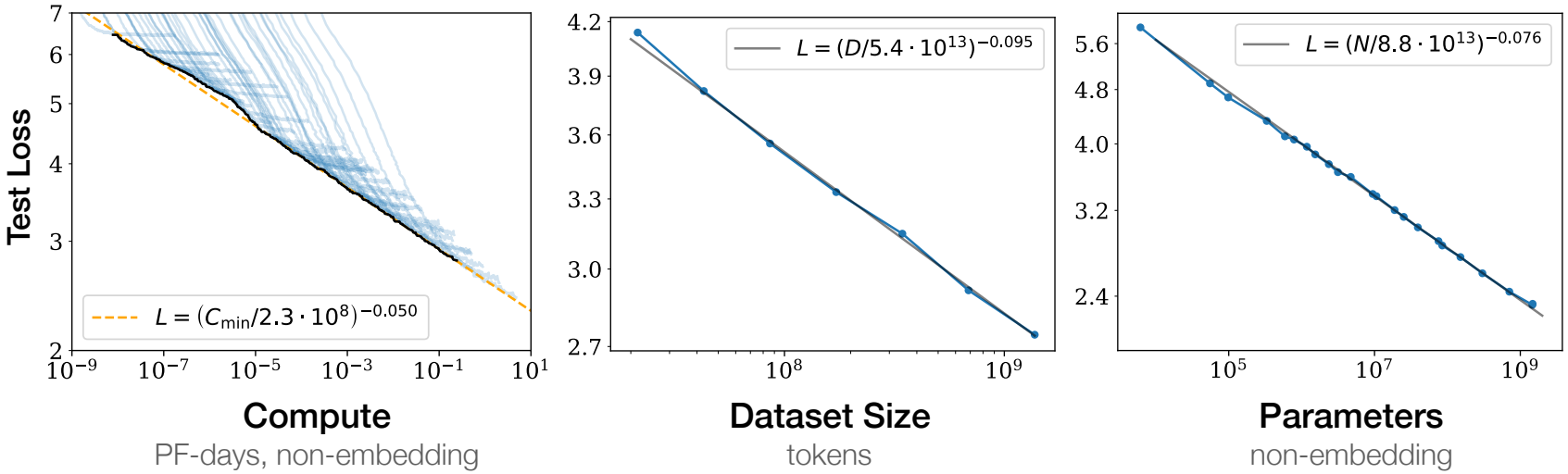
"Parallel Architecture" Harshita Sharma

- On modern hardware we can save **time** by parallelizing FLOPs but each one still costs **energy**
- Each FLOP $\rightarrow$ has a real cost

Compute As a Key Scale Metric

- **The cost of the training run can be estimated by counting the number of FLOPs executed during training**
- Crucially, the number of FLOPs required to train increases as we add parameters to an LLM and as we train it on more data

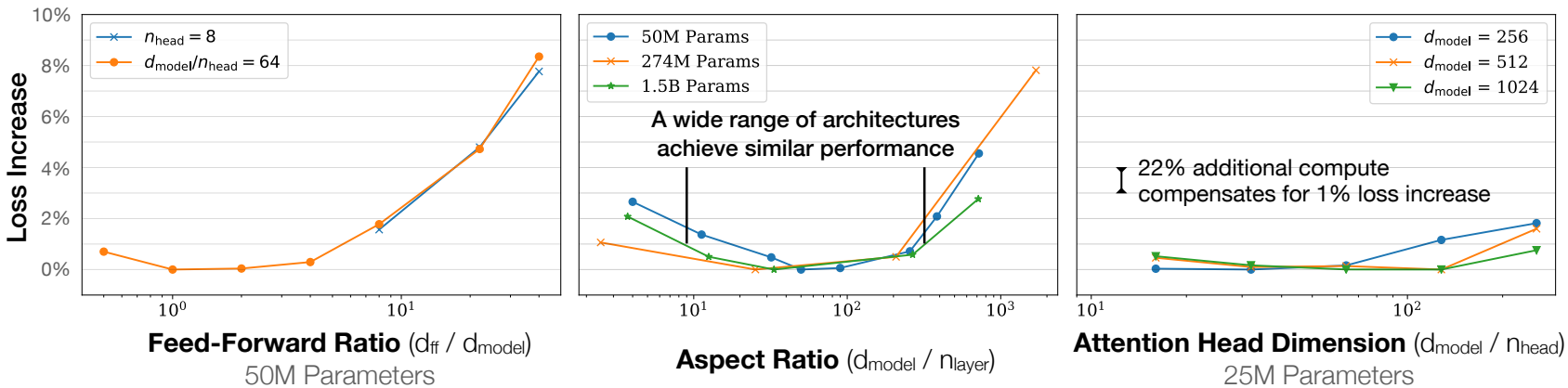
Scaling Laws – A Discovery



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

Finding: The next-token prediction loss improves smoothly like a power law in terms of the compute invested during training

