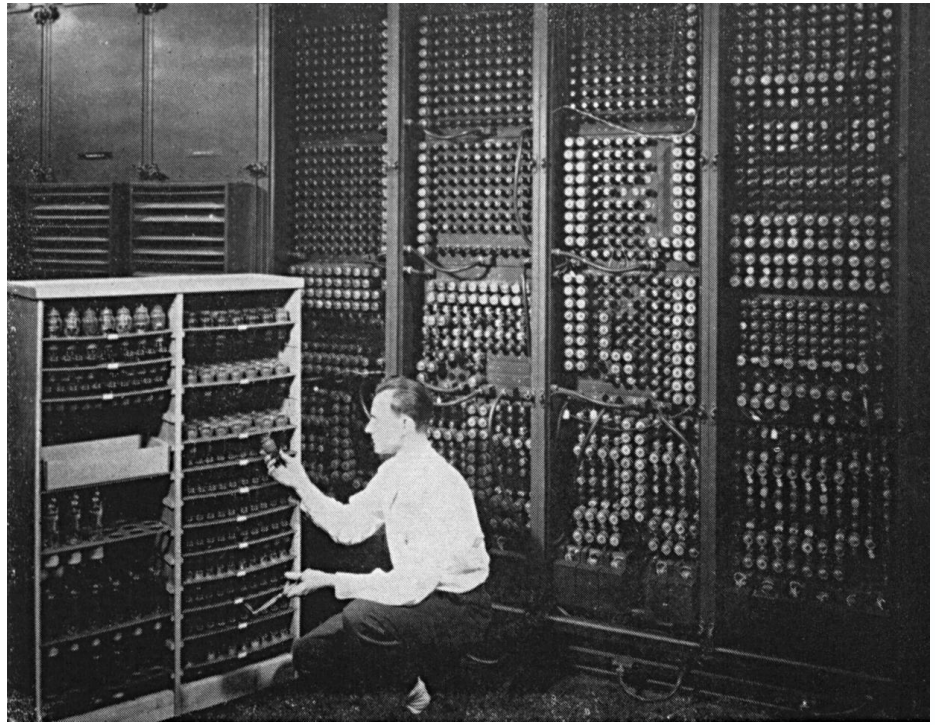
Scaling Law for Neural Language Models

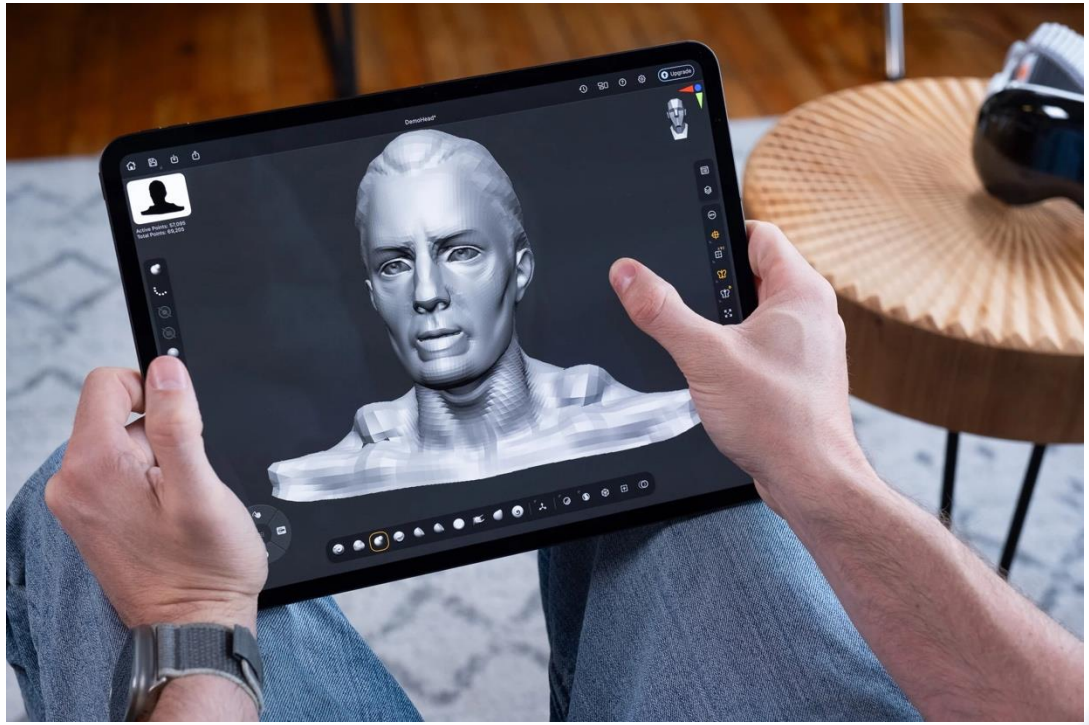
CSE 240D

Spring 2026

Hanyang Xu

From Silicon Scaling Law to Model Scaling Law





From Silicon Scaling Law to Model Scaling Law

