Understanding the Behaviors of LLMs

A perspective based on Komolgorov Complexity and Shannon Information Theory

Jian Li ấ

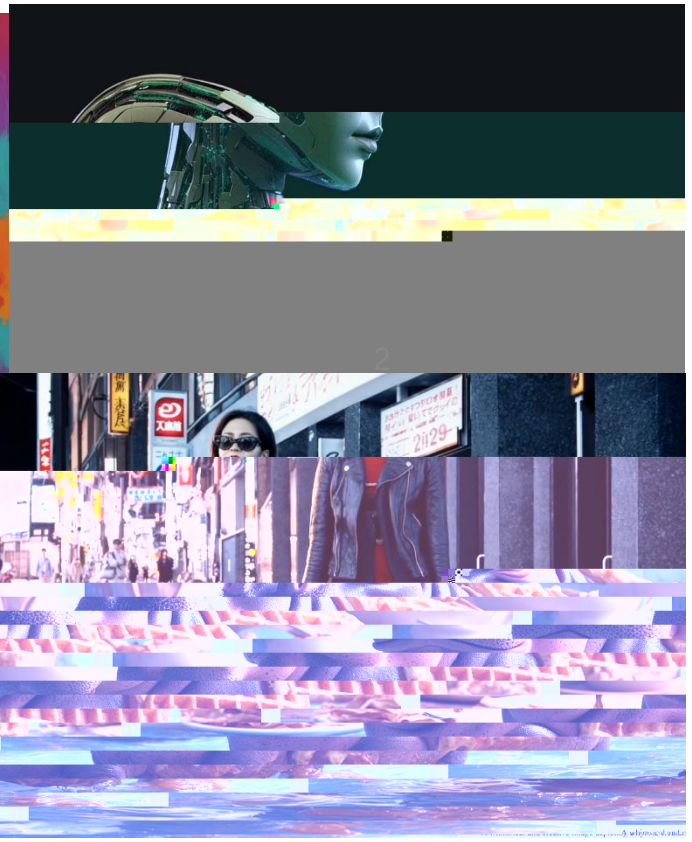
Institute of Interdisciplinary Information Science

Tsinghua University

Ľ z ʒ Ụ ˥˧ ʊ ▌æ ʑ ę

Why DL and LLMs Work So Well?

- ‡ Tremendous success in practice
- ‡ AI models are still big black boxes
- ‡ Theory, several exciting recent results (still not so satisfying)



deepseek

探索未至之境

Theory of Deep Learning & LLMs

- Theory of DL
  - Optimization (new phenomena)
  - Algorithm-dependent 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

h h O h h G h h NO h

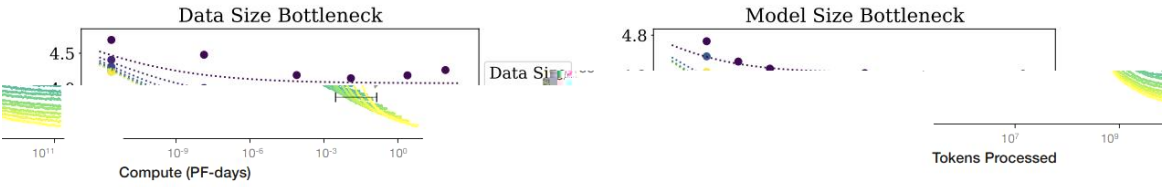
Scaling Laws

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

‡Kaplan Scaling Law (OpenAI)

Ù¿ ý 0.076, 0Ö ý 8.8 pd1013 (non-embedding parameters)

Ù½ ý 0.095, &Ö ý 5.4 pd1013 (tokens)



‡Chinchilla Scaling Laws (DeepMind)

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

Ò= 1.69, ö = 406.4, ő = 410.

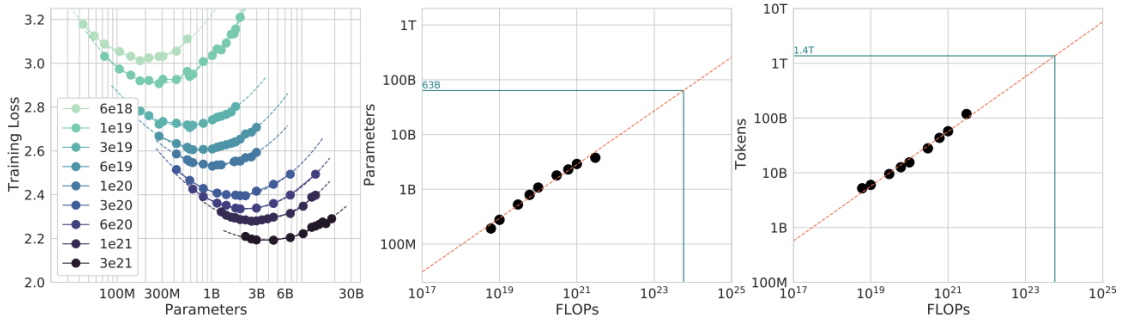


Figure 3 | **IsoFLOP curves.** For various model sizes, we choose the number of training tokens such that the final FLOPs is a constant. The cosine cycle length is set to match the target FLOP count. We find a clear valley in loss, meaning that for a given FLOP budget there is an optimal model to train (**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

Compression and Intelligence

$Q 2EVHUYDWLRQ RQ *HQHUDOL]DWLRQ³ E\ Ilya Sutskever

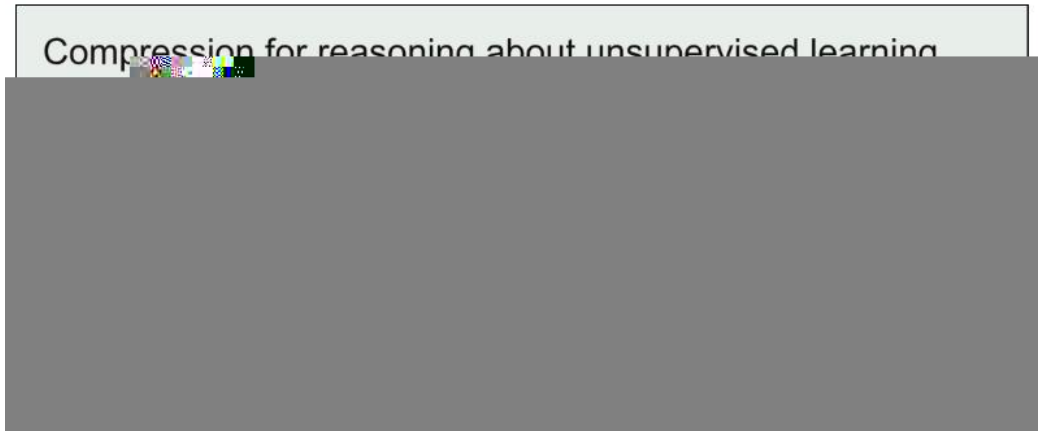
Very good talk (watch on youtube : https://www.youtube.com/watch?v=AKMuA_TVz3A)

