清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University

AIR 2025

Understanding the Behaviors of LLMs

A Data-Centric Approach

Jian Li 李建

Institute of Interdisciplinary Information Science

Tsinghua University

交叉信息研究院 清华大学

Understanding LLM Behaviors via Compression: Data Generation, Knowledge Acquisition and Scaling Laws.
Zhixuan Pan, Shaowen Wang, Pengfei Liao, Jian Li. NeurIPS 2025, spotlight.

Large Language Models

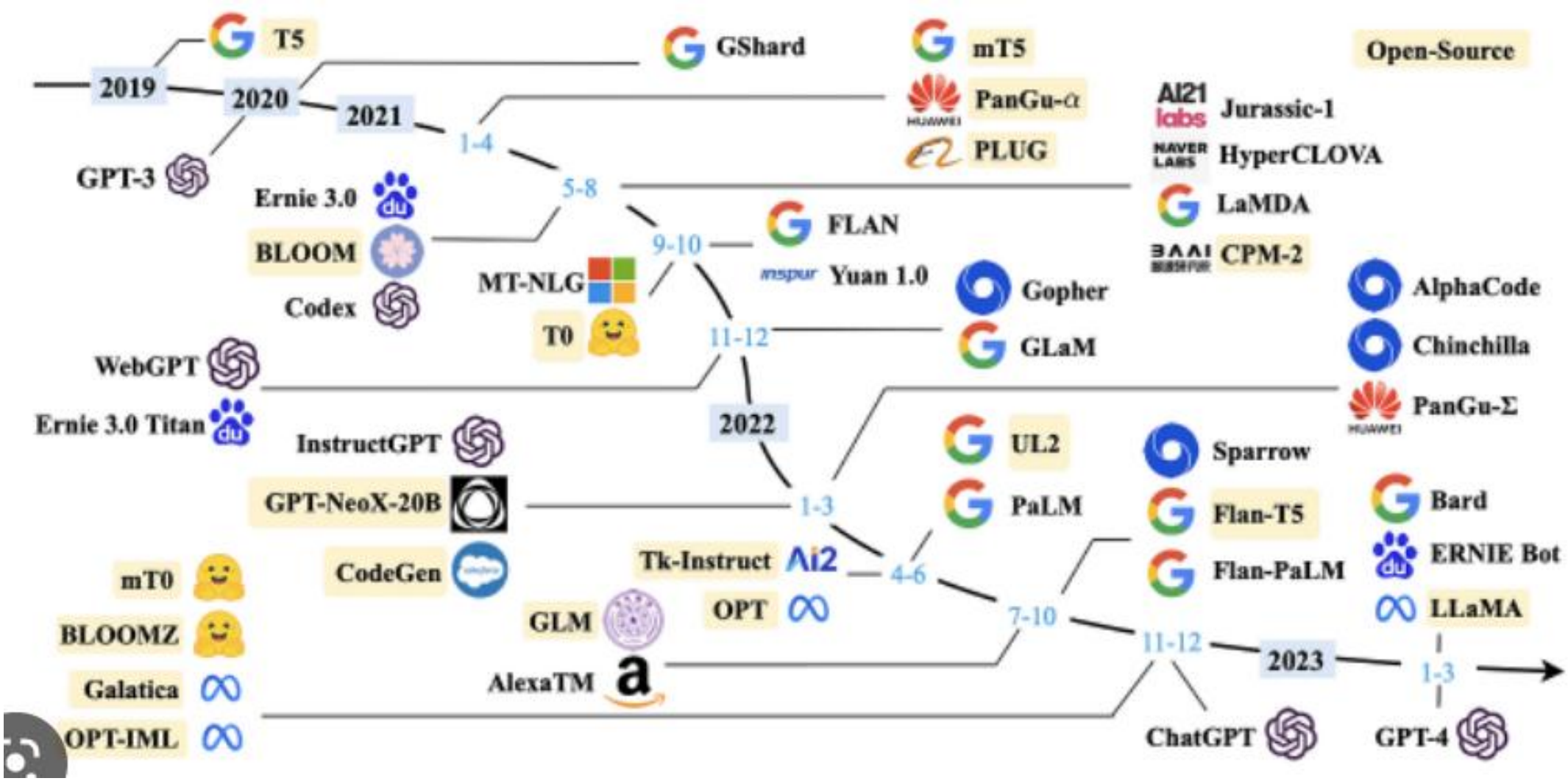
- Tremendous success in practice
- Marching towards AGI! (answering Turing's great scientific question)
- But the theory is still lagging behind

Theory of Deep Learning & LLMs

- Theory of Deep Learning
  - Optimization (new phenomena)
  - Generalization
  - Implicit bias (towards local/global min with interesting properties)
  - Understanding useful tricks: Dropout, batchnorm, layernorm, initialization
- Theory of LLM
  - Why predicting next token yields intelligence
  - Understanding Pretraining, Fine-Tuning and In-Context Learning
  - Understanding Scaling Law
  - Hallucination and Interpretability
  - Knowledge storage
  - CoT, Reasoning

Traditional Optimization and Generalization theories do NOT work any more

Pre-trained Foundation Models



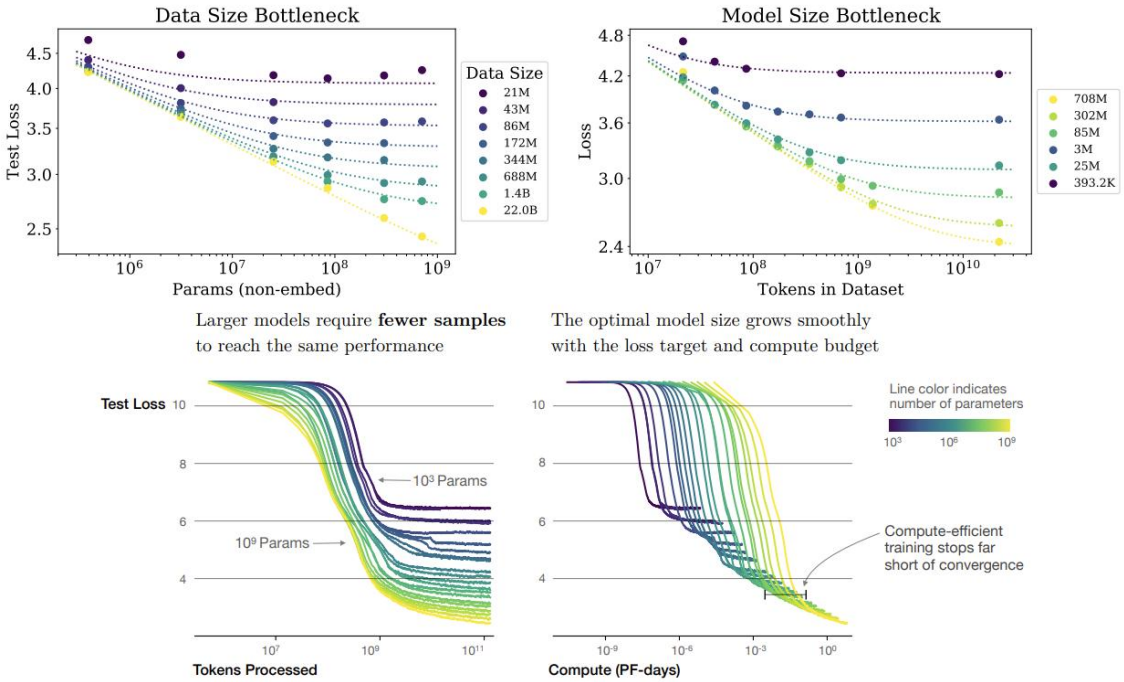
Scaling Laws

- Kaplan Scaling Law (OpenAI)

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

$\alpha_N \sim 0.076$, $N_c \sim 8.8 \times 1013$ (non-embedding parameters)

$\alpha_D \sim 0.095$, $D_c \sim 5.4 \times 1013$ (tokens)



L: the loss
N: model size;
D: dataset size (the token number of training data).

- Chinchilla Scaling Laws (DeepMind)

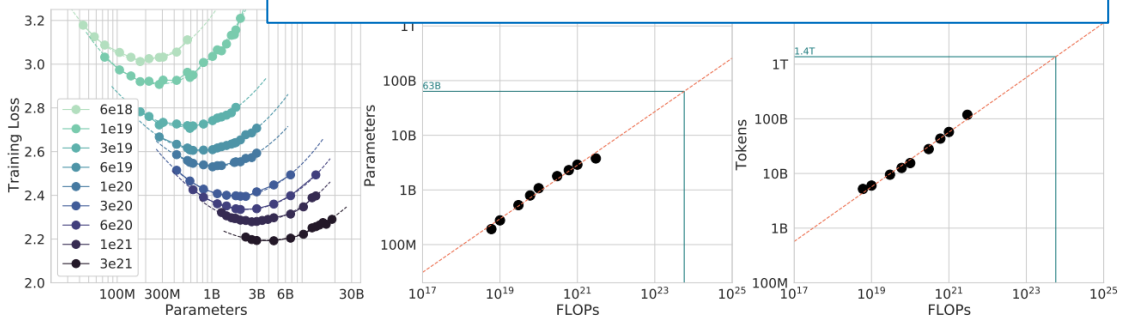
$$L(N, D) = E + \frac{A}{N^{0.34}} + \frac{B}{D^{0.28}}$$

why such strange powers appear?

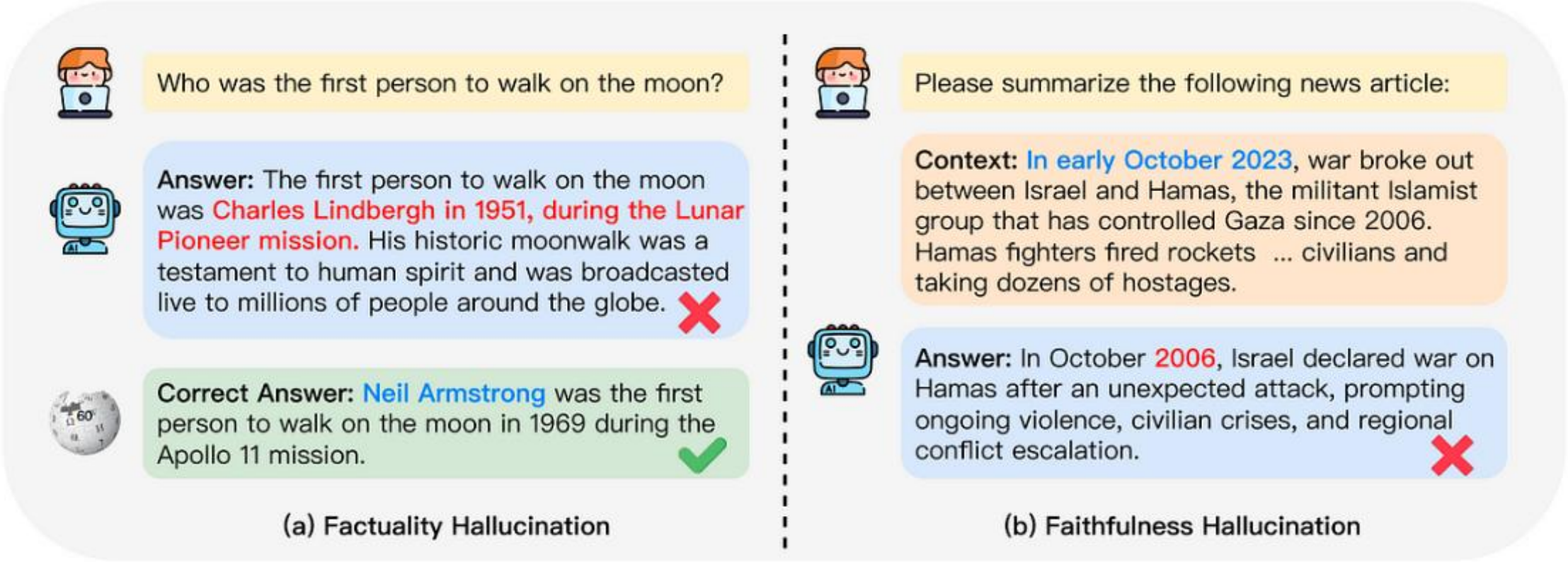
$E = 1.69$, $A = 406.4$, $B = 410$.