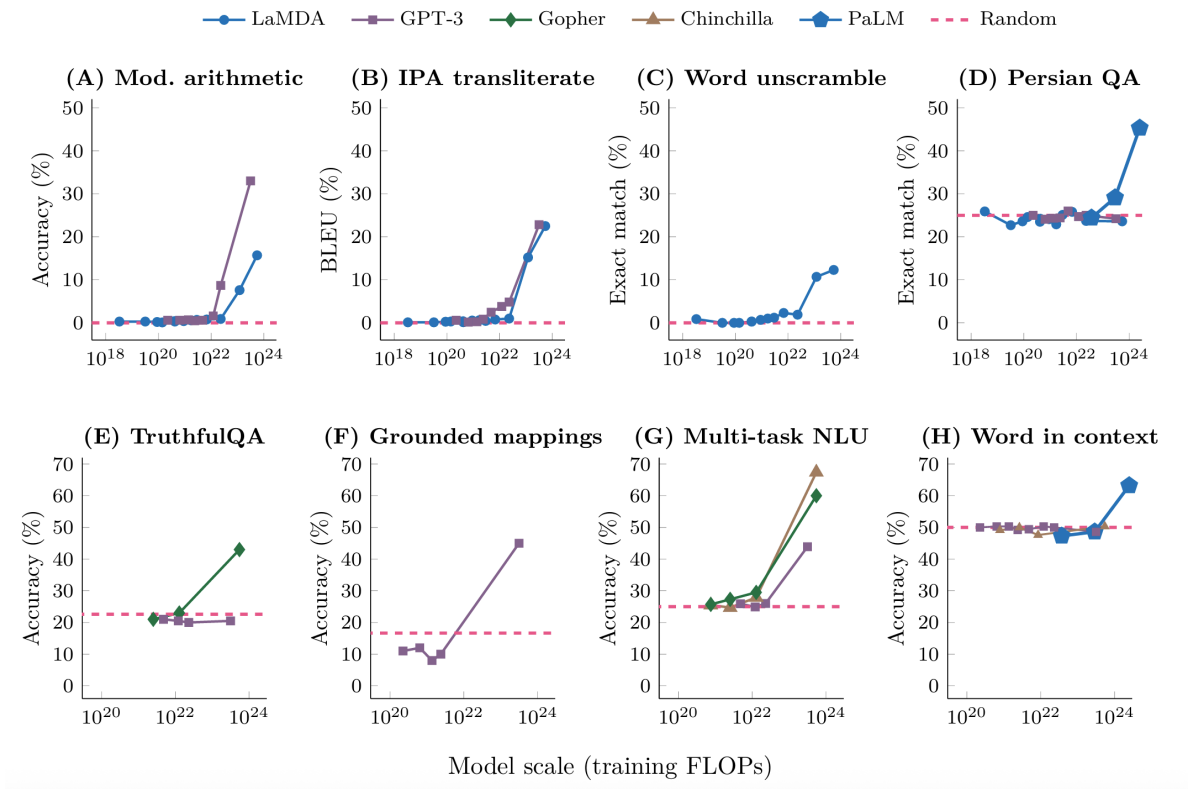
Architecture Matters Less Than Scale



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

Finding: Performance depends weakly on architectural hyperparameters such as depth vs. width. This is why "just scale it up" has become the dominant strategy in AI

Emergence of Capabilities



Wei et al., "Emergent Abilities of Large Language Models", TMLR, 2022

Finding: Certain abilities appear **suddenly and unpredictably** at specific scale thresholds *in pre-training*

Emergence Thresholds – Capabiltiies

	Emergent scale			
	Train. FLOPs	Params.	Model	Reference
Few-shot prompting abilities				
• Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3	Brown et al. (2020)
• Addition/subtraction (4-5 digit)	3.1E+23	175B		
• MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3	Hendrycks et al. (2021a)
• Toxicity classification (CivilComments)	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Truthfulness (Truthful QA)	5.0E+23	280B		
• MMLU Benchmark (26 topics)	5.0E+23	280B		
• Grounded conceptual mappings	3.1E+23	175B	GPT-3	Patel & Pavlick (2022)
• MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla	Hoffmann et al. (2022)
• Word in Context (WiC) benchmark	2.5E+24	540B	PaLM	Chowdhery et al. (2022)
• Many BIG-Bench tasks (see Appendix E)	Many	Many	Many	BIG-Bench (2022)