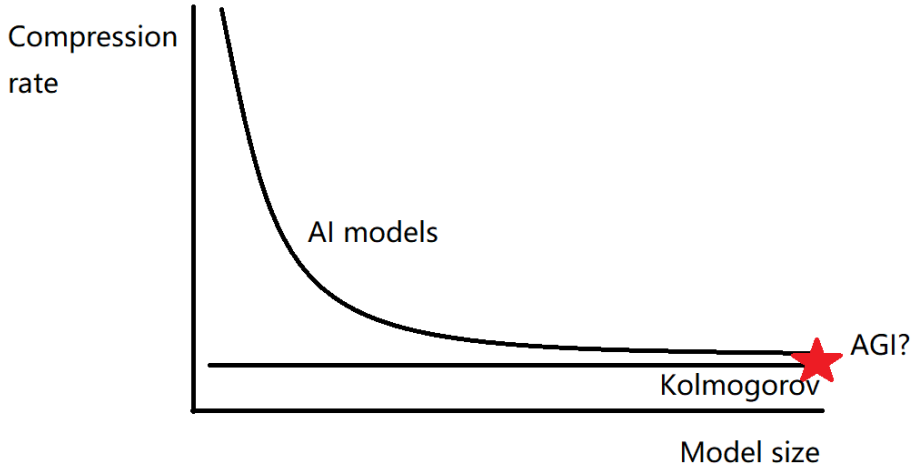
View compression from Kolmogorov complexity theory

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

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

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

Outline

- ‡LLM Theory
  - ‡Fundamental Ideas from Shannon and Kolmogorov
  - ‡Compression and Prediction
  - ‡ . R O P R J R U R Y ¶ V W K H R U \
  - ‡Data Modeling (a nonparametric model)
  - ‡Hallucination and ICL
  - ‡Universal Predictor
  - ‡Research Directions

Prediction vs Compression

The trainingtaskof a pre-trained large language mode(LLM)is "next token prediction," so we can naturallyview a pre-trained LLMas a**next tokenpredictor**.

Given an unknown source distribution2, a predictor is defined as3ã: Û \ ¿Ç which given the prefix T· Þas input and outputs the conditional probability distribution 3 : TÞ T· Þ;.

Cross Entropy Loss

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

$$\mathcal{L}(M) = -\log P_M(X_{1:n}) = -\frac{1}{n}\sum_{i=1}^{n} \log\, P_M(X_i) = \mathbb{E}_{X\sim\widehat{P}_\phi}\left[-\log P_M(X)\right]$$

$$= H(\widehat{P}_\phi \| P_M) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{t} \log\, P_M(x_t^{(i)} \mid x_{1:t-1}^{(i)}),$$

Prediction vs Compression

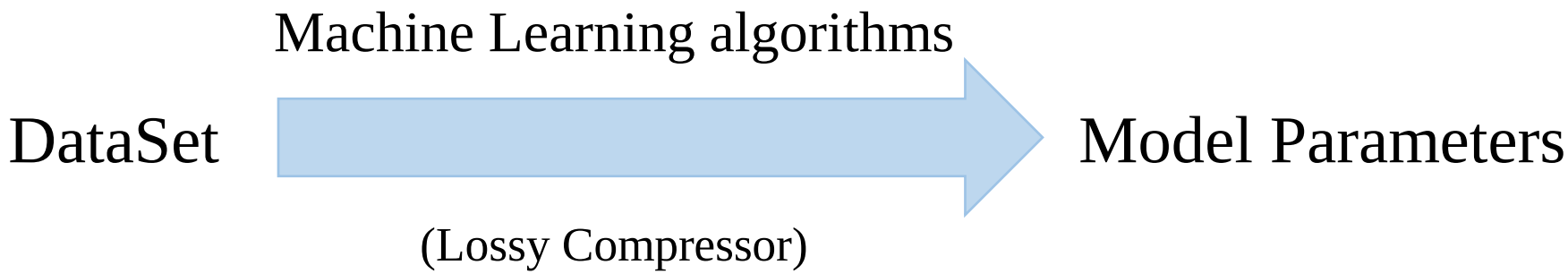
Equivalence of Prediction and Compression

The better we can predict next token, the better we can compress the sequence

We will show that if we can achieve a cross-entropy C (per token), we can essentially compress the text using C+o(1) bits (per token), and vice versa.

Equivalence of Prediction and Compression

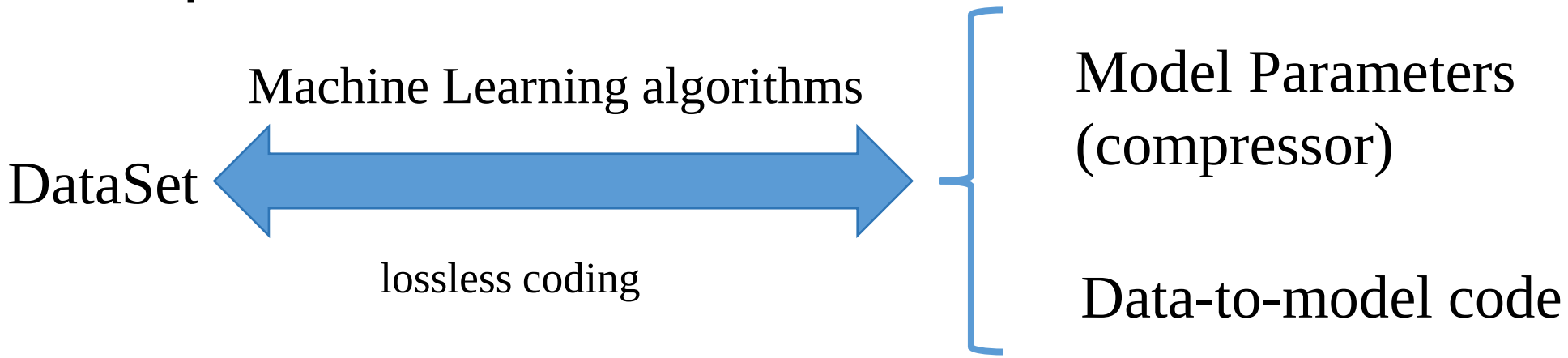
Lossy Compression?