Typical convergence rates in ML:
$O(1/\sqrt{x})$ or $O(1/x)$.
e.g., classic VC dim bound

$$\sup_{\mu} \mathbf{E}\left[\sup_{C \in \mathcal{C}} |\mu_n(C) - \mu(C)|\right] \leq L\sqrt{\frac{\mathrm{vc}(\mathcal{C})}{n}}$$



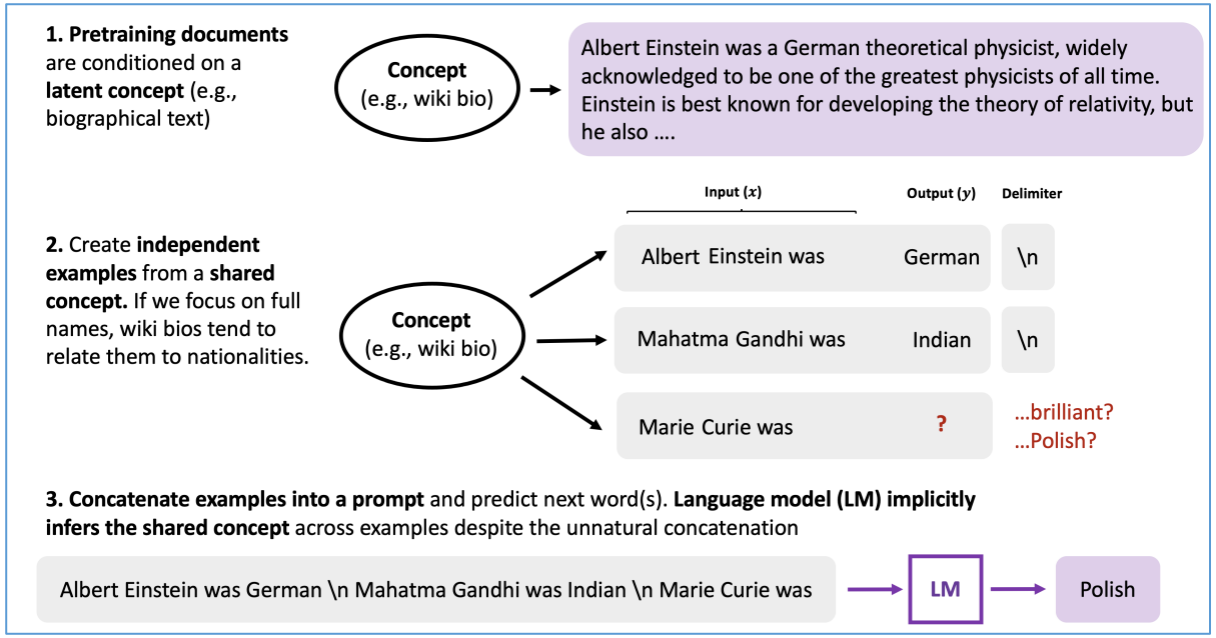
Hallucination



- Understand when and why hallucinations happen.
- Very important for safe AI

In-Context Learning

- One of the most surprising behaviors of LLMs is In-Context Learning (ICL)
- Learning to solve a new task from only a few examples from the prompt (the model's parameters does not change)
- No ML methods can learn to solve a new problem with such few example previously



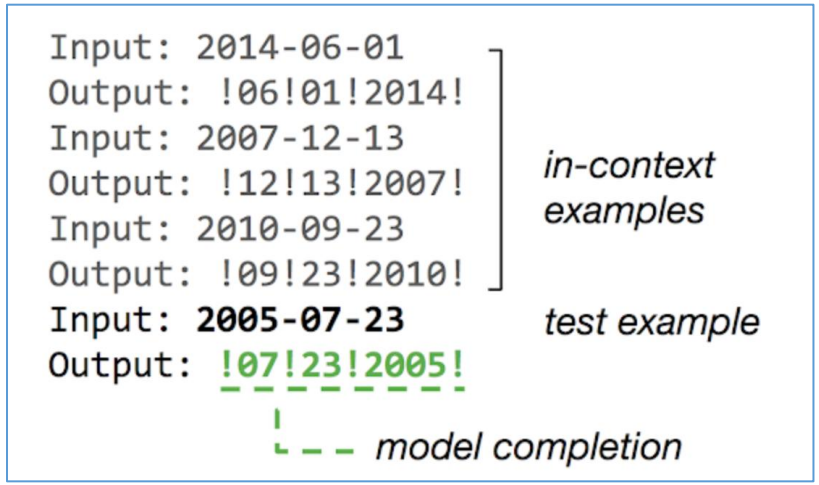
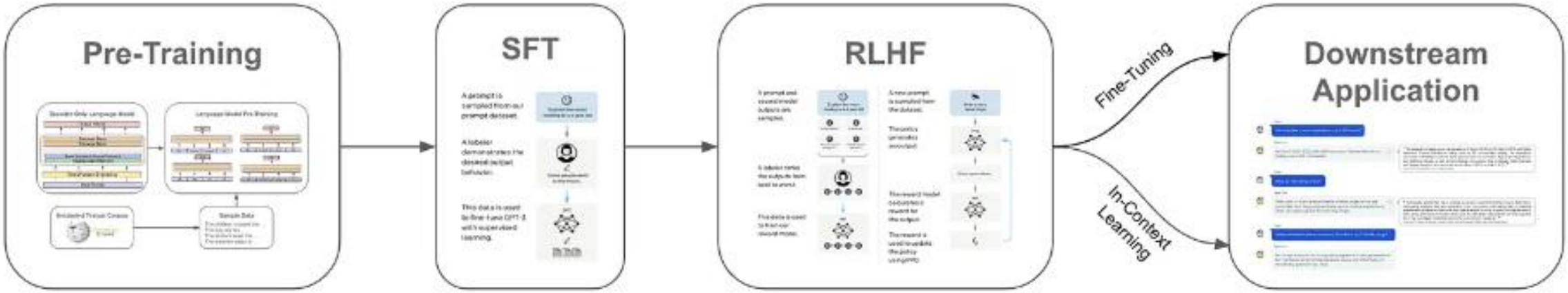


figure from [Xie et al. 2023]

Generalization in the Pretraining-Finetuning framework



- Previous learning theory paradiam
  - classic: supervised learning: i.i.d., Out-of-domain (OOD) setting, transfer learning
  - meta-learning [Jun Jeon et al. 24]

- Few result can handle such framework (generalization, sample complexity)

Related Work

Scaling Laws

- Linear regression model or random feature models [Bahri et al. 2024, Maloney et al., 2022, Spigler et al., 2020. Mei & Montanari. 2022, Bordelon et al. 2024, Lin et al. 2024]
- General linear and spiked tensor models [Mannelli et al., 2019; Mignacco et al., 2020; Mignacco & Urbani, 2022]
- Deep networks with random initialization [Bordelon & Pehlevan, 2022; Bordelon et al., 2023]
- Upper bounds of Approximation/Generalization error in Transformers [Havrilla & Liao, 2024]
- The quantization model [Hutter, 2021,Michaud et al., 2023; Brill, 2024]

In-Context Learning

- Bayesian models [Xie et al. 2021, Zhang et al. 2023, Jeon et al. 2024]
- Transformers [Garg et al. 2022, Bai et al. 2023, Zhang et al. 2023]
- Meta-optimizers [Dai et al. 2022]
- Generalization via Stability/PAC-Bayesian [Li et al. 2023, Gong et al. 2025]

Pretraining-Finetuning (Meta-Learning)

- Bayesian models [Zhang et al. 2023, Jeon et al. 2024]
- HMM models [Wei et al. 2021]
- Implicit Bias [Liu el tal. 2023]
- Representation Learning [Du et al. 2020]
- Kernel view [Malladi et al. 2023]

Outline

- A Data-Centric Approach
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

Positioning of Our Work

Achitecture

Multi-layer transformer,
Attentions, Mamba, Expressitivity,....

Data

Data filtering, Data Mixing, Data selection,
Synthetic Data generation, Tokenization,
Compression, ...

Optimizer

Adam, AdamW, Muon, Shampoo
LR schedule, landscape, ...

Post Training

Finetuning, LoRA, RLHF
CoT, Reasoning, ...

Positioning of Our Work

Achitecture
Multi-layer transformer,
Attentions, Mamba, Expressitivity,....

Data
Data filtering, Data Mixing, Data selection,
Synthetic Data generation, Tokenization,
Compression, ...

Optimizer
Adam, AdamW, Muon, Shampoo
LR schedule, landscape, ...

Post Training
Finetuning, LoRA, RLHF
CoT, Reasoning, ...

A data-centric approach

- A data generation model (to explain several behaviors of LLMs)
- Sidestep the complexity of the architecture and optimization algorithms

Positioning of Our Work

A data-centric theory

- A data generation model
- Explaining scaling laws, ICL, (certain)hallucinations, generalizations in finetuning etc.
- Scaling law achieved by **the best possible LLM**
  - Perspective from compression/universal coding
  - Bayesian setting: model and algorithm independent
  - based on information theoretic proofs
- Sidestep the complexity of the architecture and optimization algorithms
- Provide guidance for synthetic data generation

Limitation:

- Cannot say anything about the architectures and optimizers
- Cannot say anything about computability and learnability issues

Outline

- A Data-Centric Approach
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

Fundamental Ideas from Shannon

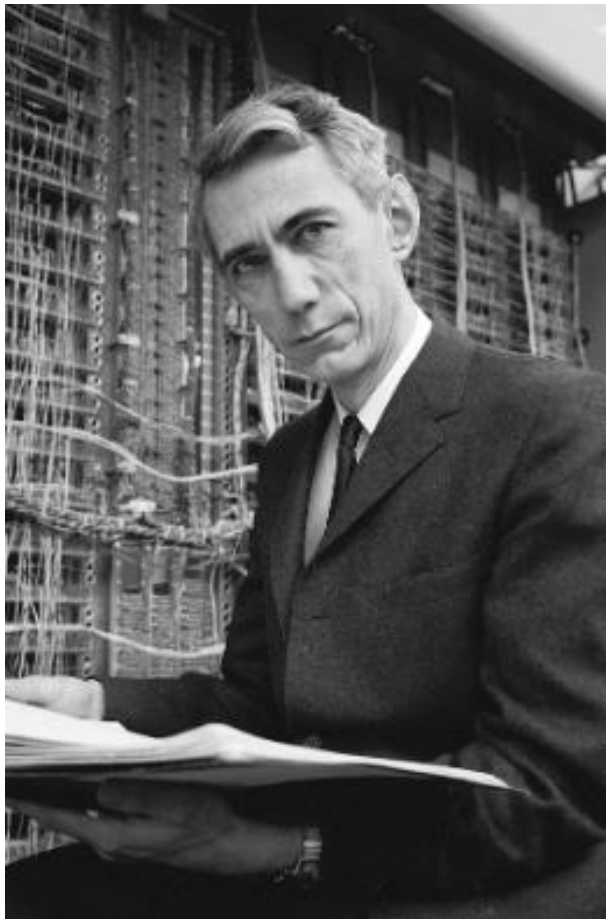
Shannon, Prediction and Entropy of Printed English (1951)

- He introduced the idea of modeling language as a stochastic process.
- Experiments to estimate language entropy (perplexity).

- "Guessing games" = ask human to predict next word

- GPT: predict next token

- Directly related to coding and compression



Lossless Compression/coding

- Compressor encoder c: $X^* \to \{0,1\}^*$
- There exists a decoder d: $R(c) \to X^*$ satisfies that $d(c(x_{1:n})) = x_{1:n}$.

The goal of lossless compression is to minimize the average code length

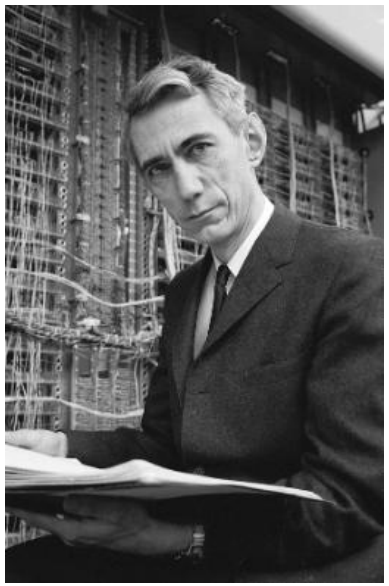
$$L_c = \mathrm{E}_{x\sim P}[l_c(x)]$$

where l_c means the bit length of $c(x)$.

Shannon's source coding theorem

$$Avg\ Code\ length\ (of\ any\ code) \geq\ entropy\ \ H(P)$$

$$H(P) = \mathrm{E}_{x\sim P}[-\log P(x)]$$



Next-Token Prediction and Cross Entropy Loss

The training task of a pre-trained large language model (LLM) is "next token prediction," so we can naturally view a pre-trained LLM as a next token predictor.

Given an unknown source distribution P, a predictor is defined as $Q_c: X^* \to \Delta^N$ which given the prefix $x_{1,\ldots,i-1}$ as input and outputs the conditional probability distribution $Q_c(x_i \mid x_{1,\ldots,i-1})$.