Wei et al., "Emergent Abilities of Large Language Models", TMLR, 2022

Specific numbers show the scale required for different capabilities

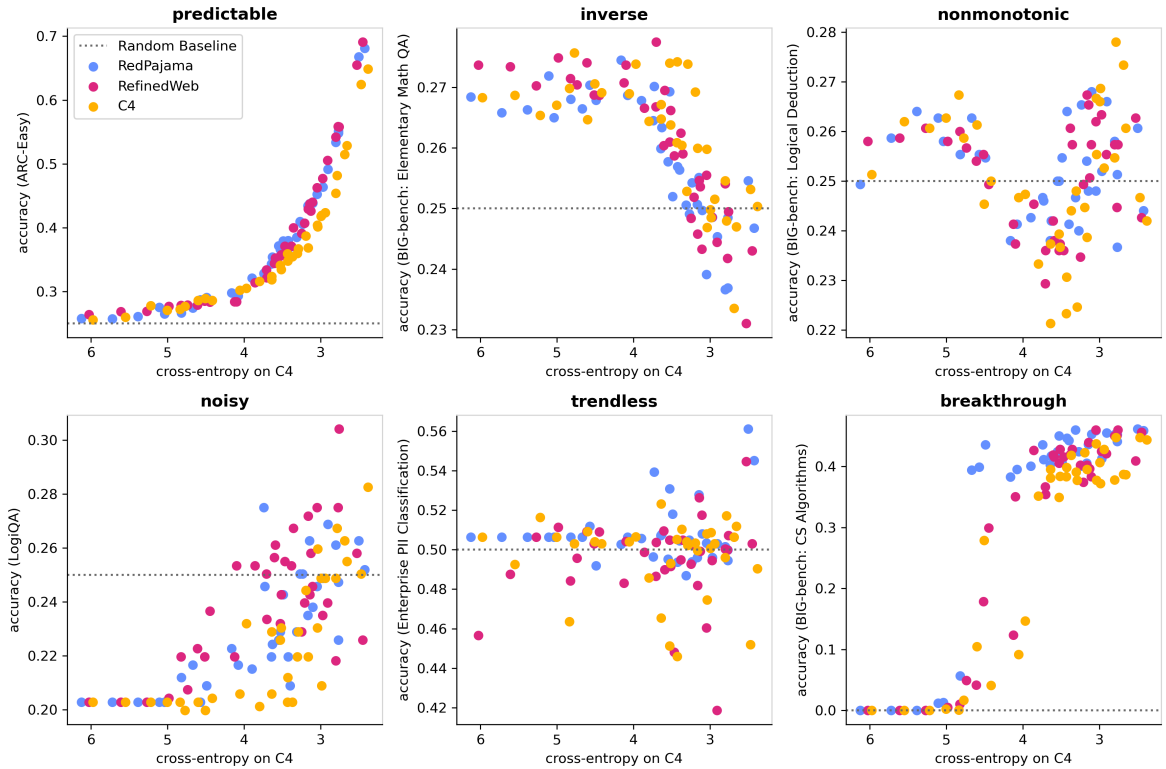
Emergence Thresholds – Prompting Techniques

	Emergent scale			
	Train. FLOPs	Params.	Model	Reference
Augmented prompting abilities				
• Instruction following (finetuning)	1.3E+23	68B	FLAN	Wei et al. (2022a)
• Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA	Nye et al. (2021)
• Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA	Wei et al. (2022b)
• Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM	Chowdhery et al. (2022)
• Differentiable search index	3.3E+22	11B	T5	Tay et al. (2022b)
• Self-consistency decoding	1.3E+23	68B	LaMDA	Wang et al. (2022b)
• Leveraging explanations in prompting	5.0E+23	280B	Gopher	Lampinen et al. (2022)
• Least-to-most prompting	3.1E+23	175B	GPT-3	Zhou et al. (2022)
• Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3	Kojima et al. (2022)
• Calibration via P(True)	2.6E+23	52B	Anthropic	Kadavath et al. (2022)
• Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM	Shi et al. (2022)
• Ask me anything prompting	1.4E+22	6B	EleutherAI	Arora et al. (2022)

Wei et al., "Emergent Abilities of Large Language Models", TMLR, 2022

Not just tasks—even the success of prompting **techniques** can be emergent!

Warning: relationship between test loss and capabilities can be mixed



Lourie et al. "Scaling Laws Are Unreliable for Downstream Tasks: A Reality Check". 2025.

Why is next token prediction such a powerful pre-text task?

- Post-training is necessary for maximal performance, but even pre-training models demonstrate the emergence of capabilities at scale.
- Why is such a simple task, i.e., next token prediction, such a good pre-text task?
- **Hypothesis:** predicting the next token well at internet scale requires a sophisticated understanding of the process that generated the data—i.e., the world

If we need vast data sets to improve our models—will we be able to keep improving the models?