Limitations: 1.Too much loss
2.No Guaranteed Generalization

Equivalence of Prediction and Compression

Lossless Compression



e.g. winzip (a compressor)

One can use LLM as a more powerful compressor

Equivalence of Prediction and Compression

Why lossless compression leads to **intelligence**

- If the model can losslessly compress well, it should have learned real feature in the dataset and will generalize well.

- **Minimal Description Length MDL**

The best interpretation of a set of data is a description of that data that is accurate and as short as possible.

- **Occam's Razor:**

Entities should not be multiplied unnecessarily.

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.

Lossless Compression/coding

‡ Compressor encoder?ã: Û \ < rás=Û

‡ There exists a decoder@ã4: ?; \ : Û satisfies that@?: T5ãã ; ; L T5ãã .

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

. Ö L ' ë1 >H Ö: T; ?

where H Ö means the bit length of ?: T;.

Shannon Sourcing Coding Theorem

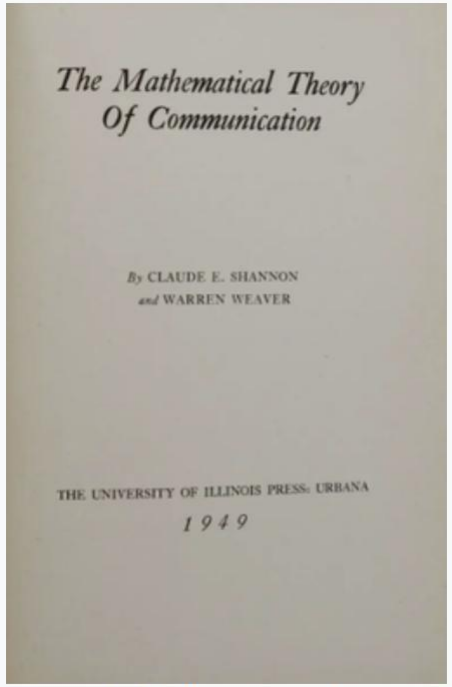
Shannon's Source Coding Theorem

Given some categorical distribution X, Shannon's Source Code Theorem tells us that no matter what C you choose, the smallest possible *expected code word length* is the entropy of X. That is,

$$E\left[|C(X)|\right] = \sum_{x \in \mathcal{X}} |C(X)| P(X = x) \geq -\sum_{x \in \mathcal{X}} P(X = x) \log P(X = x) = H(X)$$

More formally:

Theorem 1 (Shannon's Source Coding Thoerem): Given a categorical random variable X over a finite source alphabet $\mathcal{X}$ and a code alphabet $\mathcal{A}$, then for all uniquely decodable $C : \mathcal{X} \to \mathcal{A}^*$, it holds that $E[|C(X)|] \geq H(X)$.



1949 full book edition

https://mbernste.github.io/posts/sourcecoding/

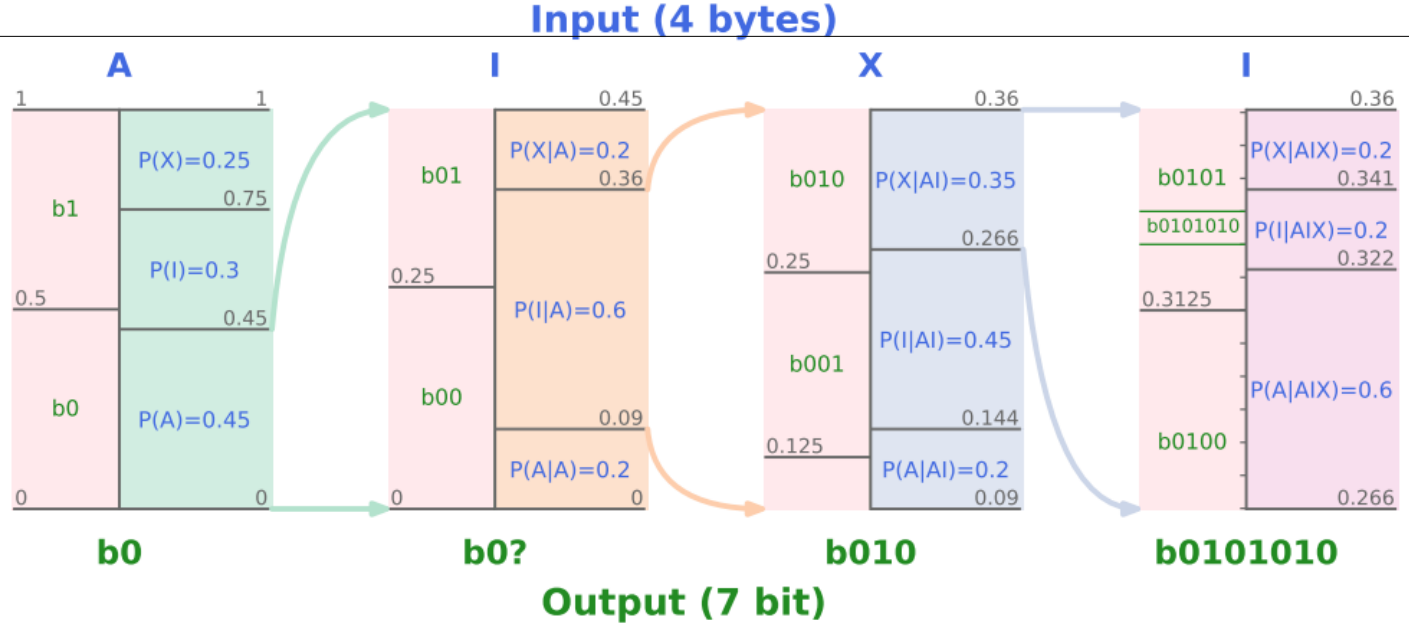
Using an Autoregressive Predictor to design a Lossless Compress

Arithmetic Encoder:

Initially, this interval is $\pm_4$ L > r á s ; .