Cross Entropy Loss

In practice, the objective we use to train our LLMs is the cross entropy loss defined as

$$L(Q_c) = \mathbb{E}_{X\sim P}\left[\sum_i -\log \underbrace{Q_c(X_i \mid X_{1,\ldots,i-1})}_{\text{prob for next token prediction}}\right] = \mathbb{E}_{X\sim P}\left[-\log Q_c(X)\right] = H(P\|Q_c)$$

Equivalence of Prediction and Compression

The better we can predict the next token, the better we can compress the sequence:
cross-entropy loss C (per token) for predicting next token IFF compressing the sequence using C+o(1) bits (per token)

Average Code length of *predictor* Q_c (*using predictive prob* $Q_c(x)$):

$$L(Q_c) := \mathbb{E}_{X\sim P_\phi}[\ell(c(X))] = \mathbb{E}_{X\sim P_\phi}[-\log Q_c(X))] = H(P_\phi \| Q_c).$$

coding len of X

We can encode X using roughly $-\log Q_c(X)$ bits using Arithmetic coding

Cross Entropy

Redundancy of code Q_c:

$$\mathrm{Red}(Q_c, P) = L(Q_c) - H(P_\phi) = H(P_\phi \| Q_c) - H(P_\phi) = D_{\mathrm{KL}}(P_\phi \| Q_c).$$

Avg code length of Q_c

Lower bound of the code length by Shannon's coding thm

KL divergence
If Q=P then $D_{KL} = 0$

LLMs as Language Compressor

- DeLang et al. Language Modeling Is Compression. (DeepMind)

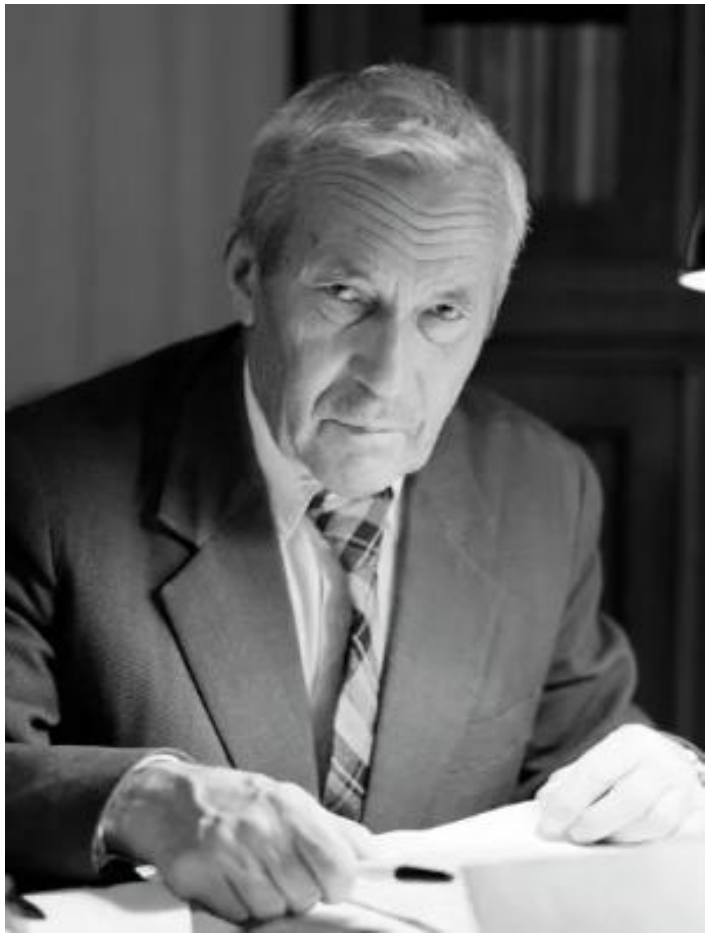
Chunk	Compressor	Raw Compression Rate (%)			
		enwik9	**ImageNet**	**LibriSpeech**	**Random**
∞	gzip	32.3	70.7	36.4	100.0
	LZMA2	23.0	57.9	29.9	100.0
	PNG	42.9	58.5	32.2	100.0
	FLAC	89.5	61.9	30.9	107.8
2048	gzip	48.1	68.6	38.5	100.1
	LZMA2	50.0	62.4	38.2	100.0
	PNG	80.6	61.7	37.6	103.2
	FLAC	88.9	60.9	30.3	107.2
	Transformer 200K	30.9	194.0	146.6	195.5
	Transformer 800K	21.7	185.1	131.1	200.1
	Transformer 3.2M	17.0	215.8	228.2	224.0
	Llama 2 (7B)	8.9	53.4	23.1	103.2
	Chinchilla 1B	11.3	62.2	24.9	108.8
	Chinchilla 7B	10.2	54.7	23.6	101.6
	Chinchilla 70B	**8.3**	**48.0**	**21.0**	100.8

- See also Li et al. 2025 Lossless data compression by large models.

Kolmogorov Complexity

Kolmogorov Complexity (algorithmic information theory)

K(X):= The minimum length of TM that outputs X.



(a fine-grained structure: Kolmogorov structure function)

- In some sense, the **ultimate notion of compression**
- Compression requires us to understand **the structure of X**
  - randomness vs regualrity
- Do not need to know the exact distribution (unlike Shannon's information theory)
- Downside: incomputable…

Examples:

- abababababababababababababababa….
- 4c1j5b2p0cv4w1x8rx2y39umgw5q85s7…
- 3141592653589793238462643383279 5…

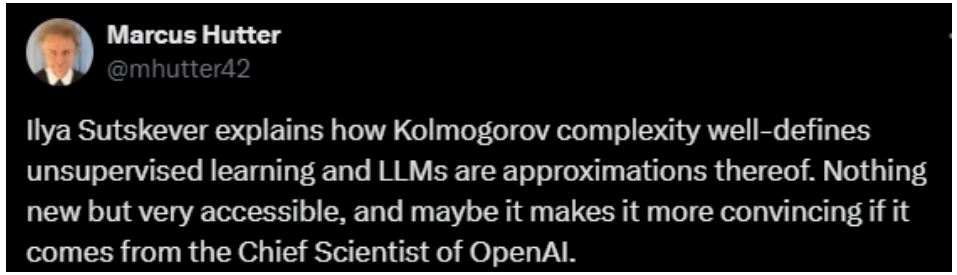
see the classic book "elements of Information Theory"

Compression leads to Intelligence

- Equivalence of **Prediction** and **Compression**
- If the model can lossless compress well, it should have learned real feature/regularities in the dataset and will generalize well.
- Occam's Razor : Entities should not be multiplied unnecessarily.
- Minimal Description Length (MDL) (Rissanen 1978): The best interpretation of a set of data is a description of that data that is accurate and as short as possible.
- Solomonoff's theory of inductive inference (1964): “If a universe is generated by an algorithm, then observation of that universe, encoded as a dataset, are best predicted by the smallest executable archive of that dataset.”

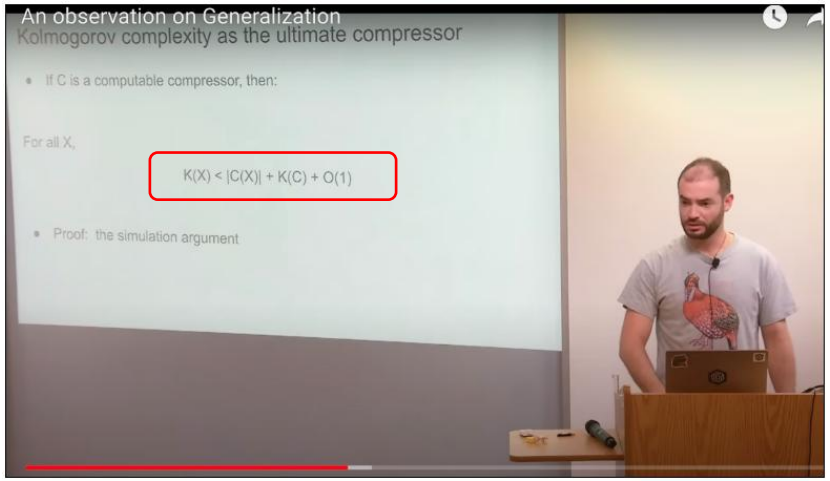
Compression and Intelligence

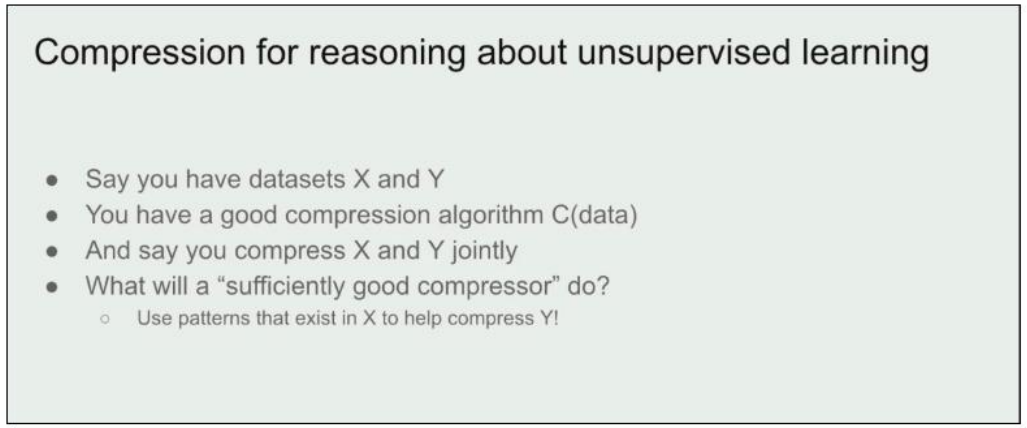
"An Observation on Generalization“ by Ilya Sutskever



Very good talk (watch on youtube : https://www.youtube.com/watch?v=AKMuA_TVz3A)

View compression from Kolmogorov complexity theory



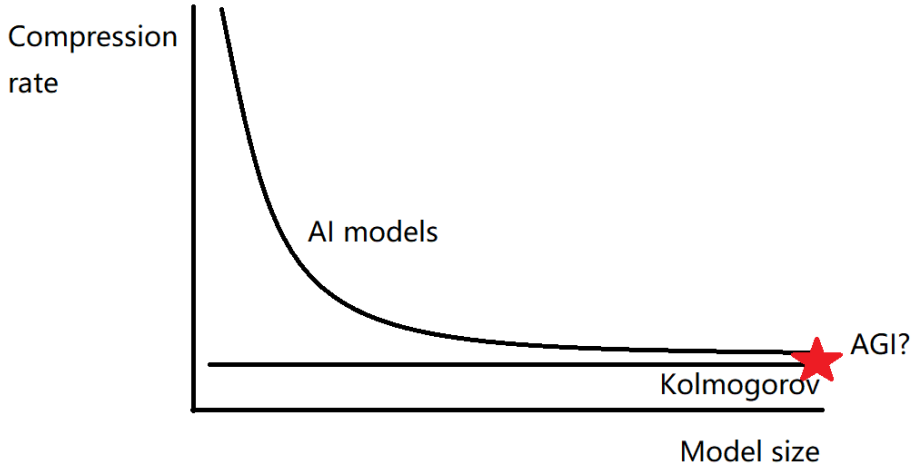


K(X): Kolmogorov complexity of string X
The minimum length of TM that outputs X.

$$K(X) < |C(X)| + K(C) + O(1)$$

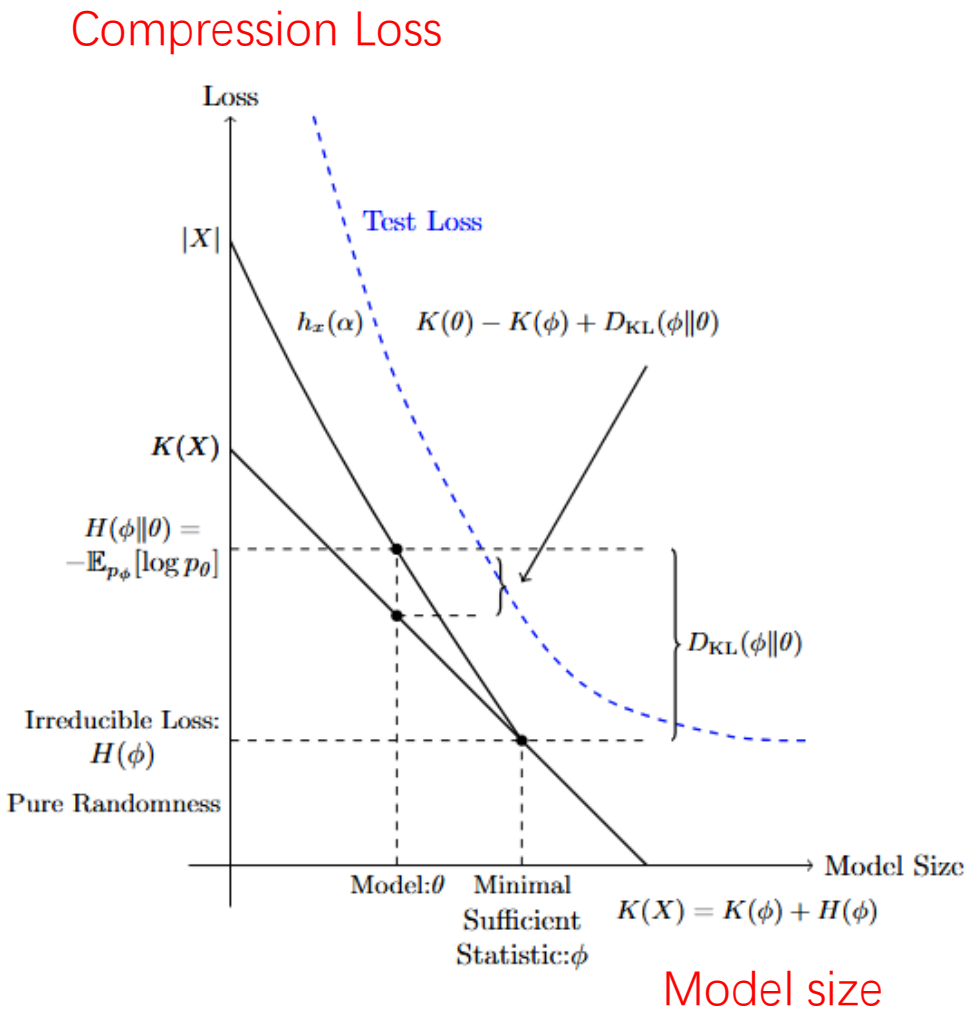
Two-part code

Kolmogorov compressor as the ultimate compressor



Why next-token prediction is enough for AGI - Ilya Sutskever (OpenAI Chief Scientist) https://www.youtube.com/watch?v=YEUclZdj_Sc

Kolmogorov structure function and LLM



- We can view LLM as a compressor
- Idealized (model) Scaling Law
  - the best compression loss we can achieve under model size constraint
- Why a power law shape (as empirical scaling laws indicate)?
- A characterization of “what should be learnt” and “what is learnt first”

Universal Predictor: Solomonoff's theory

- Dartmouth Summer Research Conference on Artificial Intelligence, where Solomonoff was one of the original 10 invitees
- A formal framework for universal inductive inference based on Kolmogorov complexity and Bayesian inference
- A **universal prior** distr $m(x)$ over all strings

$$m(x) = \sum_{p:U(p)=x*} 2^{-|p|}$$

Simpler string has a larger prior

- To predict future data, given a sequence x observed so far, and a prediction y (next token), the conditional probability is the posterior

$$m(y \mid x) = \frac{m(xy)}{m(x)}$$

- The Solomonoff predictor is **universal** $m(x) \geq c_\mu \cdot \mu(x)$.