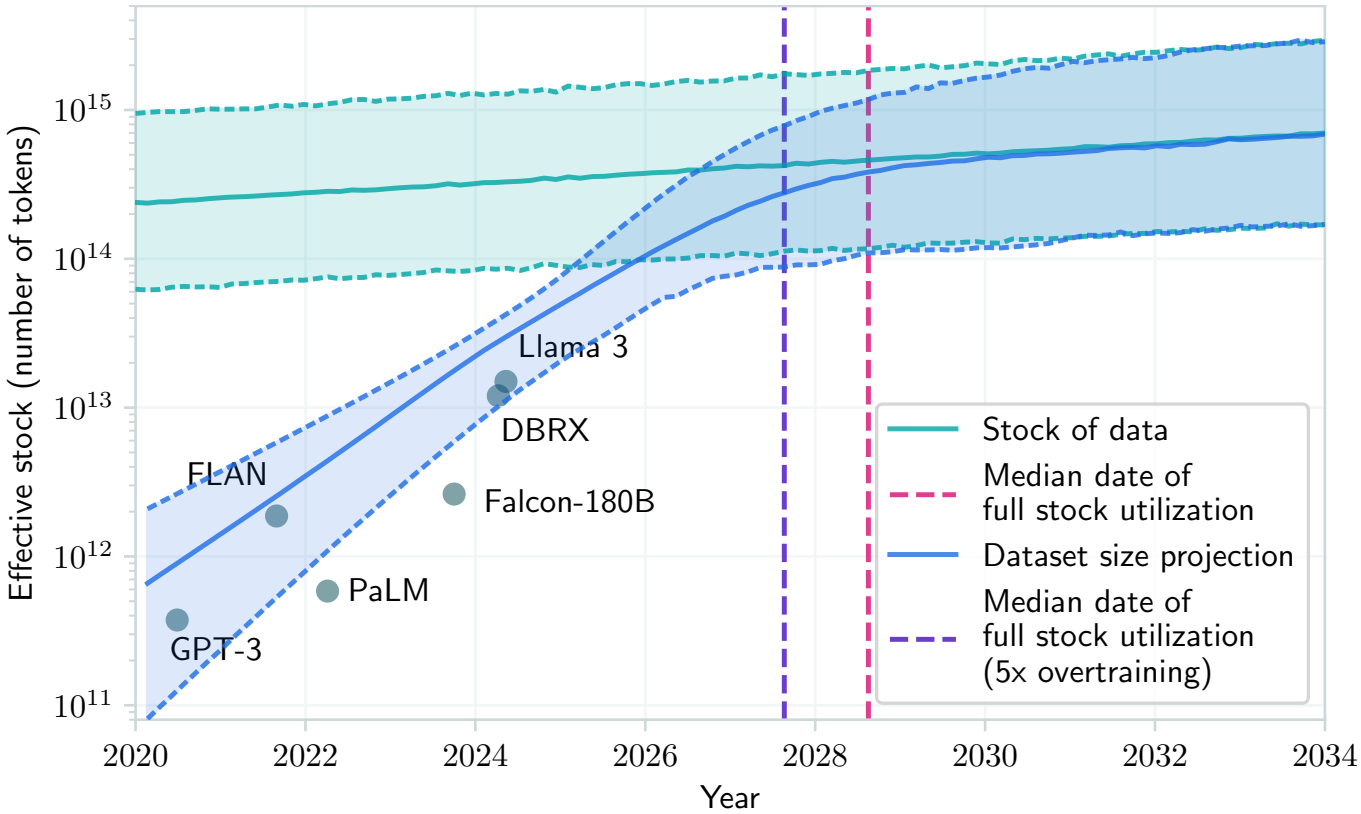
Will We Run Out of Data?

Key challenge: The success of these models relies on human-generated public text data, which may not sustain scaling beyond this decade.

- Current largest datasets: $\sim$10-15T tokens (Llama 3 used 15T)
- Villalobos et al. (2024) estimate that demand for data from AI companies growing at 2.4$\times$ per year whereas growth of data stock is only at 0% to 10% per year.

Villalobos et al., "Will we run out of data?", ICML, 2024

Data Stock Estimates



- **Median exhaustion year: 2028**
- By 2032, exhaustion becomes "very likely"

Villalobos et al., "Will we run out of data?", ICML, 2024

Section Summary – The Role of Data

- **Scaling Laws:** Performance improves predictably with scale (7+ orders of magnitude)
- **Emergence:** Capabilities appear suddenly at specific thresholds—unpredictable
- **The Challenge:** We may exhaust public text data by $\sim$2028