When encoding $T_Þ$, we first partition the previous interval $\pm_{Þ?5}$ L > $H_{Þ?5}$ á $Q_{Þ?5}$; into ű sub-intervals $\pm_Þ$: T_5 ; á $\pm_Þ$: T_6 ; á ä ä ä one for each letter from L < T_5 á ä ä á $T_Ç$. The size of sub-interval $\pm_Þ$: U ; that represents letter U is : $Q_{Þ?5}$ F $H_{Þ?5}$; I 2 : U T O G ; ä

e.g., encoding AIXI



The encoding length of arithmetic encoder is H_O : $T_{5ãá}$; L F > H K_6 2 : $T_{5ãá}$; ?E s

Prediction vs Compression

Equivalence of Prediction and Compression

The better we can predict next token, the better we can compress the sequence

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

Redundancy of code

KL divergence

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

Outline

- ‡LLM Theory
  - ‡Fundamental Ideas from Shannon and Kolmogorov
  - ‡Compression and Prediction
  - ‡ . R O P R J R U R Y ¶ V W K H R U \
  - ‡Data Modeling (a nonparametric model)
  - ‡Hallucination and ICL
  - ‡Universal Predictor
  - ‡Research Directions

Kolmogorov Complexity

Kolmogorov Complexity (algorithmic information theory)

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

(a fine-grained structure: Kolmogorov structure function)

‡ Direct connection to compression

‡ In some sense, the ultimate notion of compression

‡ o×p o³p p p o³p oøoøo÷o³p p o³oþp p p o³p oûoøo³oøp oôoöp o³o÷oüp p p oüoõp p oüp p o³o»p p oÿ theory)

‡ Downside: incomputablep

Examples:

‡ abababababababababababababababa....

‡ 4c1j5b2p0cv4w1x8rx2y39umgw5q85s7...

‡ 3141592653589793238462643383279 5...

p oøoøo³p oûoøo³oöoÿoôp p oüoöo³oõp p oþo³p!oøoÿoøp o

Kolmogorov Complexity

Can the following program output K(s)?

```
function KolmogorovComplexity(string s)
    for i = 1 to infinity:
        for each string p of length exactly i
            if isValidProgram(p) and evaluate(p) == s
                return i
```

Universality of Kolmogorov complexity

Theorem 14.2.1 (*Universality of Kolmogorov complexity*) *If $\mathcal{U}$ is a universal computer, for any other computer $\mathcal{A}$ there exists a constant $c_{\mathcal{A}}$ such that*

$$K_{\mathcal{U}}(x) \leq K_{\mathcal{A}}(x) + c_{\mathcal{A}} \tag{14.3}$$

for all strings $x \in \{0, 1\}^*$, and the constant $c_{\mathcal{A}}$ does not depend on x.

Kolmogorov Complexity

‡Algorithmic randomness

‡What is a random string? A impressible one.

‡We say a seq $T_5 .. T_á$ is algorithmically random if

$$K(x_1 x_2 \ldots x_n | n) \geq n.$$

‡Most sequences are random (interesting sequences are rare)

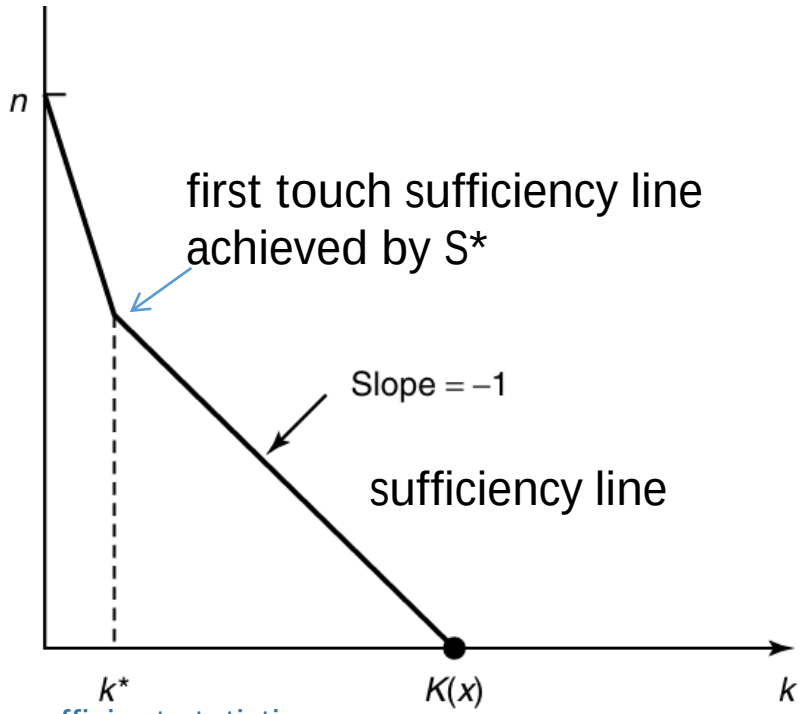
Theorem 14.5.1 *Let $X_1, X_2, \ldots, X_n$ be drawn according to a Bernoulli $(\frac{1}{2})$ process. Then*

$$P(K(X_1 X_2 \ldots X_n | n) < n - k) < 2^{-k}. \qquad (14.44)$$

This can be easily seen from the fact that $|\{x \in \{0,1\}^* : K(x) < k\}| < 2^k$.

Kolmogorov structure function

$$h_X(\alpha) = \min_S\{\log|S| : S \ni X; K(S) \leq \alpha, \}$$



minimum sufficient statistics

FIGURE 14.4. Kolmogorov sufficient statistic.