Universal Predictor: Solomonoff's theory

- Solomonoff's predictor can be viewed as a Bayesian mixture of Turing machines

Prior:	$2^{-\|TM1\|}$		$2^{-\|TM2\|}$		$2^{-\|TM3\|}$	.....	$2^{-\|TMk\|}$ .....
	TM1		TM2		TM3	.....	TMk .....
Posterior after seeing data x	0 if TM1 cannot generate x....		0 if TM2 cannot generate x...		larger posterior prob if TM3 generates x....		smaller posterior prob if TMk also generates x....

- Hypothesis: modern LLMs is a (rough) approximation of Solomonoff's predictor.
- Drawbacks
  - Unfortunately, Solomonoff's predictor is again incomputable.
  - Solomonoff's predictor ignores the following important aspects
    - the model size constraint
    - Some (even simple) TMs are not efficiently learnable (XOR problem, one way functions etc.)

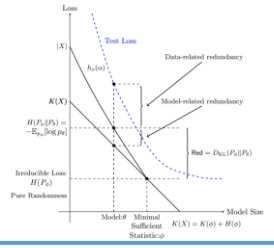
ROADMAP

Equivalence between Compression and Prediction

- minimizing cross-entropy loss ≈ shortest code for the data
- code len = entropy + redundency

Kolmogorov Complexity

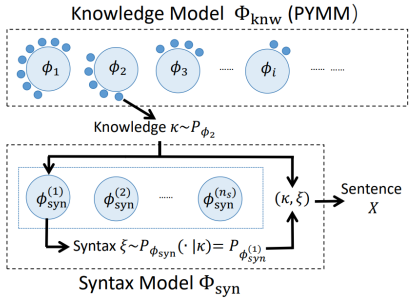
- charaterizes randomness and regularity in the data
- **Kolmogorov structure function:** idealized (model) Scaling Law
- Solomonoff's universal predictor



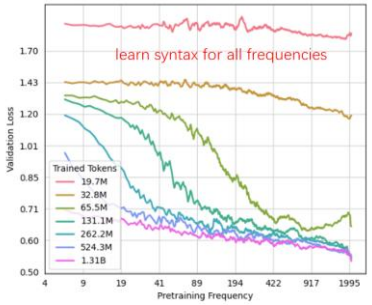
A Bayesian Data Generation Model

(Syntax-Knowledge Model)

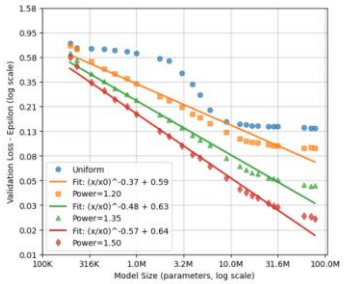
- Inspired by the insight from KSF, Heap's and Zipf's Law
- Bayesian Coding Game: Redundency = Mutual information



Data Scaling Law



Model Scaling Law



In-Context Learning

```
Input: 2014-06-01
Output: !06!01!2014!
Input: 2007-12-13
Output: !12!13!2007!
Input: 2010-09-23
Output: !09!23!2010!
Input: 2005-07-23
Output: !07!23!2005!
```

in-context examples

test example

Outline

- A Data-Centric Approach
- Fundamental Ideas from Shannon and Kolmogorov
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

Data Modeling – Syntax-Knowledge Model

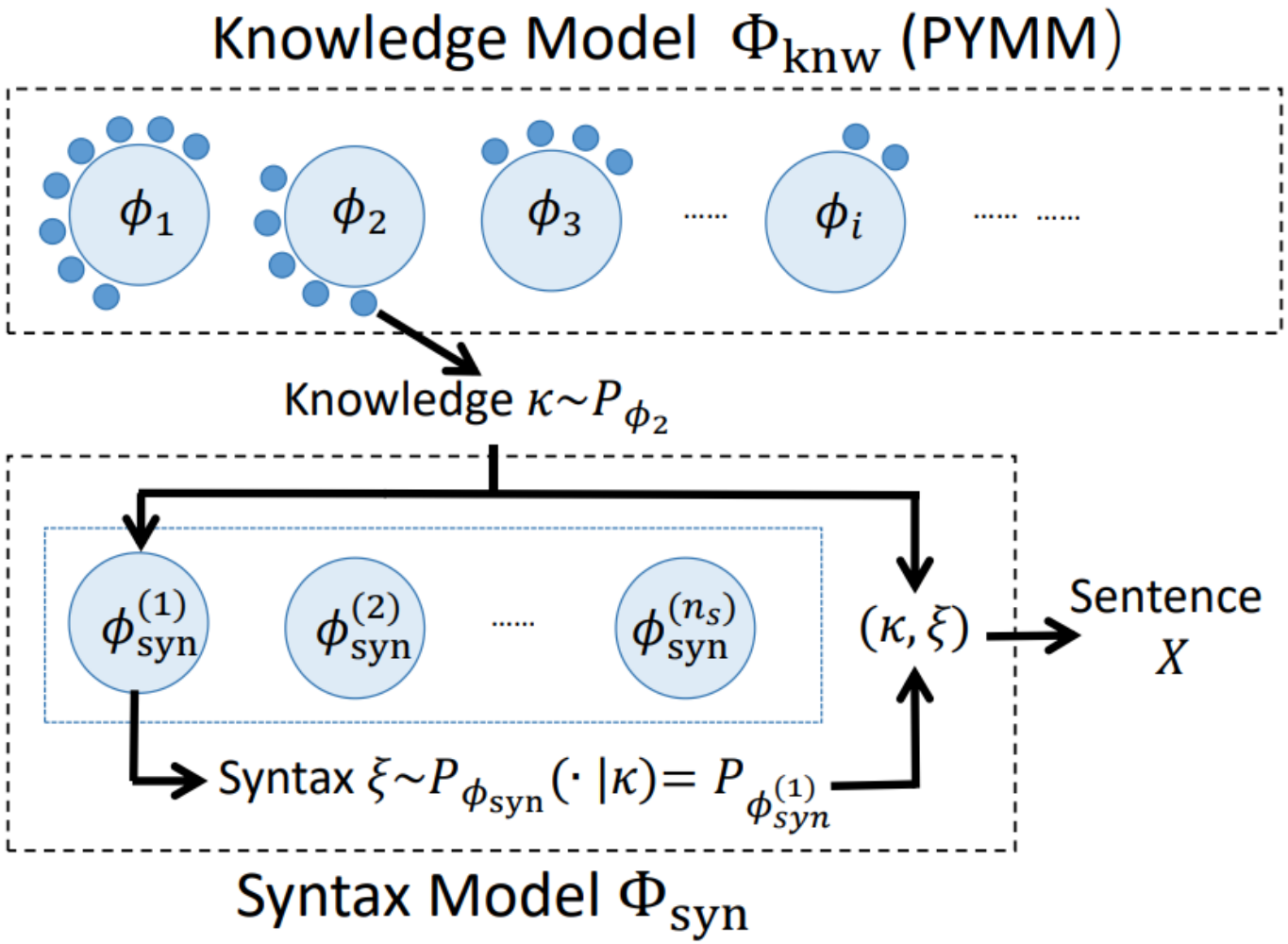
- **A Hierarchical (Nonparametric Bayesian) Model:**

Three major inspirations from compression:

(1) consider best possible compression (under model constraint) --- Kolmogorov structure function

(2) What data distribution give rise to power law structure function?

(3) Solomonoff' bayesian mixture



Data Modeling – Syntax-Knowledge Model

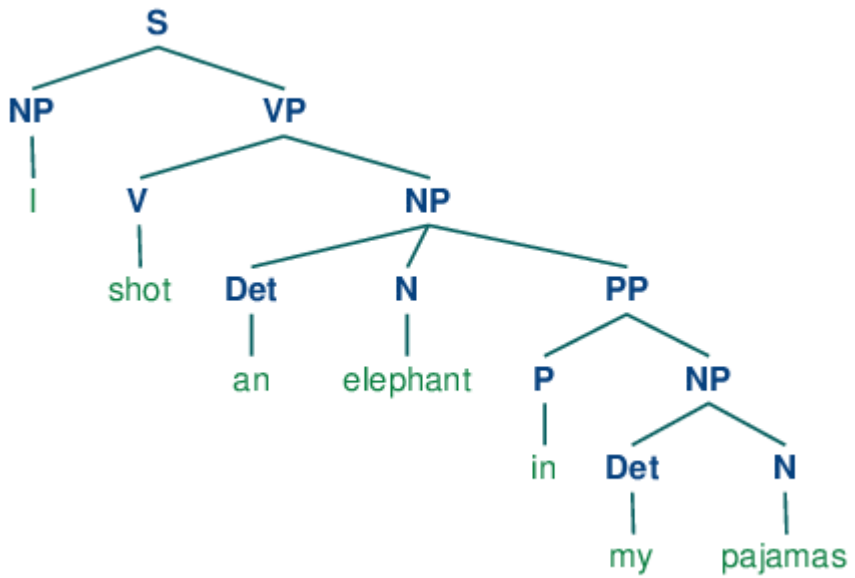
A Hierarchical Model:

- **An encoder (syntax, most common knowledge, basic logic): can be captured by a fixed-sized parameteric model that is fairly easy to learn (low sample complexity).**
- World (factual) Knowledge: a large body of knowledge, that is constantly growing (consider the number of facts, set of proteins, species, chemical substances etc.)

A syntax encoder: (probabilistic) grammar

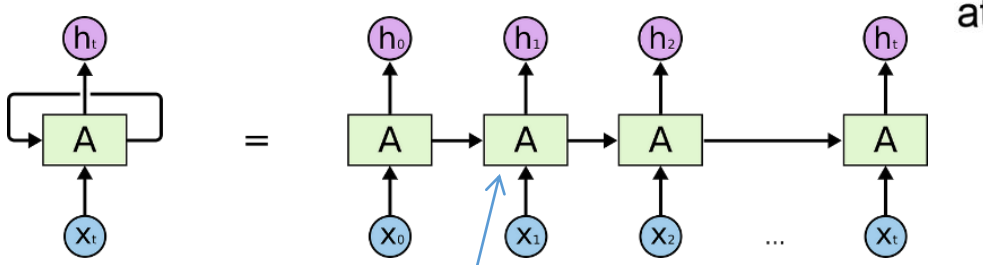
```
grammar1 = nltk.CFG.fromstring("""
  S -> NP VP
  VP -> V NP | V NP PP
  PP -> P NP
  V -> "saw" | "ate" | "walked"
  NP -> "John" | "Mary" | "Bob" | Det N | Det N PP
  Det -> "a" | "an" | "the" | "my"
  N -> "man" | "dog" | "cat" | "telescope" | "park"
  P -> "in" | "on" | "by" | "with"
  """)
```

Det the	Adj little	N bear	V saw	Det the	Adj fine	Adj fat	N trout	P in	Det the	N brook
Det the	Nom bear		V saw	Det the	Nom trout			P in	NP it	
NP He			V saw	NP it				PP there		
NP He			VP ran					PP there		
NP He			VP ran							



Syntax Encoder

- Why we model the syntax encoder using a fixed-sized parameteric model?
- Even a small RNN (fixed size) can learn language syntax fairly well. Of course a fixed-sized transformer can do it as well.



- compress the prefix to a fixed-sized state

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

World Knowledge

- An encoder (syntax, most common knowledge, basic logic): can be captured by a fair fixed parameteric model that is fairly easy to learn (low sample complexity).