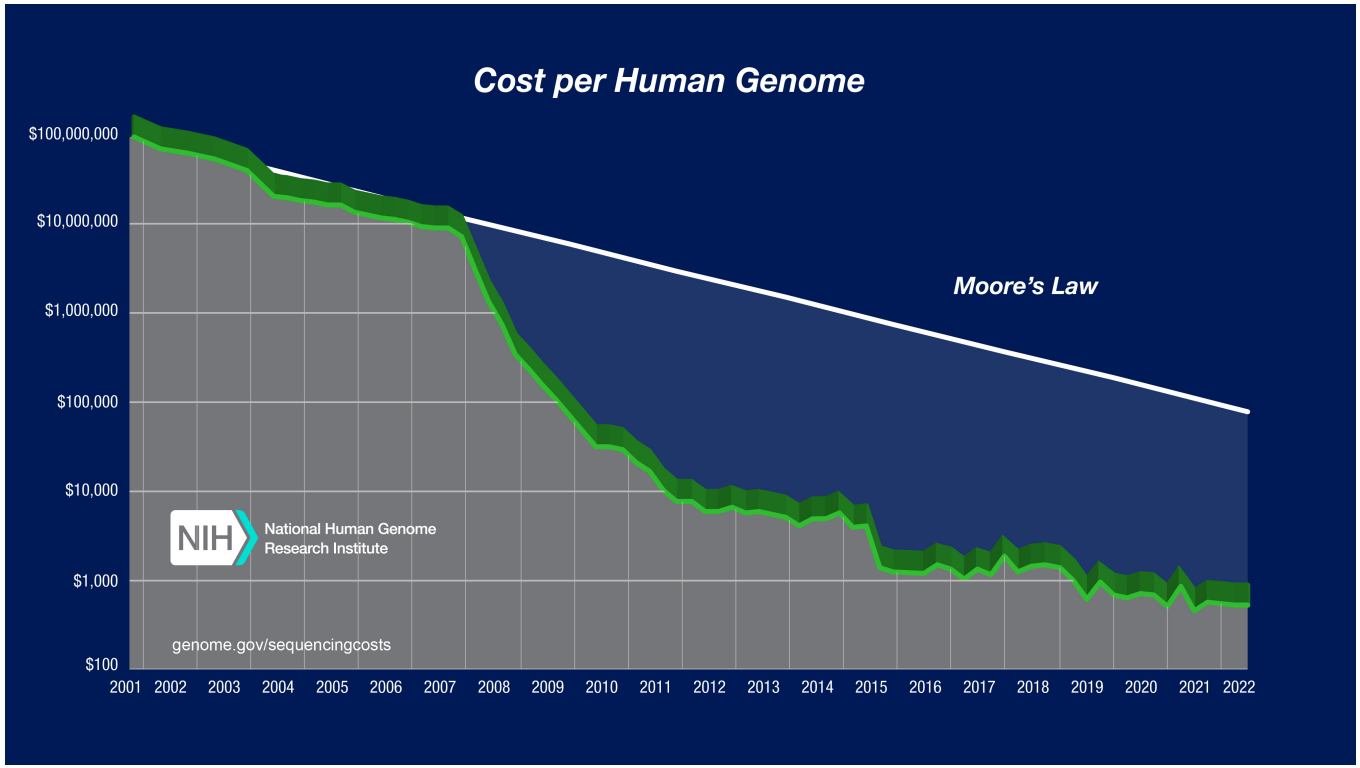
Large Genomics Models

The LLM revolution was powered by "free" internet data.
Question: Where is the abundant data in biology?

One answer: **Genomics**

Genomics produces several million terabytes of compressed data per year—a massive and growing source of structured biological information.

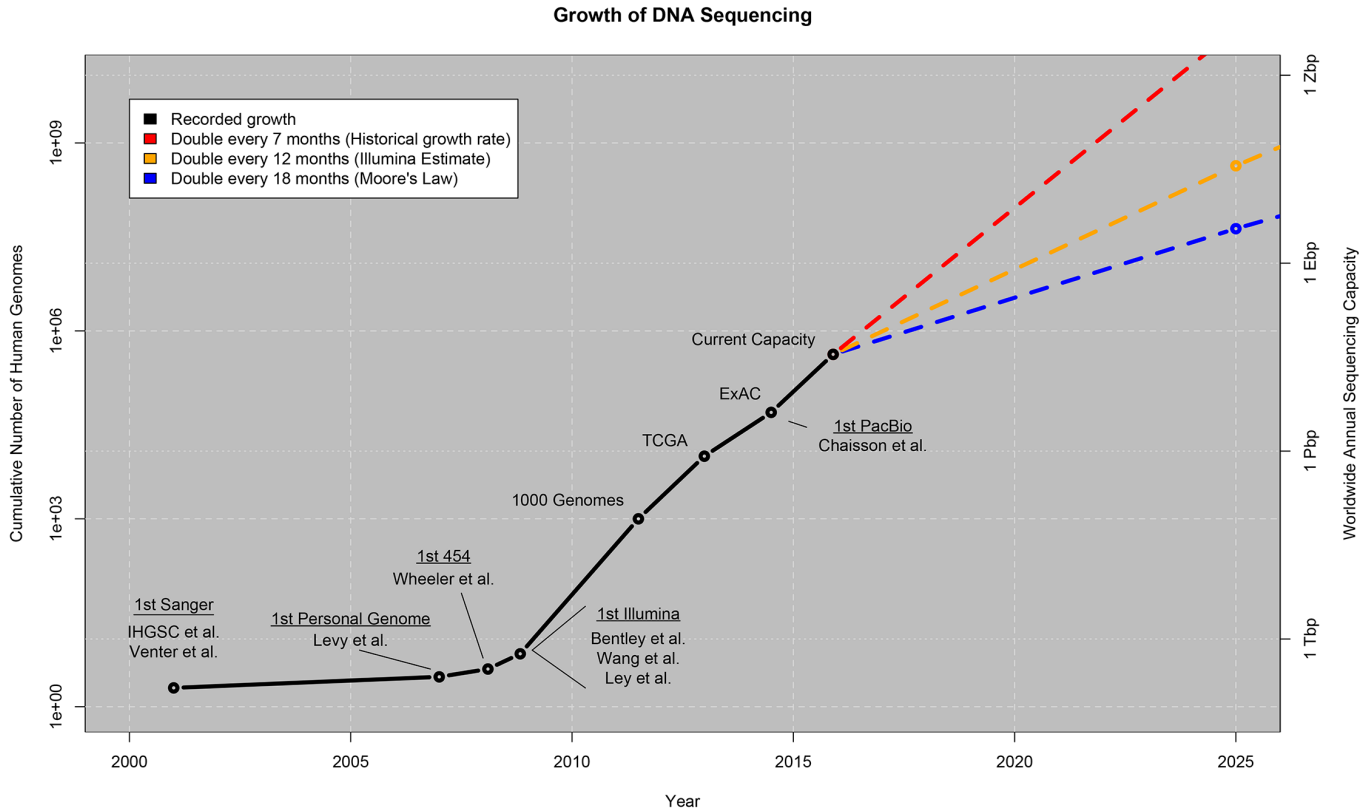
Why is Genomics Data So Abundant?



- **Cost of sequencing dropped faster than Moore's Law**
- Total reduction: **$\sim$100,000-fold in $\sim$20 years**

NHGRI, "The Cost of Sequencing a Human Genome", genome.gov (data through 2022)

The Scale of Genomics



Stephens et al., "Big Data: Astronomical or Genomical?", PLoS Biology, 2015

- 2015: worldwide sequencing capacity: >35 petabases per year
- 2025 estimate: approaching one zettabase of sequence per year by 2025

Analogy to Internet Data

- **Internet data for LLMs:** Generated as byproduct of human activity online, essentially "free"
- **Genomics data for biological models:** Generated as byproduct of medical care and research, becoming "free" at scale

Is genomics data the biology equivalent of the internet's text?