‡ Two part code (model-to-data code)

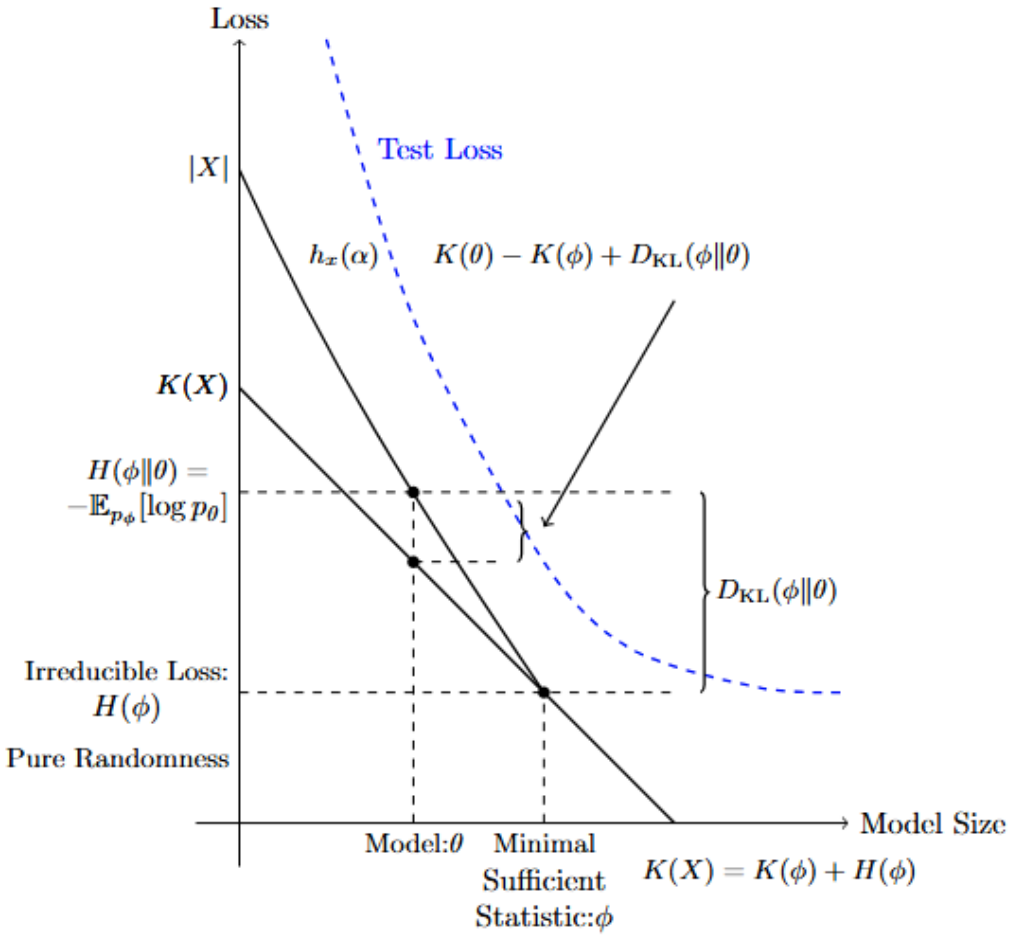
$$K(X) < |C(X)| + K(C) + O(1)$$

‡ Hence the set S Ûcaptures all the structure within x.

‡ The remaining description of x within S Ûs essentially the description of the randomness within the string.

‡ Hence S* is called the Kolmogorov sufficient statistic for x.

Kolmogorov structure function



‡ Two part code (model-to-data code)

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

‡ Closely related to Scaling Law

‡ Distinguish structure & random noise
 ‡ a fine-grained hierarchy

‡ Why a power law shape?

‡ oÔo³oöoûoôp oôoöp oøp oüp oôp oüp oÿoøoôp p p p"o³oôp o÷o³p!p oûoôp o³

Outline

- ‡LLM Theory
  - ‡Fundamental Ideas from Shannon and Kolmogorov
  - ‡Compression and Prediction
  - ‡ . R O P R J R U R Y ¶ V W K H R U \
  - ‡Data Modeling (a nonparametric model)
  - ‡Hallucination and ICL
  - ‡Universal Predictor
  - ‡Research Directions

Data Modeling A Hierarchical Nonparametric Model

A Hierarchical Nonparametric Model:

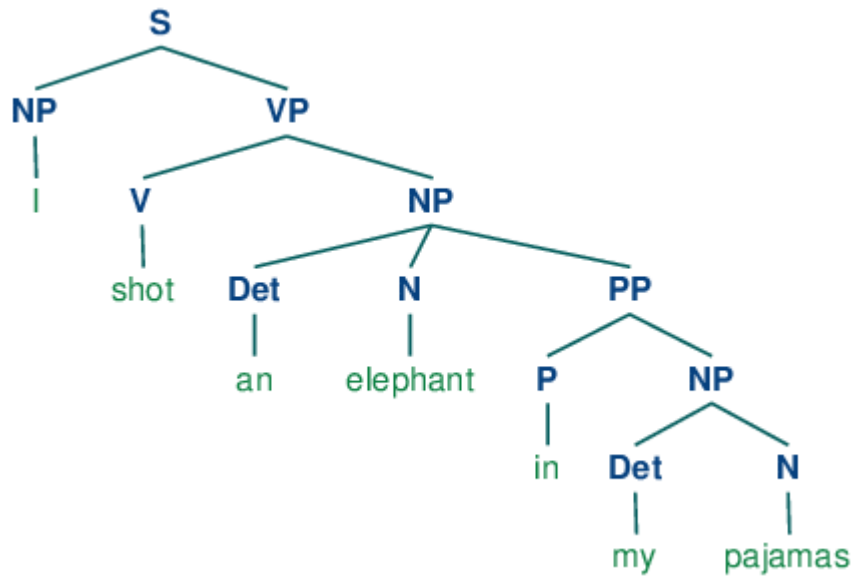
‡ An encoder(syntax, most common knowledge, basic logic): can be captured by a fair small- sized probabilistic TM that is fairly easy to learn (low sample complexity).