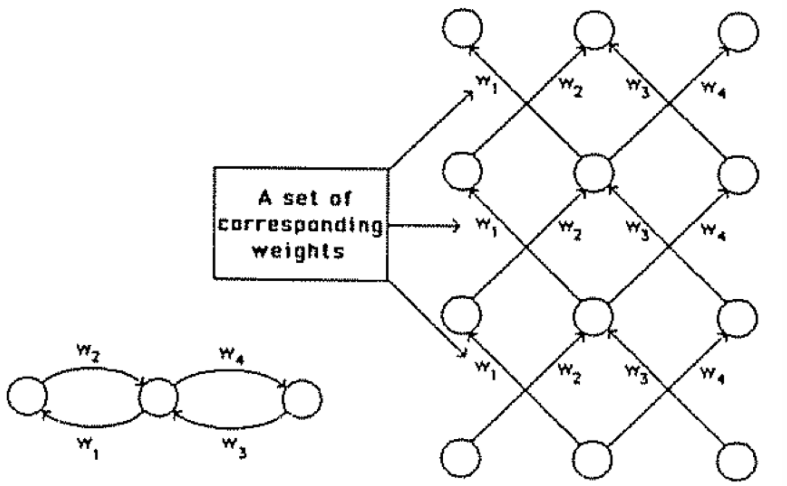


IMAGE CREDITS: NATHAN LAINE/BLOOMBERG / GETTY IMAGES

AI

Sam Altman would like to remind you that humans use a lot of energy, too

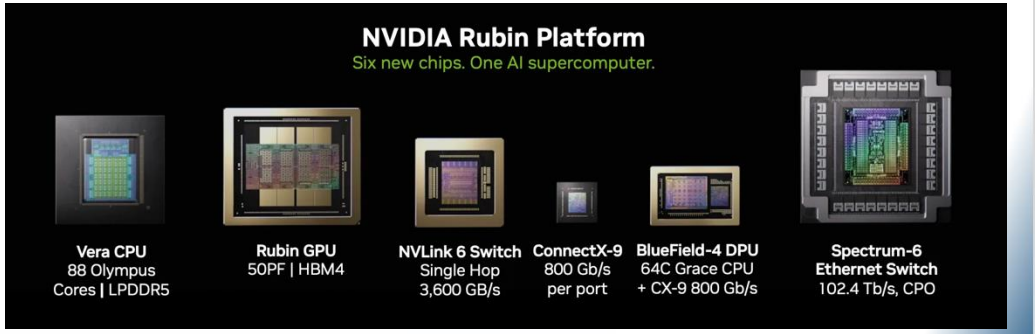
Anthony Ha — 1:38 PM PST · February 21, 2026

How do we get here?