Large Genomics Models

We have established:

- LLMs are powered by massive data and scaling
- Genomics is abundant biological data

Now: What happens when we apply LLM-style approaches to genomics? Three landmark models:

- **ESM2:** Protein language model (2023)
- **Evo 2:** DNA foundation model (2025)
- **AlphaGenome:** Regulatory variant prediction (2026)

Predicting Evolutionary Correlations as Pre-Text Tasks

- Being able to predict correlations well requires a deep understanding of the process that produced them
- Correlations between distinct positions on the genome are the consequence of evolution
- Evolutionary processes are drive by the structural and functional properties and of complex macromolecular interactions across time
- **Key hypothesis:** Being able to predict correlations at distinct positions on the genome requires a deep understanding of the genome's structural and functional consequences

This is the central motivation behind large genomics models

Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model", Science, 2023

ESM2 Training – Masked Language Modeling

Architecture: Transformer **protein** model

Scale: 8 million to **15 billion parameters**

Training objective: Masked Language Modeling (MLM)

- Randomly mask **15% of amino acid positions**
- Model predicts masked amino acids from surrounding context
- "Fill in the blank" for proteins

Training data: UniRef database, ∼65 million unique sequences

Masked LM:

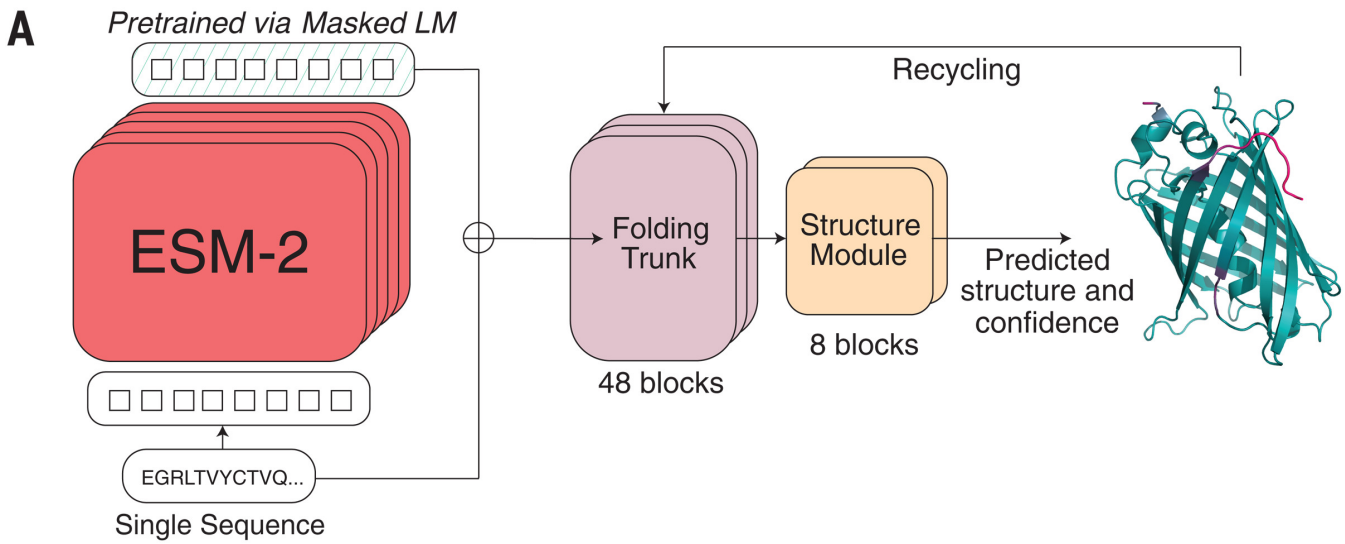
M K [?] F L [?] G

↓ predict

M K A F L V G

Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model", Science, 2023

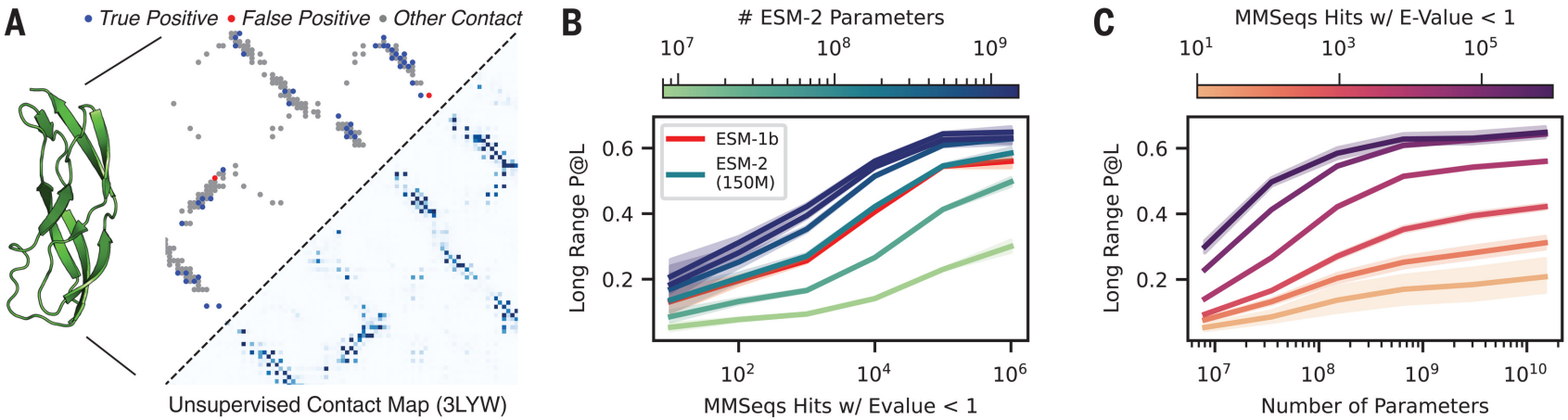
ESMFold — Replacing the MSA



Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model", Science, 2023

- Replace the MSA module with an internal ESM2 representation in an AF-like structure predictor—ESMFold

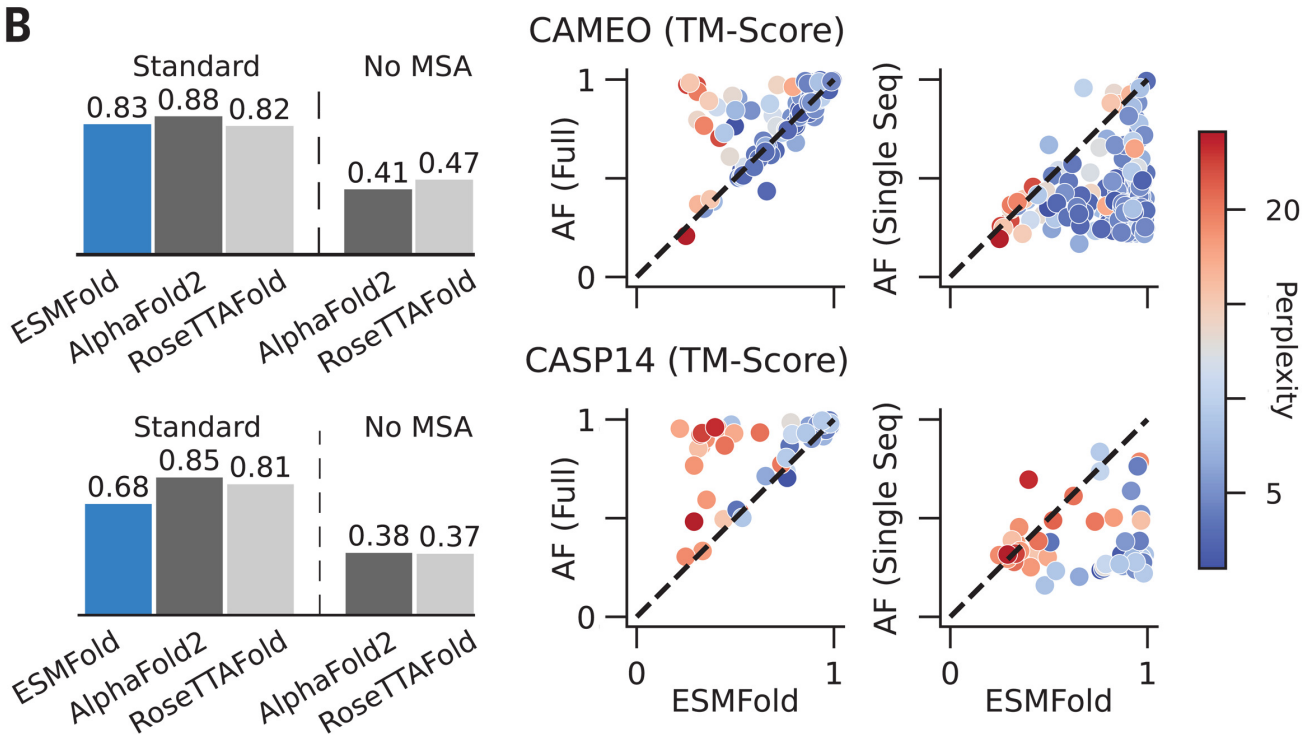
ESM2 – Emergence of Structure



Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model", Science, 2023

- Linearly probe the Transformer's attention layer to predict the residue contact map
- Measure the precision of the top L predicted contacts
- Accuracy of the predicted contacts varies as a function of the number of evolutionarily related sequences and ESM2 parameters

ESM2 – Performance



Lin et al., "Evolutionary-scale prediction of atomic-level protein structure with a language model", Science, 2023

- TM-Score is a measure of alignment (distance based) between protein structures (higher is better)
- ESMFold comparable performance to AF2 but without MSA—ESM2 learns some of the evolutionary information contained in the MSA relevant to folding

Evo 2 – The Vision

Motivation: Roughly 4 billion years of evolution has produced an extraordinary wealth of natural genomic diversity across the tree of life.

ESM2's limitation: Only protein coding sequences.