‡ 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							



Data Modeling A Hierarchical Nonparametric Model

A Hierarchical Nonparametric Model:

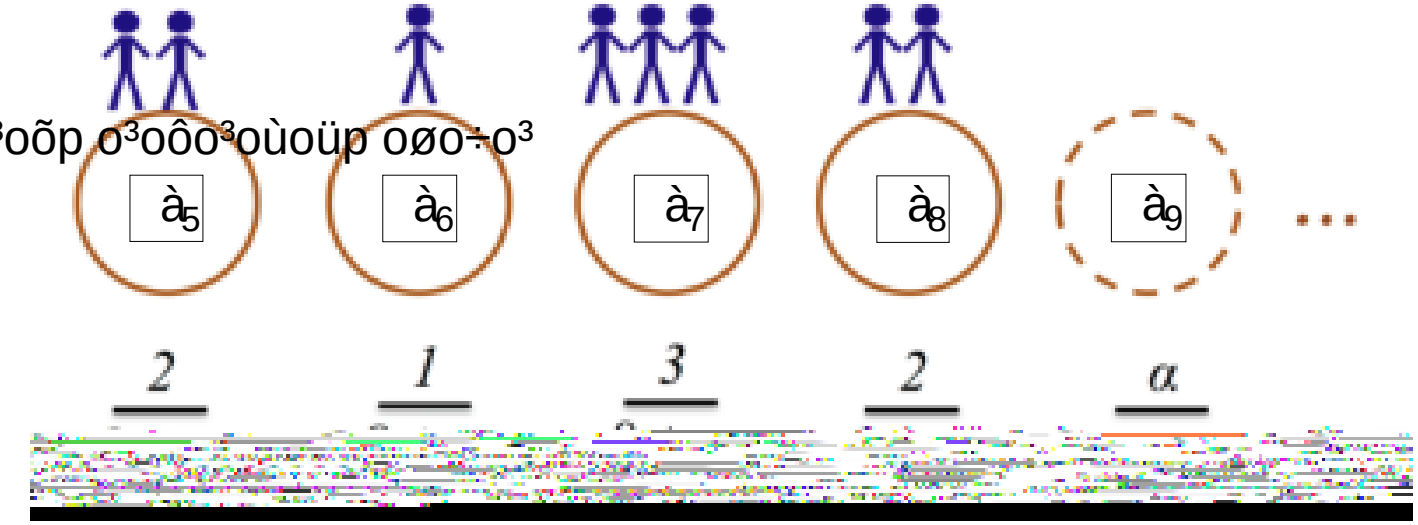
‡ An encoder (syntax, most common knowledge, basic logic): can be captured by a fai small- sized probabilistic TM that is fairly easy to learn (low sample complexity).

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

World (factual) Knowledge:

‡ oÖoôp p p o³oõoøo³oöoôp p p p oøo÷o³oõp o³oôo³oùoüp oøo÷o³ parametric model

‡ Modeled by Pitman- Yor Chinese Restaurant Process (PY- CRP)



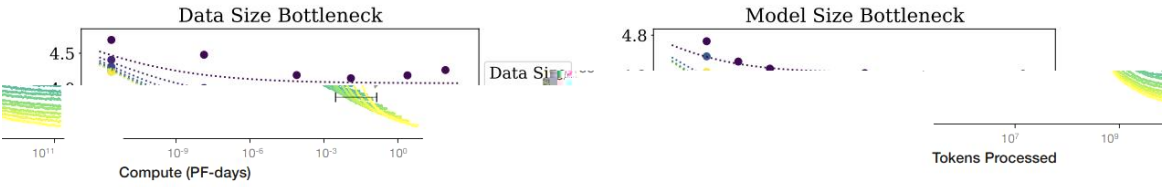
Scaling Laws

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

‡Kaplan Scaling Law (OpenAI)

Ù¿ ý 0.076, 0Ö ý 8.8 pd1013 (non-embedding parameters)

Ù½ ý 0.095, &Ö ý 5.4 pd1013 (tokens)



‡Chinchilla Scaling Laws (DeepMind)

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

Ò= 1.69, ö = 406.4, ő = 410.

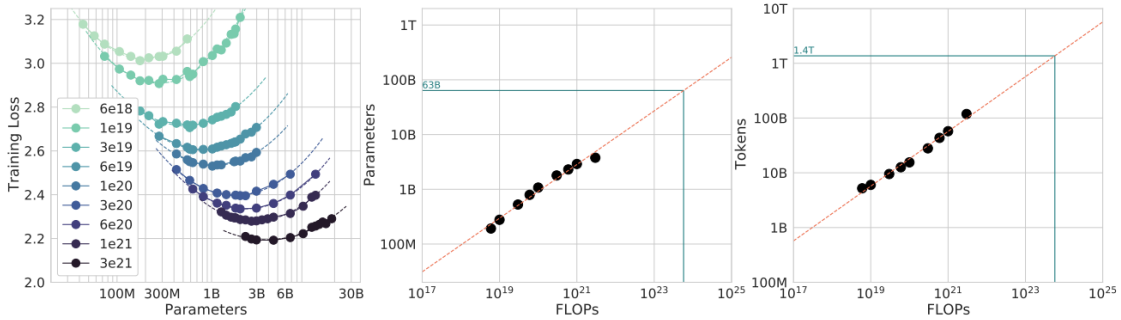


Figure 3 | **IsoFLOP curves.** For various model sizes, we choose the number of training tokens such that the final FLOPs is a constant. The cosine cycle length is set to match the target FLOP count. We find a clear valley in loss, meaning that for a given FLOP budget there is an optimal model to train (**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

A Coding 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 $x_1 \ldots x_n$ one by one
- Minimize the following log-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)

A Coding Game

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 redundancy and mutual information

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.

Understanding Scaling Law

Under the Bayesian prediction framework, one can show that the minimum Bayesian redundancy is equal to the mutual information (between the data and prior)

Theorem (informal): Under the above data generative model and unlimited model size, we can prove that the Bayesian predictor (or Maximum Likelihood predictor in large data regime) has the following loss

Loss incurred by learning the knowledge (power law)

Loss incurred by learning the syntax encoder

Irreducible loss (pure randomness)

‡ Related to Heap s Law (Heaps, 1978)an empirical relationshipstating that the vocabulary size grows sublinearly with the size of a corpus N ,and Zipf s Law (Zipf, 2016).