- **World (factual) Knowledge: a large body of knowledge, that is constantly growing (consider the number of facts, set of proteins, species, chemical substances etc.)**

Should be consistent with **Heap's law** and **Zipf's Law**

Heap's law: empirical relationship stating that the vocabulary size grows sublinearly with the size of a corpus N

Zipf's Law: the frequency of the rth frequent word is power-law distributed

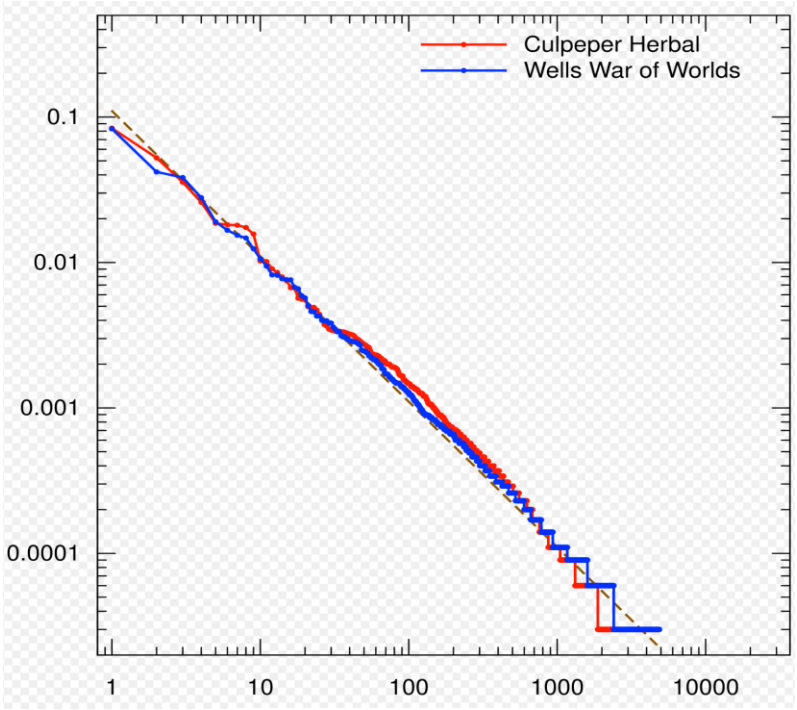
World (factual) Knowledge:

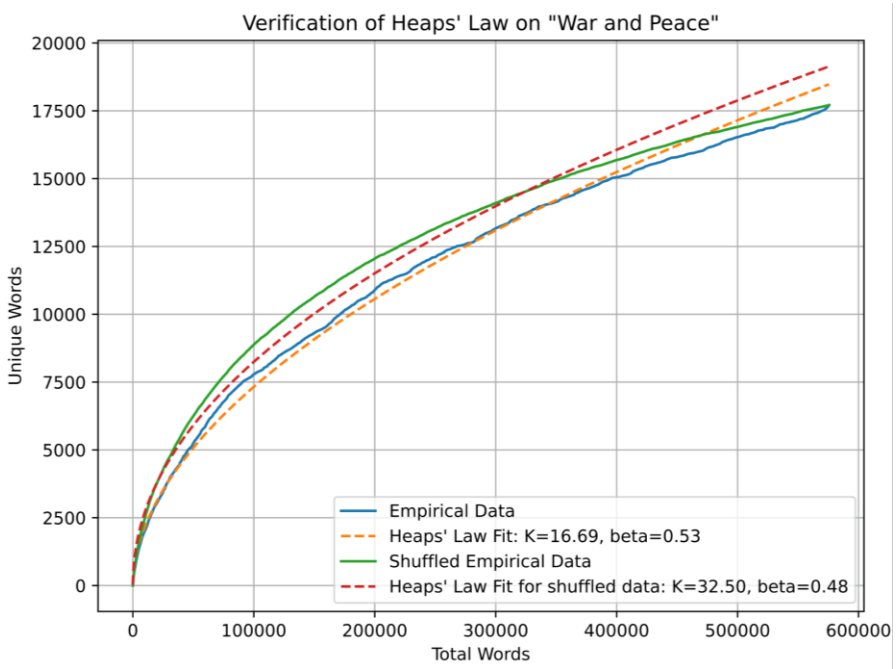
- Can't be captured by a fixed parametric model
- Modeled by **Pitman-Yor Chinese Restaurant Process (PY-CRP)**

World Knowledge: Zipf's Law and Heaps' Law

Heap's law: empirical relationship stating that the vocabulary size grows sublinearly with the size of a corpus N

Zipf's Law: the frequency of the rth frequent word is power-law distributed





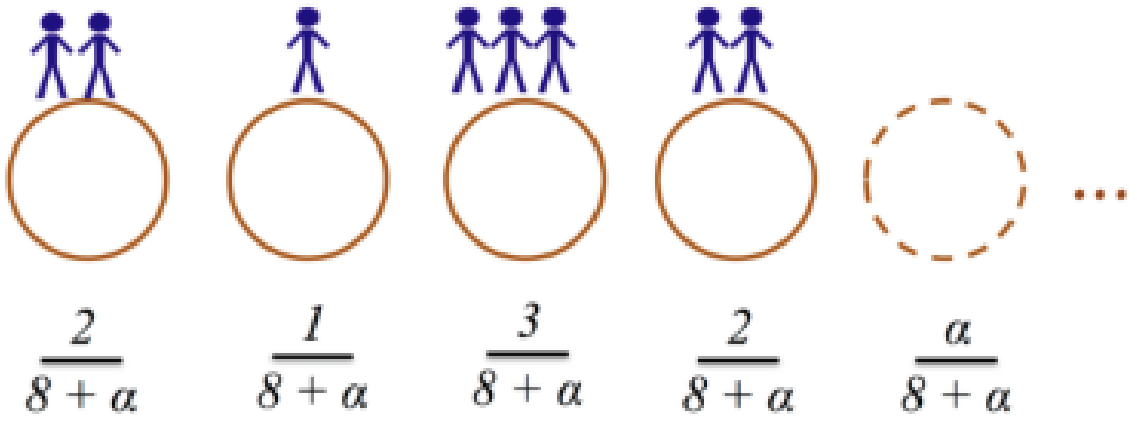
Zipf's Law: $\text{frequency} \propto \frac{1}{(\text{rank} + b)^a}$

Heaps' Law: $V_R(n) = Kn^\beta$

Pitman-Yor Chinese Restaurant Process (PY-CRP)

- A non-parametric model
- Preferential attachment

For the n-th customer:
$$\begin{cases} \frac{N_k - \alpha}{n - 1 + \beta}, & \text{if joining an existing table } k, \\ \frac{\beta + \alpha K}{n - 1 + \beta}, & \text{if starting a new table,} \end{cases}$$



- consistent with **Heap's law** and **Zipf's Law**
- leads to a power-law distribution

Lemma G.2 (Theorem 3.13 in Pitman (2006)). *Let* $p = (p_1, p_2, \dots) \sim \mathrm{PYCRP}(\alpha, \beta)$ *be the sequence of mixing weights drawn from a Pitman–Yor process. Then, the following limit almost surely exists:*

$$S_{\alpha,\beta} = \lim_{i \to \infty} i^{1/\alpha} p_i.$$

Outline

- A Data-Centric Approach
- Fundamental Ideas from Shannon and Kolmogorov
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

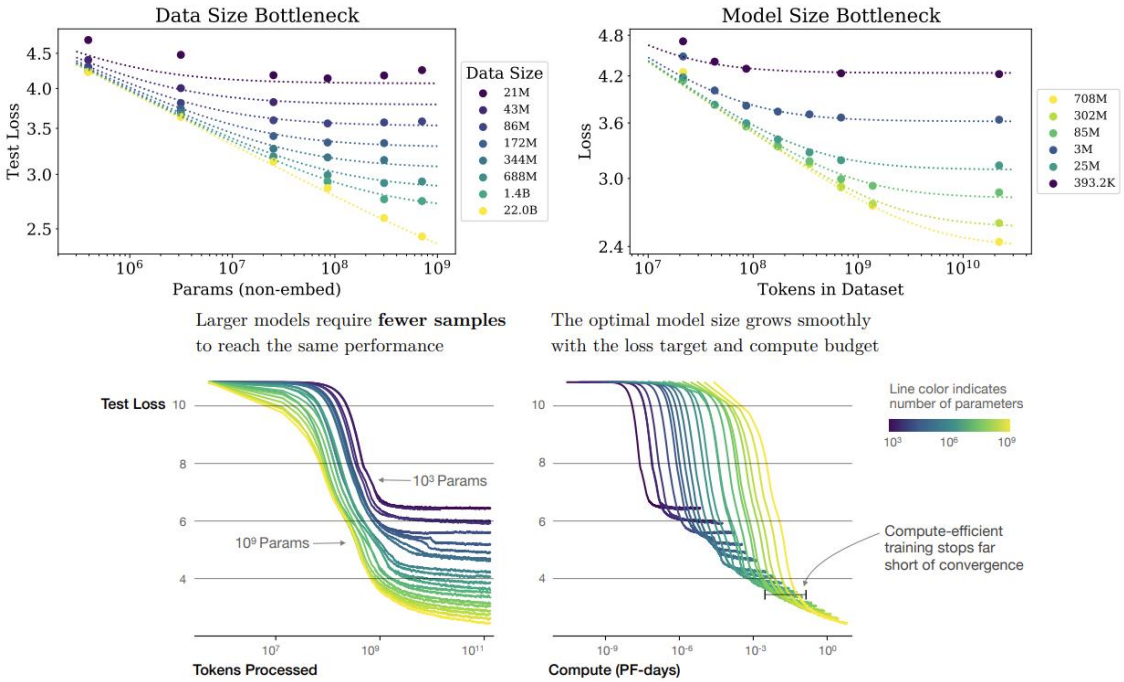
Scaling Laws

- Kaplan Scaling Law (OpenAI)

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

$\alpha_N \sim 0.076$, $N_c \sim 8.8 \times 1013$ (non-embedding parameters)

$\alpha_D \sim 0.095$, $D_c \sim 5.4 \times 1013$ (tokens)



L: the loss
N: model size;
D: dataset size (the token number of training data).

- Chinchilla Scaling Laws (DeepMind)

$$L(N, D) = E + \frac{A}{N^{0.34}} + \frac{B}{D^{0.28}},$$

$E = 1.69$, $A = 406.4$, $B = 410$.

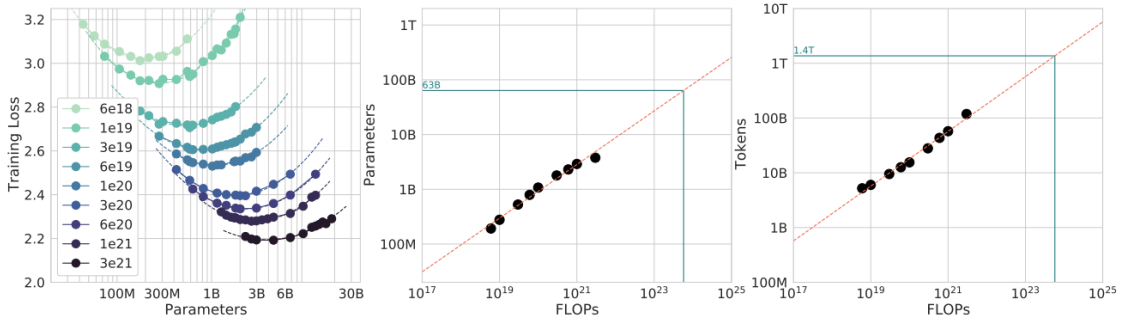


Figure 3 | **IsoFLOP curves.** For various model sizes, we choose the number of training tokens such that the final FLOPs is a constant. The cosine cycle length is set to match the target FLOP count. We find a clear valley in loss, meaning that for a given FLOP budget there is an optimal model to train (**left**). Using the location of these valleys, we project optimal model size and number of tokens for larger models (**center** and **right**). In green, we show the estimated number of parameters and tokens for an *optimal* model trained with the compute budget of *Gopher*.

A Bayesian Compression Game

- Suppose P_θ is a distribution indexed by θ.
- Bayesian setting: there is a prior π over θ
- The player would like to model P_θ using Q
- Observe x1...xn one by one
- Minimize the following Bayesian compression loss

$$\inf_Q \int_\Theta \pi(\theta) \mathbb{E}_{x_{1:n} \sim P_\theta} \left[\log \frac{1}{Q(x_{1:n})} \right] \mathrm{d}\theta = \inf_Q \int_\Theta \pi(\theta) \sum_{i=1}^{n} \mathbb{E}_{P_\theta} \left[\log \frac{1}{Q(x_i \mid x_{1:i-1})} \right] \mathrm{d}\theta.$$

minimize the (bayesian) code length (cross-entropy)

We can encode x_i using $-\log Q_c(x_i | x_{1:i-1})$ bits

Redundancy and Mutual Information

Minimizing the (Bayesian) Redundancy:

$$\inf_Q \int_\Theta \pi(\theta)\mathbb{E}_{x_{1:n}\sim P_\theta}\left[\log\frac{1}{Q^n(x_{1:n})}) - \log\frac{1}{P_\theta^n(x_{1:n})})\right]\mathrm{d}\theta$$

$$= \inf_Q \int_\Theta \pi(\theta) D_{\mathsf{KL}}(P_\theta^n \| Q^n)\mathrm{d}\theta \;=\; \inf_Q \int_\Theta \pi(\theta)\mathsf{Red}(Q^n, P_\theta^n)\mathrm{d}\theta \;\triangleq\; \inf_Q \mathsf{Red}_n(Q,\Theta)$$

Connection of coding redundancy and mutual information (Clarke & Barron, 1994; Duchi, 2024; Jeon & Van Roy, 2024)

Lemma A.3. *The minimum Bayesian redundancy is attained by the Bayesian mixture code Q_π, and is equal to the mutual information between random variable θ (from the prior π over Θ) and the data $x_{1:n}$.*

$$\inf_Q \mathsf{Red}_n(Q,\Theta) = \inf_Q \int_\Theta \pi(\theta) D_{\mathsf{KL}}(P_\theta^n \| Q^n)\mathrm{d}\theta = \int_\Theta \pi(\theta) D_{\mathsf{KL}}(P_\theta^n \| Q_\pi^n)\mathrm{d}\theta = I(x_{1:n};\theta).$$

Here, $\theta \in \Theta$ is sampled from the prior π, and $x_{1:n}$ are sampled from P_θ^n.

mutual information between the data and the prior

Understanding Scaling Law

The Bayesian loss (Avg code len) can be decomposed into three parts.