Evo 2's vision: A generalist model that can:

- Model sequences across **all domains of life**
- Handle **long-range dependencies at single-nucleotide level**
- Model both coding AND noncoding sequence

Brixi et al., "Genome modeling and design across all domains of life with Evo 2", bioRxiv, 2025

Evo 2 – Architecture Differences

Why did ESM2 model proteins? Transformers have quadratic complexity with sequence length and genomes are too long

Evo 2 used a new type of model that manages this complexity:

- State-space models (linear complexity in time)
- Sliding-window attention
- Multi-head attention for long-range dependencies

Scaled up to a context length of 1 million tokens = 1 million base pairs

Brixi et al., "Genome modeling and design across all domains of life with Evo 2", bioRxiv, 2025

Evo 2 – Training Objective Contrast

- **ESM2:** Masked Language Modeling (fill in the blank)
- **Evo 2:** Next-Token Prediction (autoregressive)
  - Predict the next nucleotide given all previous
  - Same as GPT-style LLMs!

Tokenization: Byte-level (single nucleotide)

- Each nucleotide (A, T, G, C) is a single token
- No subword tokenization

Brixi et al., "Genome modeling and design across all domains of life with Evo 2", bioRxiv, 2025

Evo 2 – Training Scale

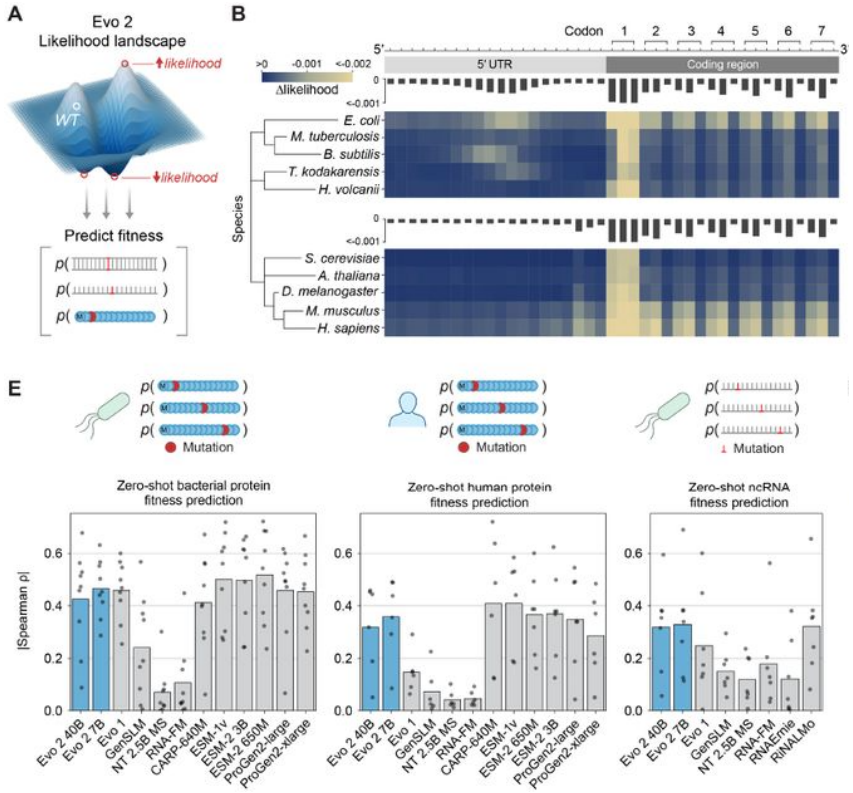
Model	Parameters	Training Tokens	Compute
Evo 2 7B	7B	2.4T	1.1×10^{23} FLOPs
Evo 2 40B	**40B**	**9.3T**	2.5×10^{24} FLOPs

Training data (OpenGenome2):

- Prokaryotes: 357 billion nucleotides from prokaryotic genomes
- Eukarya: 6.98 trillion nucleotides from eukaryotic genomes

Brixi et al., "Genome modeling and design across all domains of life with Evo 2", bioRxiv, 2025

Evo 2 – Mutation Effect Prediction



Brixi et al., bioRxiv, 2025

- Calculate the likelihood ratio (under Evo 2) of a genetic sequence with and without a mutation—zero-shot mutation effect prediction
- Correlates reasonably well with fitness in deep mutational scans—best among the DNA-level models

ESM2 vs Evo 2 Comparison

Aspect	**ESM2**	**Evo 2**
Molecule	Proteins only	DNA (all genomic)
Training	Masked LM	Next-token prediction
Context	$\sim$1,000 amino acids	**1 million nucleotides**
Scope	Protein sequences	All domains + noncoding
Key output	Structure prediction	Variant effects, generation

Analogy: "If ESM-2 is learning to read protein sentences, Evo 2 is learning to read entire books."

Large Genomics Models

- Genetic data is incredibly abundant
- It contains structural and functional information
- Training large flexible models on this data apparently allows us to extract this information

Parting thoughts

- **Data:** Story of AI in the last 5 years is a story of vast data sets
- **Models:** Emphasis is on generic "simple" architectures
- **Evaluation:** It is critical to understand how progress is evaluated to understand its value
- Data is such a critical part of the AI story that I want to quote two important articles—predating the LLM age—that foreshadow the importance of data and scale.

The Unreasonable Effectiveness of Data

> *Simple n-gram models or linear classifiers based on millions of specific features perform better than elaborate models that try to discover general rules. In many cases there appears to be a threshold of sufficient data.*

Halevy, Norvig, and Pereira. "The Unreasonable Effectiveness of Data". 2009.

The Bitter Lesson

> *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. The two methods that seem to scale arbitrarily in this way are search and learning.*

Rich Sutton. "The Bitter Lesson". 2019.