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

$$\begin{aligned}\text{minimize}\quad & \mathbb{E}_i[\mathcal{D}_i(m_i)] = \sum_{i=1}^{\infty} p_i\mathcal{D}_i(m_i),\\ \text{subject to}\quad & \mathbb{I}(\Phi_0;M_C^*)\leq C,\quad m_i=\mathbb{I}(\phi_i;M_C^*)\geq 0 \text{ for all } i\in\mathbb{N}^+\end{aligned}$$

can be solved using KKT condition introduced in the last class

Understanding Instruction Fine-Tuning

‡ Instruction fine-tuning:

- ‡ Replace the encoder with another one (change the way the knowledge is encoded)

‡ The knowledge component stays the same

We can show the loss for Instruction fine-tuning can be bounded as (J ‘s the size of Instruction F-T data)

Table 1. Examples of Pretraining and Instruction Fine-Tuning Data

Dataset Type	Example
Pretraining	"Gracie Tessa Howell was born in Camden, NJ. He studied Biomedical Engineering and worked at UnitedHealth Group. He entered the world on April 15, 2081, and is employed in Minnetonka. He is an alumnus/alumna of Buena Vista College."
Instruction Fine-Tuning	"Q: What area of study did Gracie Tessa Howell focus on? A: Biomedical Engineering"

$$\frac{1}{\text{Red}_{\text{[illegible]}}} = \mathbb{I}(\kappa_{1:N+n}; \phi_{\text{[illegible]}}) - \mathbb{I}(\kappa_{1:N}; \phi_{\text{[illegible]}}) = \frac{(N+n)^{\alpha} - N^{\alpha}}{\text{[illegible]}} \overset{(1)}{=} O(N^{\alpha-1})$$

Knowledge in pretrained phase is retained

Redundancy incurred by learning the new encoder

Practical Implication: Instruction F-T is more useful for generation according to the instruction, but less effective for injecting new knowledge

Universal Predictor: Solomonoff's theory

- Hypothesis: modern LLMs is a (rough) approximation of oæp oÿp p p p p o predictor.
  - We can explain various behaviors of LLMs using this theory
- Drawbacks
  - Unfortunately, oæp oÿp p p p predictor is again incomputable.
  - oæp oÿp p p p predictor ignores the following important aspects
    - the model size constraint
    - Architecture constraints of transformer (can not represent certain function composition)
    - Some (even simple) TMs are not efficiently learnable (XOR problem, one way functions etc.)
  - We should refine Solomonoff's theory to take account of the above (computational and statistical) barriers

Research Questions (LLM theory)

‡Theory on more general data generative model (beyond syntax and factual knowledge models)

‡ . R O P R J theory can be a good guideline

‡ Refined Universal Predictor (6 R O R P R)

‡ Methodology: Physics of LLMs (controlled experiments)

‡Scaling law, ICL, instruction following, emergence etc.

‡Data <----> Skills

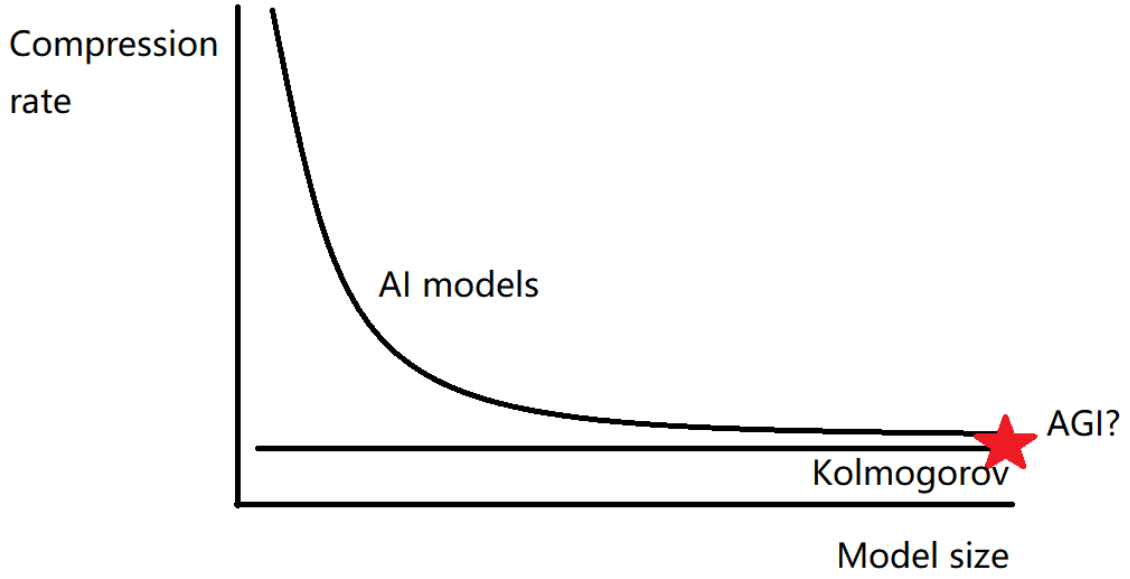
‡Data importance

‡High quality synthetic data (motivated from theory)

‡Detecting Hallucination/Safety issues (from activation pattern etc.)

Discussions

Compression vs AGI ?



Ilya Sutskever [3] DQ REVHUYDWLRQ RI JHQHUDOL]DWLRQ´

Discussions

Compression • $ * ,

"ŠN("r5yOüo ^Ø)<ð!:Eƒ"ä*O5.6 +?eO/&Eƒo o

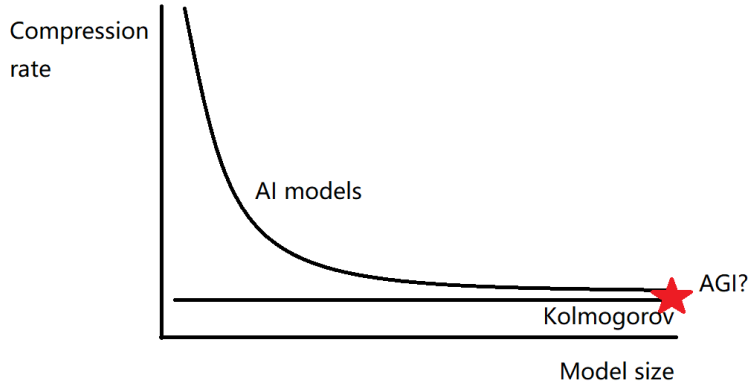
Compression = Intelligence in inductive inference

"ŠN("r2§C 5yOü

"ŠN(4ß;Ô ¦D 2¡L!5yOü (&ù •2¡L!Eƒ! _ !šo

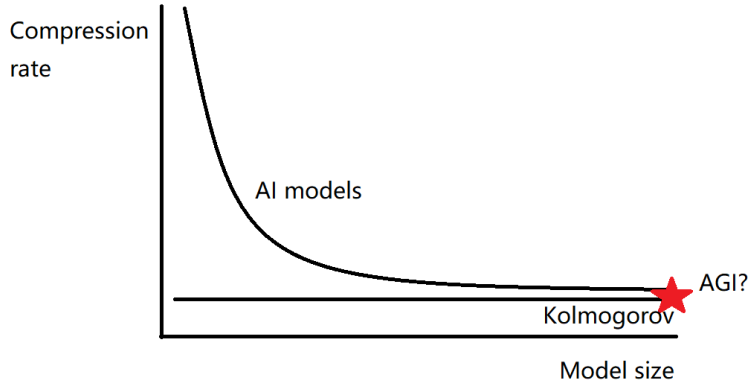
Compression by itself cannot reflect exploration of physical world and mathematical/logical reasoning, in my opinion

Ilya Sutskever ³ D Q R E V H U Y D W L R Q R I J



Discussions

Compression vs AGI ?