Theorem (informal): Under the above data generative model and unlimited model size, we can prove that the **Bayesian predictor** (the optimal predictor) has the following loss

Bayesian Redundancy (bounding Mutual Info)

$$\inf_M \frac{1}{N}\mathbb{E}_{\phi_{data}\sim\pi, X_{1:N}\sim P_{\phi_{data}}}\left[-\log P_M(X_{1:N})\right] = \tilde{O}\left(\frac{d_{knw}}{N^{1-\alpha}} + \frac{n_s d_{syn}}{N}\right) + \frac{1}{N}H(X_{1:N}|\mathbf{\Phi}_{data}).$$

Bayesian loss
Avg code length

Loss incurred by learning the knowledge (power law)

Loss incurred by learning the syntax encoder

Entropy
Irreducible loss (pure randomness)
cf. Shannon Sourcing coding theorem

Takeaway: the power law for knowledge part is mainly due to the power law distribution of the data

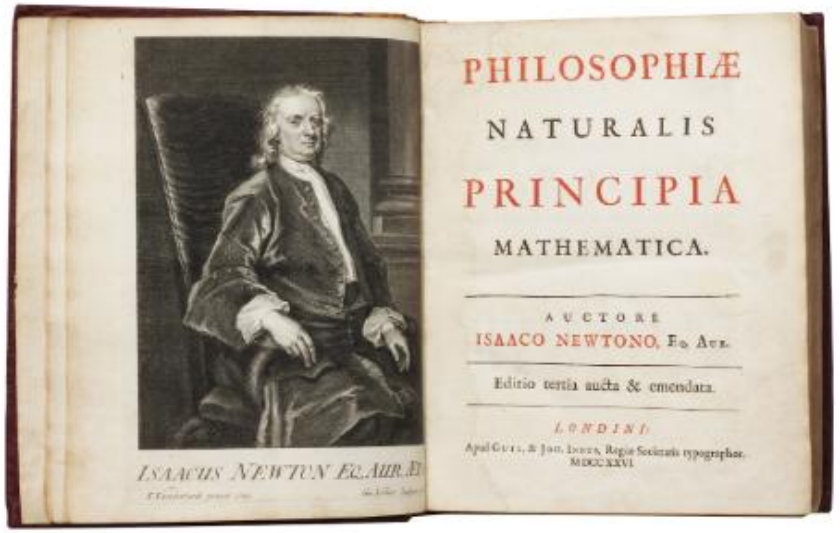
- Similar insight in Hutter 2021; Michaud et al., 2023; Brill, 2024
- A related result (dynamic analysis): Disentangling feature structure: A mathematically provable two-stage training dynamics in transformers. Gong, Li, Liu, Teng

Controlled experiments

Methodology popularized by Allen-Zhu and Li in a series of papers **“Physics of LLMs” -1,2,3**

Ethological approach

Controlled experiments
“LLM monkeys”

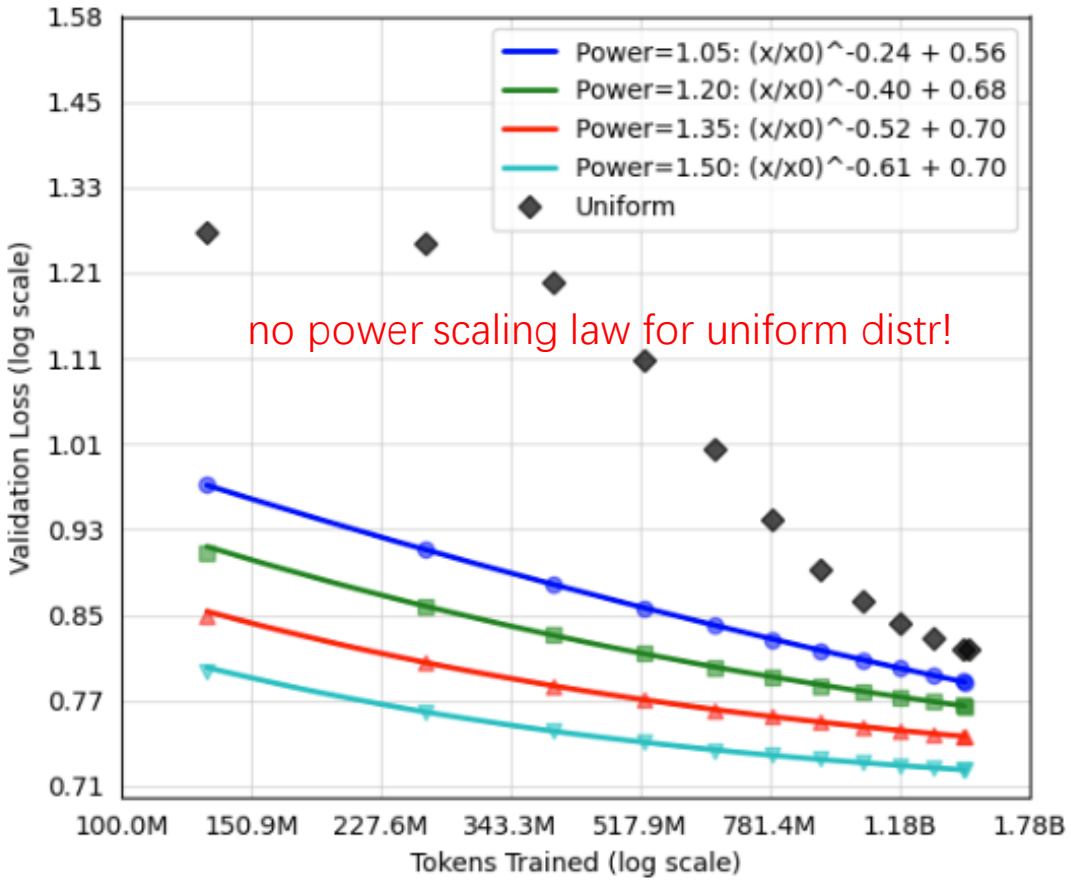


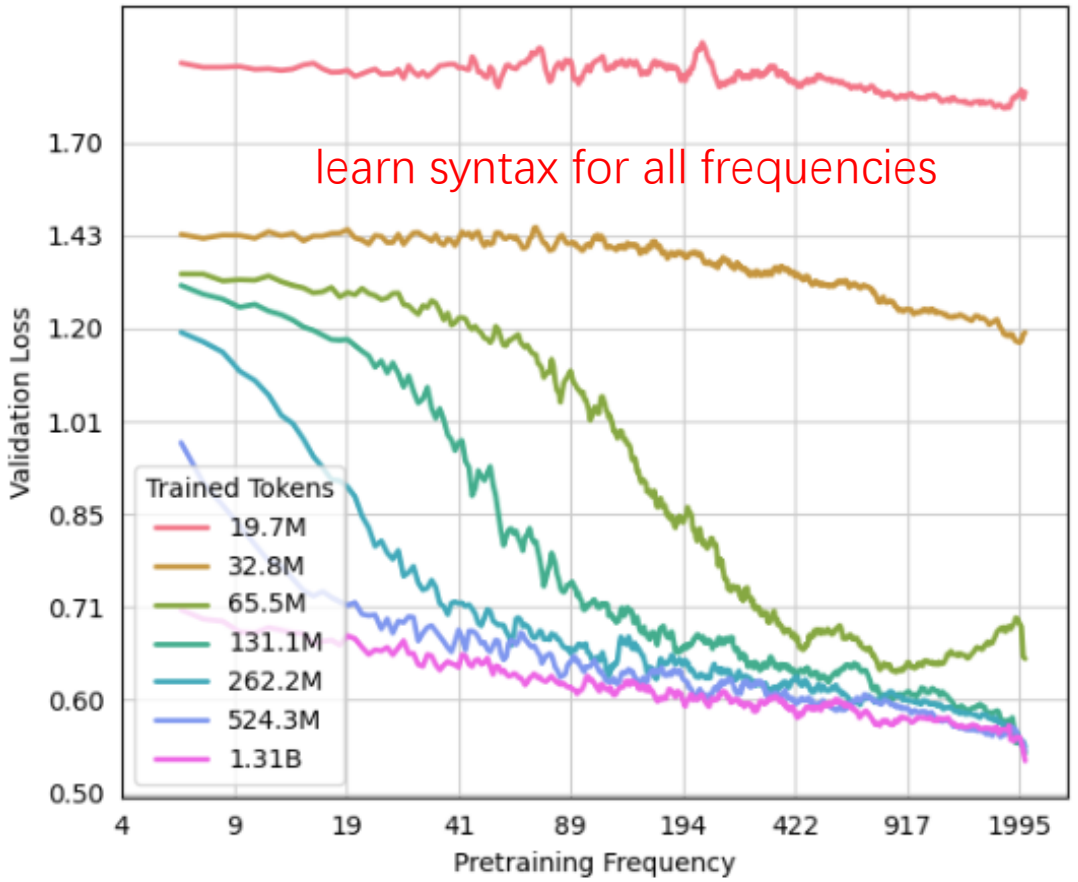
Physics

bioS dataset: we generate profiles for 400,000 individuals. Each profile contains six attributes: date of birth, birth city, university, major, employer, and employer city

e.g., “Gracie Tessa Howell wasborn in Camden, NJ. He studied Biomedical Engineering and worked at UnitedHealth Group. He entered the world on April 15, 2081, and is employed in Min-netonka. He is an alumnus/alumna of Buena Vista College.”

Data Scaling Law





Syntax learnt first, then higher frequency knowledge, to rarer knowledge

Outline

- A Data-Centric Approach
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

Model Scaling Law

Redundancy for the ith knowledge cluster:

$$\mathcal{D}_i(R) = \min_{\mathbb{I}(\phi_i;M_C^*)\leq R} \mathbb{E}_{\phi_i\sim\pi_{\text{knw}}}\left[\mathbb{E}_{q\sim P_{\phi_i}}\left[D_{\text{KL}}\left(P_{\phi_i}(A\mid q)\,\|\,P_{M_C^*}(A\mid q)\right)\right]\right]$$

Model scaling law (total redundancy) can be captured by the following optimization problem

$$\text{minimize} \quad \mathbb{E}_i[\mathcal{D}_i(m_i)] = \sum_{i=1}^{\infty} p_i \mathcal{D}_i(m_i), \qquad \text{minimizing redundency}$$

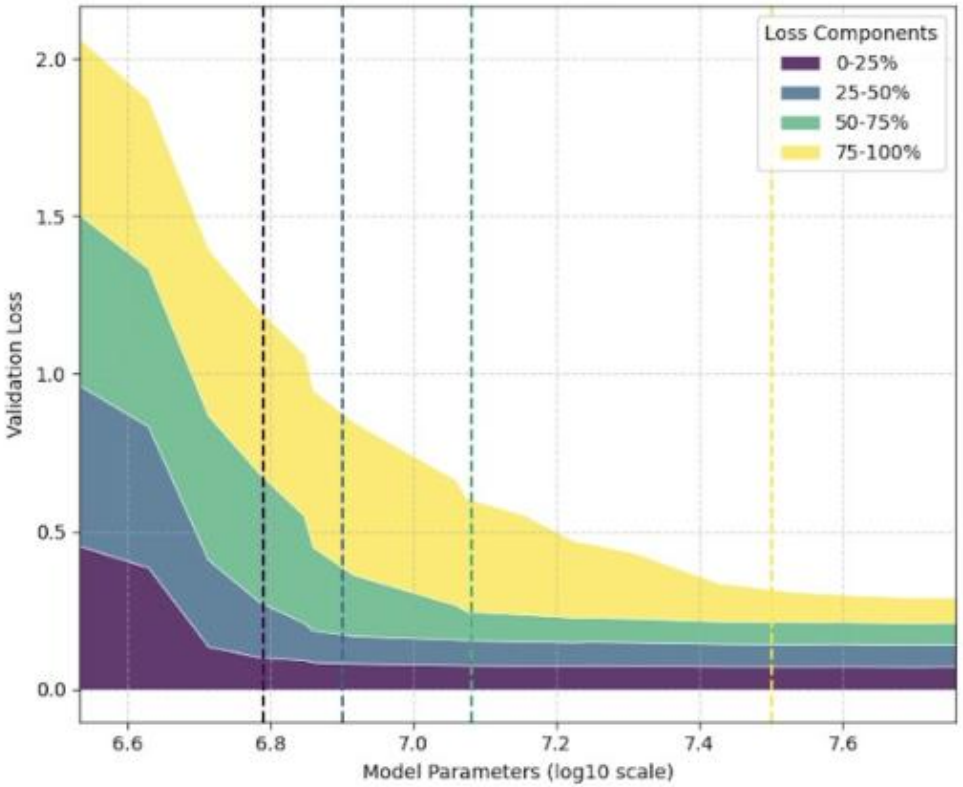
$$\text{subject to} \quad \mathbb{I}(\Phi_0; \boldsymbol{M}_C^*) \leq C, \quad m_i = \mathbb{I}(\phi_i; \boldsymbol{M}_C^*) \geq 0 \text{ for all } i \in \mathbb{N}^+.$$