Attention Is All You Need

Model/Model Size(N)

(D) Data



(C) Compute

What is the relation between C, D and N

Our assumption

bigger models → better performance

*This may be true, but is increasing model size the most **efficient** way of improving performance?*

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Understanding FLOPs
(floating point operations)

$$C \sim 6ND$$

C = number of FLOPs (computations)
N = number of model parameters
D = amount of training data

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Understanding FLOPs — Forward Pass

Matrix multiplication (e.g., attention QKV projection) requires **2 * size of matrix** (1 for multiplication, 1 for addition)

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1n}x_n \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2n}x_n \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mn}x_n \end{bmatrix}$$

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Understanding FLOPs — Forward Pass

N is roughly the **sum of size of all matrices**

FLOPs for **forward** pass on a **single token** is roughly **2N**

FLOPs for **forward** pass for the **entire dataset** is roughly **2ND**

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Understanding FLOPs — Backward Pass

Backward pass needs to calculate the derivative of loss with respect to **each hidden state** and for **each parameter**

FLOPs for **backward** pass is roughly **twice** of **forward** pass

FLOPs for **backward** pass for the **entire dataset** is roughly **4ND**

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Increase N $\rightarrow$ better performance

Increase D $\rightarrow$ better performance

But we have a **budget** on $\mathbf{C \sim 6ND}$

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Given a particular FLOPs (Floating Point Operation) budget, how should one trade-off model size and training data?

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

C = number of FLOPs (computations)
N = number of model parameters
D = amount of training data

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

N, D should scale at **same rate**

Approach	Coeff. a where $N_{opt} \propto C^a$	Coeff. b where $D_{opt} \propto C^b$
1. Minimum over training curves	0.50 (0.488, 0.502)	0.50 (0.501, 0.512)
2. IsoFLOP profiles	0.49 (0.462, 0.534)	0.51 (0.483, 0.529)
3. Parametric modelling of the loss	0.46 (0.454, 0.455)	0.54 (0.542, 0.543)
Kaplan et al. (2020)	0.73	0.27

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Key Question (rephrased)

*What is the **relationship** between **loss** and **N, D**?*

Irreducible Loss | Model Size Penalty | Data Size Penalty

$$\hat{L}(N, D) \triangleq \boxed{E} + \boxed{\frac{A}{N^{\alpha}}} + \boxed{\frac{B}{D^{\beta}}}$$

Credit to Anika Maskara and Simon Park

What is the relation between C, D and N

Main Results

Performance scales with **model size (N)** and **dataset size (D)**

If assuming **fixed batch size**,

D should increase by **1.7x** when **N** increases by **2x**

If assuming **optimal batch size**,

D should increase by **1.3x** when **N** increases by **2x**

Credit to Anika Maskara and Simon Park

What is the optimal batch size?

Compute-Optimal Batch Size

Critical Batch Size **dependent on the loss** (not N, D) (McCandlish et al., 2018)

(e.g., ~1M at the end of training for the best models in Experiments 1~3)

B << Critical Batch Size: **FLOP** minimized

B >> Critical Batch Size: **Training Time (i.e., step size)** minimized

B == Critical Batch Size: **Trade-off**

Credit to Anika Maskara and Simon Park

Shortcoming: Empirical Observation

- Just like the Moore's law, scaling law is ***empirical***

- **Theory work on why** analog to Amdahl's Law still on going:
  - Data Size-Model Size Scaling:

Understanding Scaling Laws with Statistical and Approximation Theory for Transformer Neural Networks on Intrinsically Low-dimensional Data – NeurIPS 2024

  - Emergence of New Capability

An exactly solvable model for emergence and scaling laws in the multitask sparse parity problem – NeurIPS 2024

What about in year 2026?

- We are running out of data to training the model -> Pre-train scaling slows
- What do we do now to make model smarter?