Compression = Intelligence in inductive inference

Ilya Sutskever ³ D Q R E V H U Y D W L R Q R I J

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

1 XÈ.–p""ŠN("r5yOüp!"éi&ductive inferenceR Ds 5.*ø.– <; 6 capture5yOü R :!§2¡L! exploreB®',
o #‹C Z¹o #‹45eÅ4o2mEƒ^ÆI <

1 .‡e½1‰xploreB®',Eƒ!9>¿!§!šD' compression2Î^ï ù6d < ÿ)At.9EƒZó;Ôo! 9 …5ó(|Eƒ predict next
tokeno eÿX€ :!§2¡L! DK"ÐB¯XÃ.Šo @5# predict next token#‹"ŠN(6 rKú <

.R@52¡L!.–!/ …4o2mo ! 6•/:MÒ^Ø š4o2mo 6d"ÐB¯XÃ.Šo eÿX€Eƒ+05.5ó(|Eƒ"ŠN(^Ø š4o2m <

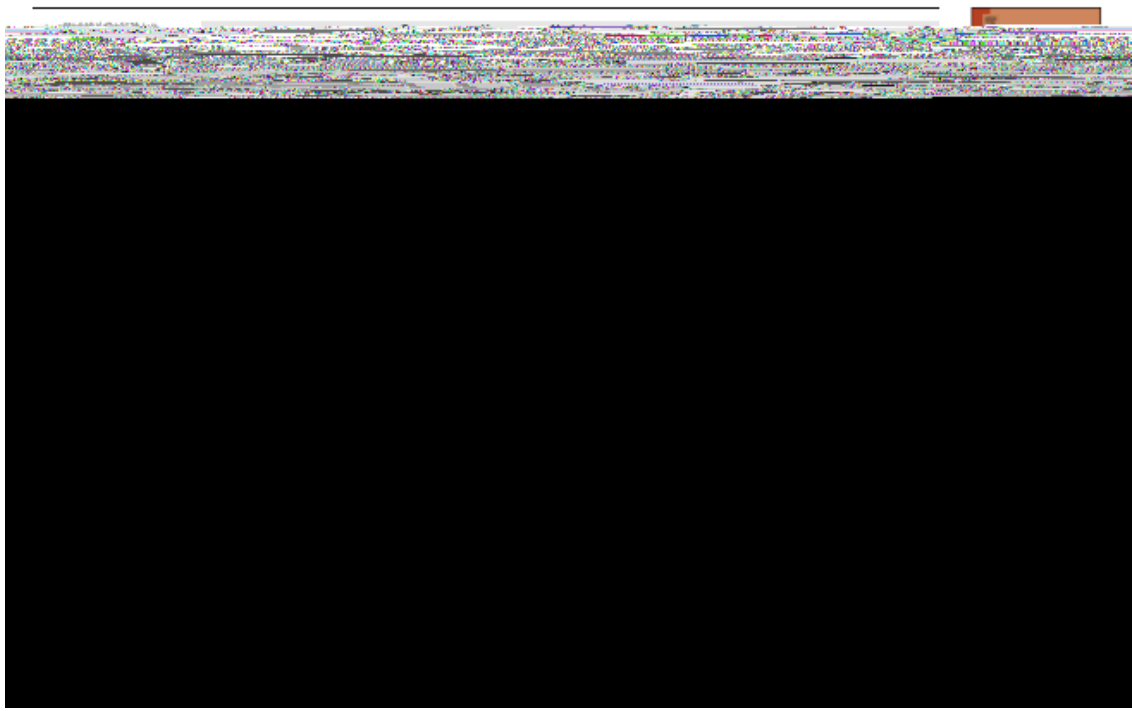
"ŠN("r5yOü2 Eƒ5.R¶.–4o2m# Eƒ5yOü^ÆI o E5. ¿ Gip!§2¡L!o 1 XÈ.–D'"ŠN(Xâ`É … < ^+05. Ilya
talk`Ë6)%ý H minimize perplexity!/ … AGI <

Discussion: RL and Reasoning

^Ø)XÁ?¸ 1 XÈ.– reward is enough_¢) paperXÁ?¸5ó786+ <

5yOü ¦D Eƒ5ó786+Eƒ"ž%ß5.4ßeOEƒ DK*ø6 eO\C=•5yOü R
&'Ax*™/Ä´ MØ rewardo 6 eO\C=•5yOü REí7 +05.5ó6
4GEƒR¶.–^Ø šward <that is all < 9 …R¶.–rewardo "º
exploreo R¶.–Eƒdata_ ^Æcompressh• G ¦D … intelligentEƒ
*ø DKEƒZ£ZÅo @5# 'lÇtelligentEƒbehavioro @5# Œ5ó(|
Eƒ45eÅ4o2m <

:Ó(•*ú1} math proof^Ø) Š/Äo #‹ ¹KzZ£ZÅ DKKz ;o 1 Zó
Eƒ_¢)proof6ƒ1 Eƒ4ßeOEƒ%ý`Ë6 4ß4o ¹KzEí!L r.ÂEƒ*™
C o :Ó(• ÿ)6 1000)6` õEƒ*™C o o Eí!L^Ø š)?¸ ;
6 $dreward <Eí!LEƒ4o2mo 6 rewardEƒo "écover u ,.‡+
Eƒproportion <^Ø) reward5.Eí!L ¹Kz4o*e*µ\J ‡Eƒo ^Ø)
reward function#‹4o*eG Iu`ËfaEƒ ÿ š : HE÷ ro :Ó(•EóXÈ
: HEƒo :Ó(•* N•o ^×6 4o*e5.#%eÿX€#‹B¯*• DK^Ý2¤o
^×5.*WK¸Eƒ2§Co von Neumann, Taoo <



But do we really need RL?? RL vs SFT on smartly collected data