subject to total model capacity constraints

the optimization problem can be solved using KKT condition

Model Scaling Law

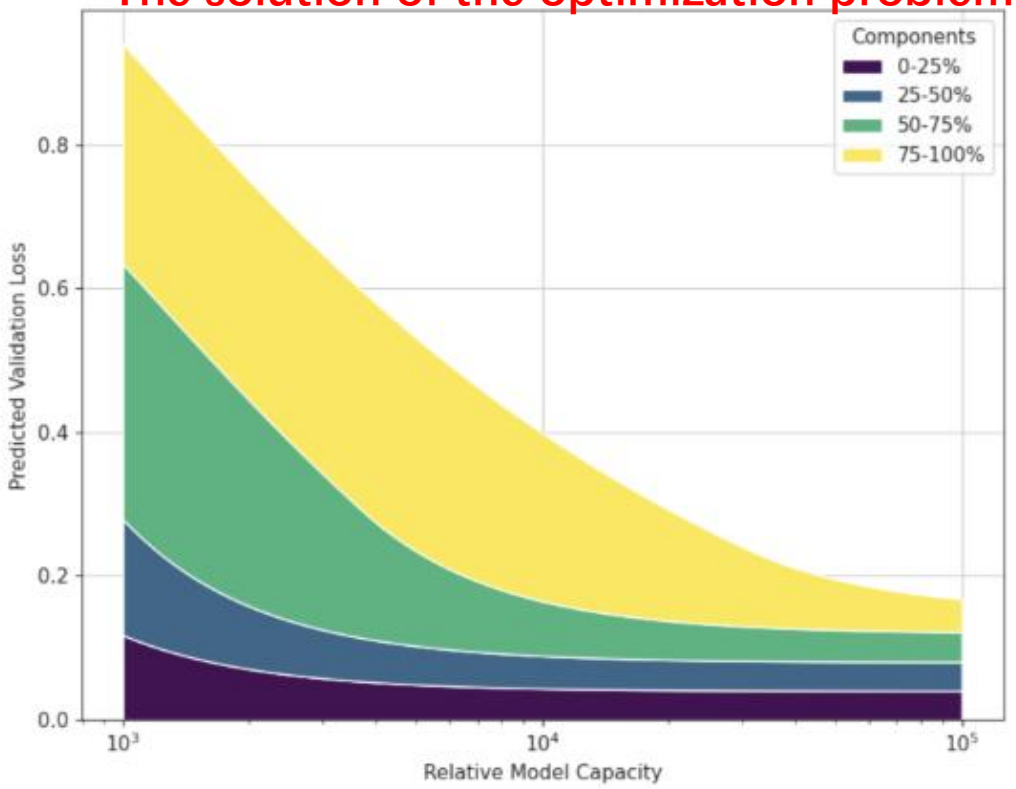
Under the same data model, we have a theory of model scaling law (inspired by Kolmogorov structure function)



Empirical results

color corresponds to data with different frequency

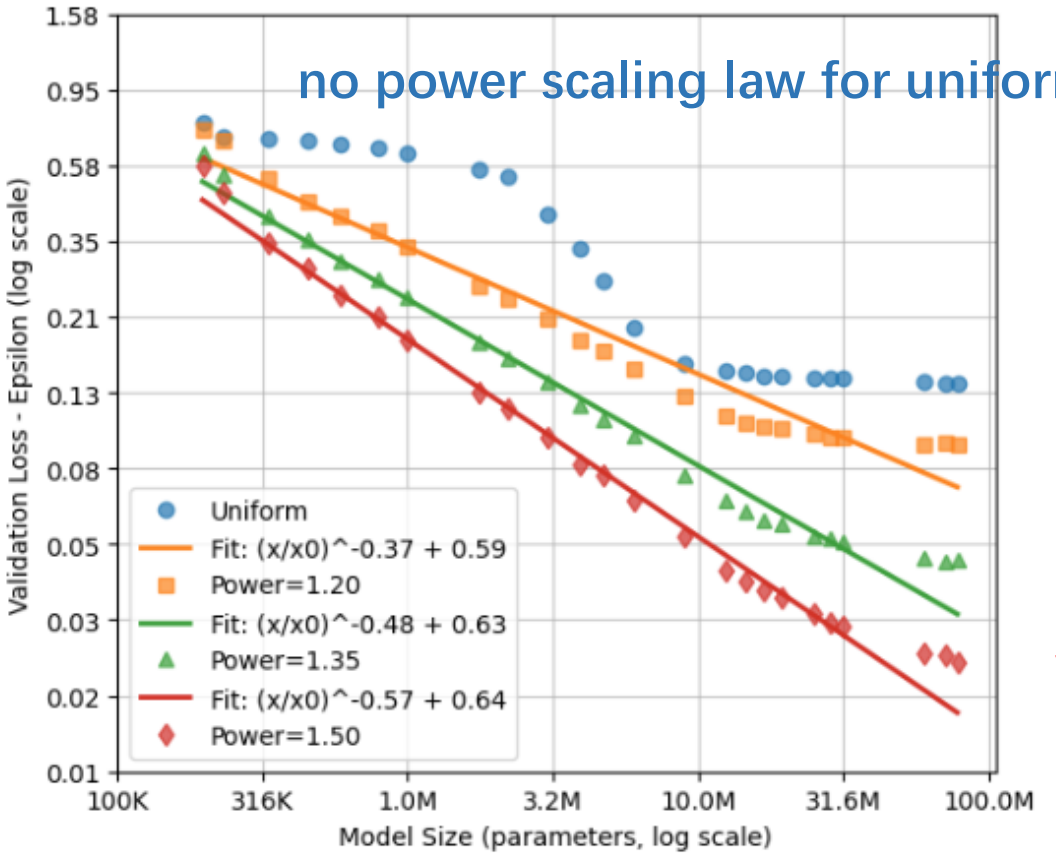
The solution of the optimization problem



$$\text{minimize} \quad \mathbb{E}_i[\mathcal{D}_i(m_i)] = \sum_{i=1}^{\infty} p_i \mathcal{D}_i(m_i),$$

$$\text{subject to} \quad \mathbb{I}(\Phi_0; \boldsymbol{M}_C^*) \leq C, \quad m_i = \mathbb{I}(\phi_i; \boldsymbol{M}_C^*) \geq 0 \text{ for all } i \in \mathbb{N}^+$$

Model Scaling Law

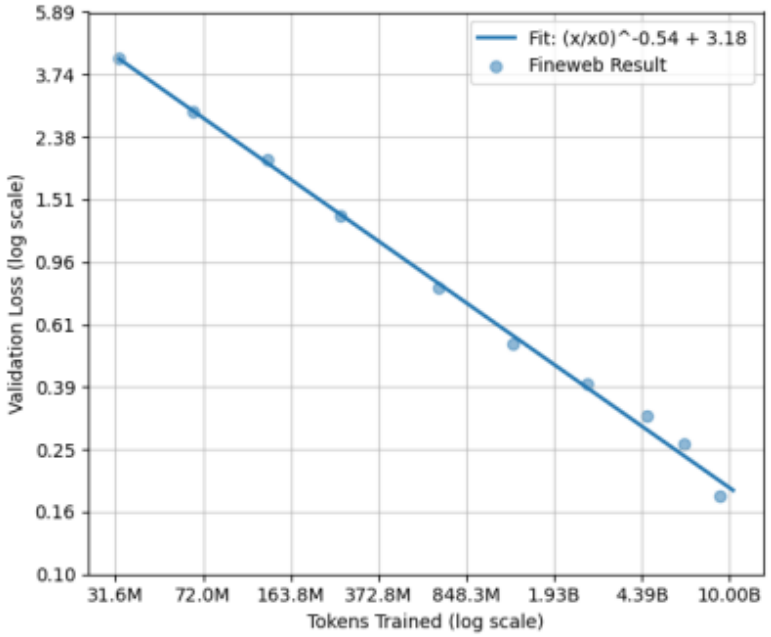


power law distributed data yields power scaling law

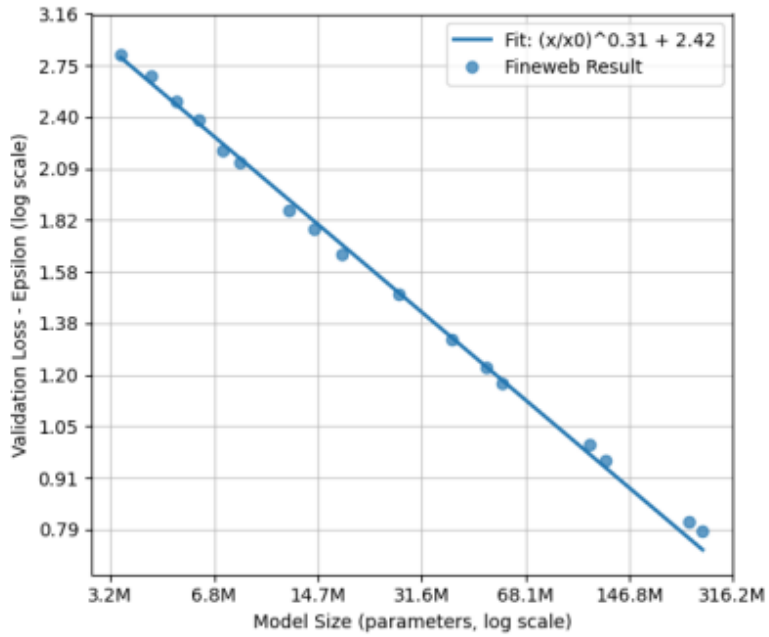
- For data generated from a uniform distribution, the loss curve significantly deviates from a power law

Experiments on Real-world Data

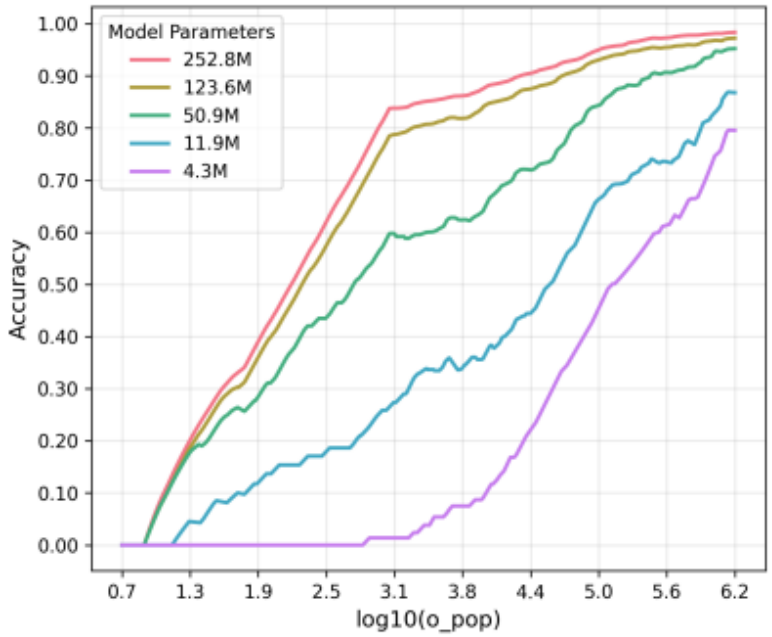
Real-world data naturally follows power-law distribution (so we can observe the power law shape scaling law)



Data scaling law:
The estimated exponent for the power-law of fineweb is between 1.35 and 1.5.



Model scaling law



frequency

PopQA dataset:
Smaller model cannot learn less frequent knowledge

Outline

- A Data-Centric Approach
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

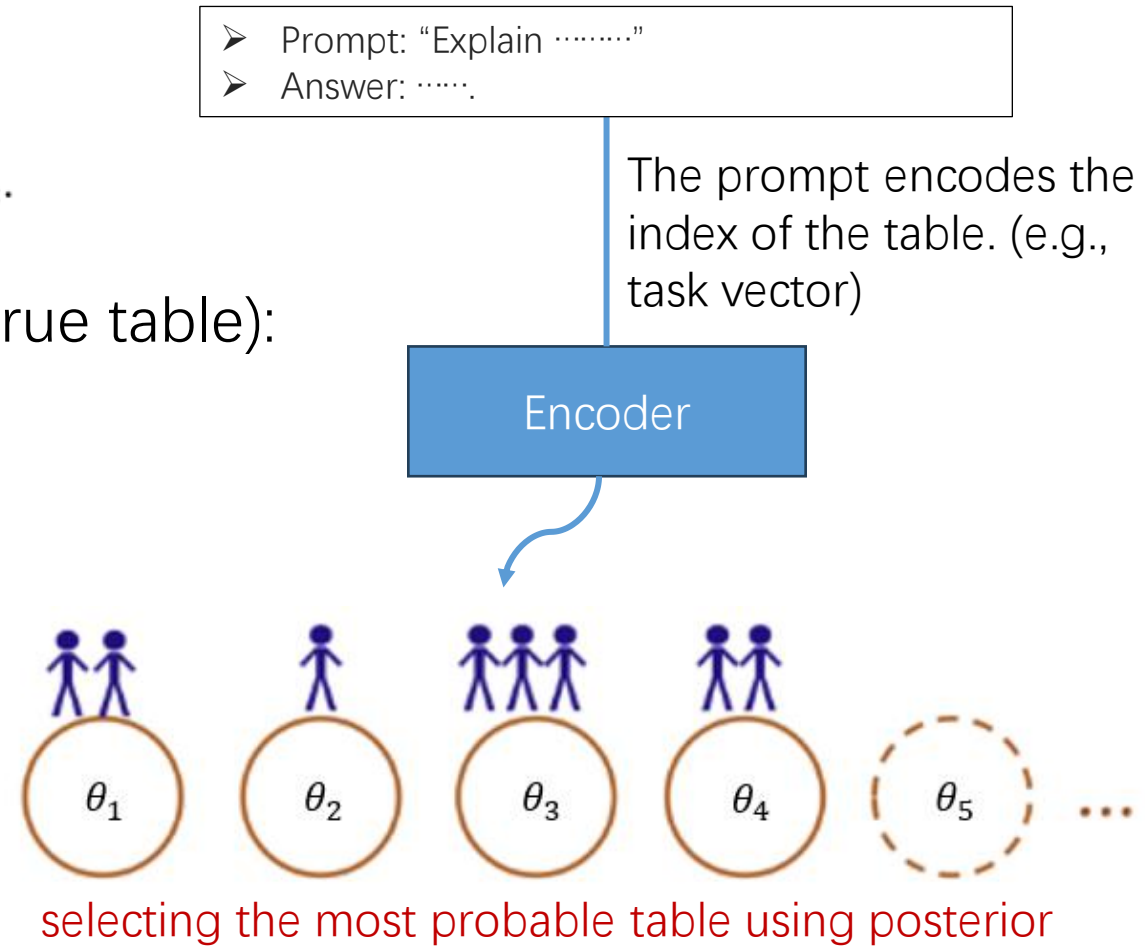
In-Context Learning

- Explaining In-Context Learning (Bayesian view)
  - Consider $P(next\ token \mid prompt)$
  - Intuition: Prompt increase the posterior probability of the relevant table
- Bayesian predictor:

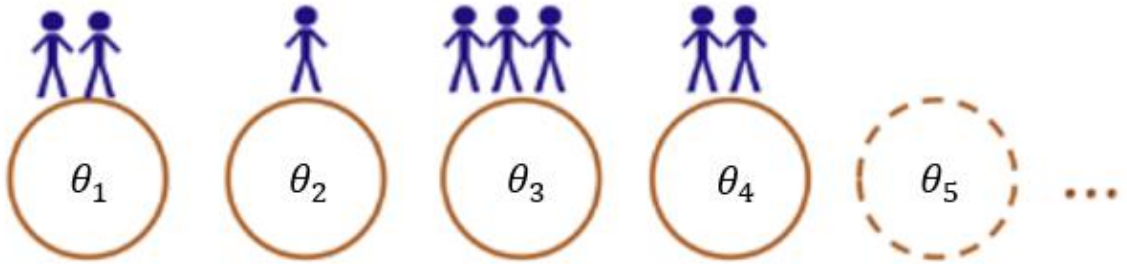
$$P_{\mathrm{ICL}}(x_n \mid x_{1:n-1}) = \int p(x_n \mid x_{1:n-1}, \phi_{data})\, p(\phi_{data} \mid X_{1:N})\, d\phi_{data}.$$

- Optimal predictor (knowing the index of the true table):

$$P_{\mathrm{OPT}}(x_n \mid x_{1:n-1}) = p(x_n \mid x_{1:n-1}; \phi_z)$$



In-Context Learning



Theorem (informal): under reasonable assumptions, the Hellinger distance between the Bayesian predictor and the optimal predictor is bounded as follows:

$$h(P_{\mathrm{ICL}}, P_{\mathrm{OPT}}) \leq \tilde{O}\left(\frac{1}{\sqrt{p_z N}}\right) + \exp\left(-\Theta(n)\right)$$

#pretrained data for table z

Len of prompt or #examples in the prompt

The ICL performance depends on the number of pretrained samples for the same/similar topic/table.

We only need log # examples to make the error small, if the table is well-trained in the pretraining phase.

Comparing with [Xie et al. An Explanation of In-context Learning as Implicit Bayesian Inference], our bound is quantitative and reflect the role of pretrained data

Outline

- A Data-Centric Approach
- Compression and Prediction
- Data Modeling (a nonparametric model)
- Data Scaling Law
- Model Scaling Law
- In-Context Learning
- Conclusions and Connections to LLM Practice

Conclusions

- Our theory is data-centric! (Kolmogorov structure function is defined over the data)
- Can provide guidance how to design/mix training data

- Limitation of our theory
  - only concerns the optimal learner
  - hence, cannot say anything the architectures and training algorithms (which also affect scaling laws)
  - real learner (transformer) is not optimal

LLM Pre-training

- Training data selection (increasingly important)
  - Measuring data quality
  - Data Pruning
  - Synthetic data generation (data augmentation, distillation)

Llama 3 is pretrained on over 15T tokens that were all collected from publicly available sources

- a series of data-filtering pipelines. These pipelines include using heuristic filters, NSFW filters, semantic deduplication approaches, and text classifiers to predict data quality.
- included code snippets from various languages aimed at enhancing the model's code generation and understanding capabilities; include mathematical data to enhance the model's reasoning capability
- Data Mix: to create a high-quality language model, it was essential to carefully determine the proportion of different data sources in the pre-training mix.
  - Knowledge Classification
    - The goal was to categorize the web data into different knowledge domains to ensure a balanced and diverse dataset.
    - Classifier: A classifier was developed to categorize different types of information present in the web data. The classifier categorized the data into various domains, such as arts and entertainment, science, technology, etc.
    - Downsampling Over-represented Categories: Categories that were overrepresented, like arts and entertainment, were downsampled.
    - Balancing the Mix: The data mix was balanced to provide a comprehensive knowledge base for the model
  - Scaling laws were used to determine the optimal size of the model and the data mix required to achieve the best performance given the computational budget (training on smaller dataset and models for different data mixes and then scale up).

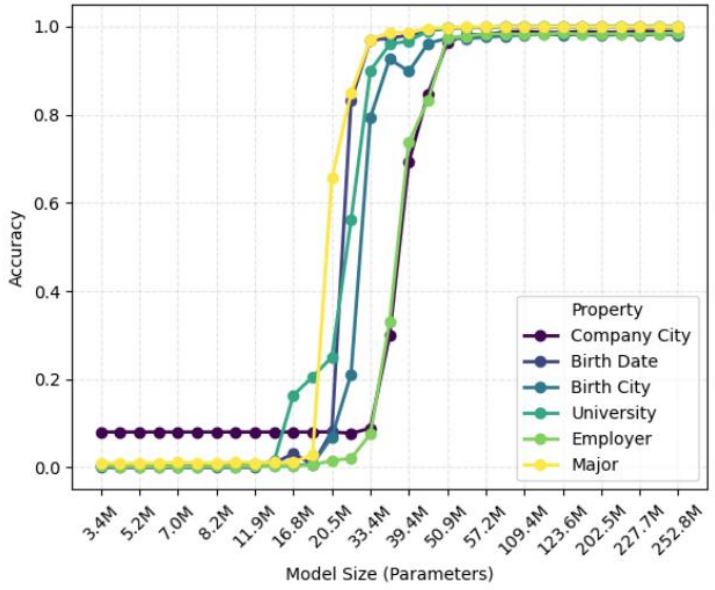
Potential Connections to LLM Practice

- **Data Mixing** strategies: phase transition in data mixing [Gu et al. 25]
  - Modifying mixing ratio = changing the data distribution
- **Synthetic Data Generation**
  - What syntax encoder to use? What is the mixture distribution is more efficient?
- **Instruction fine-tuning**: essentially change the syntax encoder
  - not very effective in injecting new knowledge
- **Tokenization**: design better syntax encoder (e.g., for better compression)
  - Ongoing work: a new tokenization method

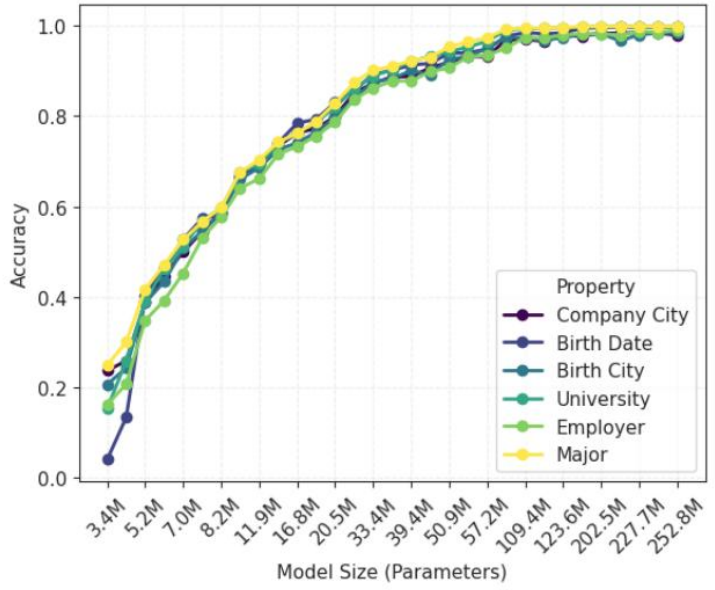
Connections to LLM Practice

Knowledge acquisition: uniform distribution vs power-law distribution (varying model sizes)

uniform distribution



a power-law distribution

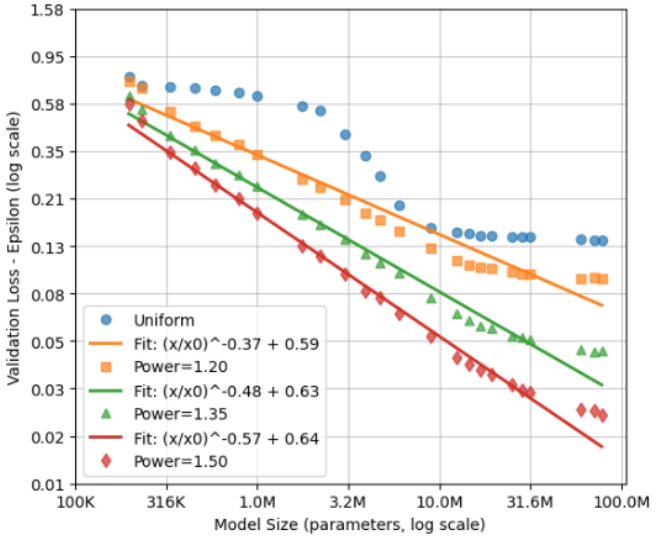


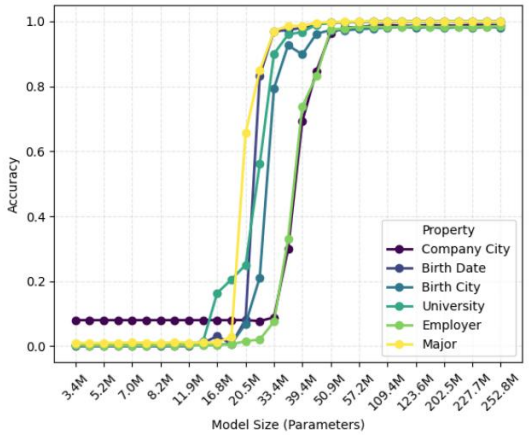
- (Factual) Hallucinations due to capacity constraints
  - low frequency knowledge will not be learnt
  - Important implications in data mixing

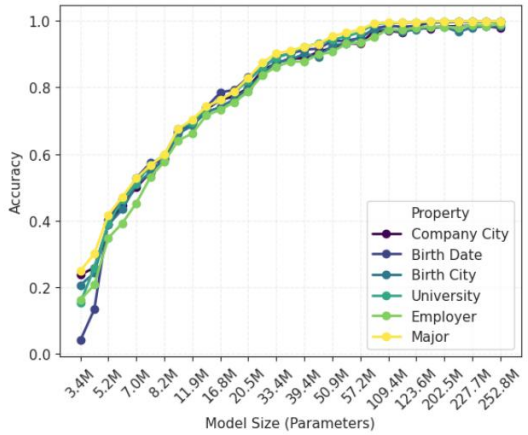
    e.g., Llama 3 data mixing strategy, CLIMB by NVIDIA

Potential Connections to LLM Practice

no power scaling law for uniform distr!







- For data generated from a uniform distribution, the loss curve significantly deviates from a power law
- It may be advantageous to have power-law-distributed data (next slides).
- Maybe more effective than the uniform case, where no one element stands out and the model lacks guidance on what to prioritize.

Future Directions

- More expressive data generative model
(beyond syntax and factual knowledge models)
  - Model task composition
  - Model reasoning
  - Kolmogrov's theory can be a good guideline

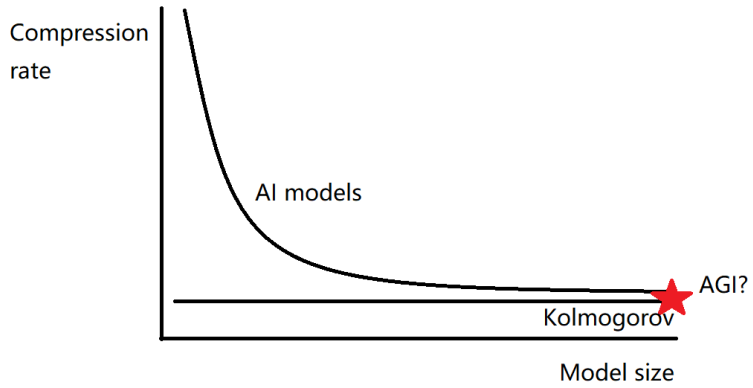
- Theory connecting data distribution to optimization?
  - E.g., power law data distr -> power law covariance spectrum -> scaling law?
  - How data distribution affects loss landscape

- Refined Universal Predictor (Solomonoff's) Model
  - How LLM approximate universal predictor empirically?

Discussions
Compression vs AGI？（Sutskever’s view)

Compression ≠ AGI!!

Ilya Sutskever “an observation of generalization”



压缩即智能？

Compression = Intelligence in inductive inference

压缩即推理智能

压缩无法产生探索智能（基于探索的创造力）

Compression by itself cannot reflect exploration of physical world and mathematical/logical reasoning

Thanks