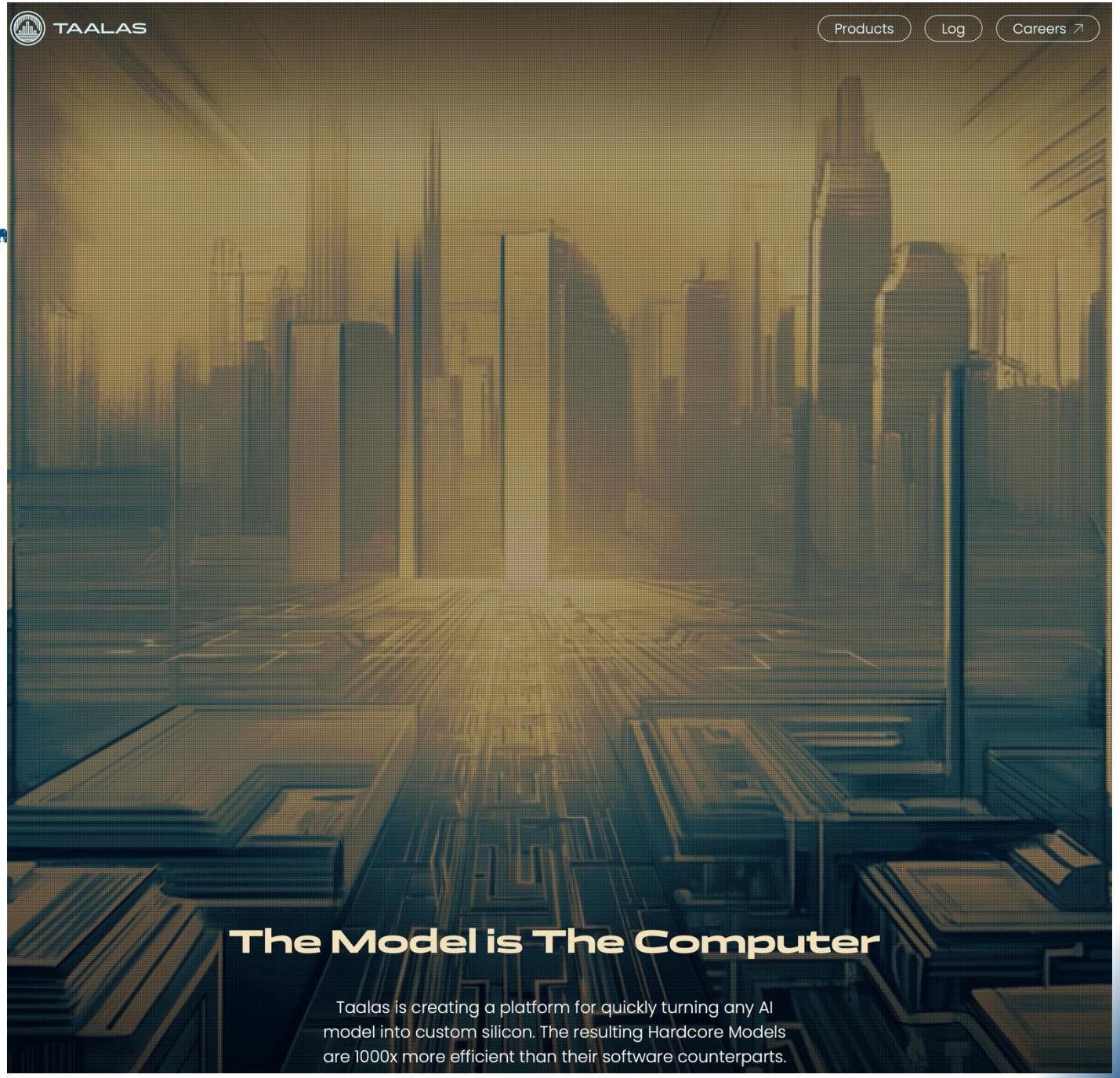
  We teach model to think, and we teach them to think longer and smarter (aka Test-Time Scaling)

  We need better data/better training/test-time scaling paradigm
  - Tool use? Claude Code is smarter than Claude chatbot on the same model
  - Expert data
  - Reinforcement Learning (Verifiable Rewards, Group Relative Policy Optimization, Large Scale Physical Simulations, etc)

  …

The End Game?

A piece of ASIC that wire in the model weight and contains all human knowledge?



Exponential Growth/Decline is Powerful

Transistor, Model Size, Data, COVID, Social Media, Population etc…