Bayes and language models, three ways

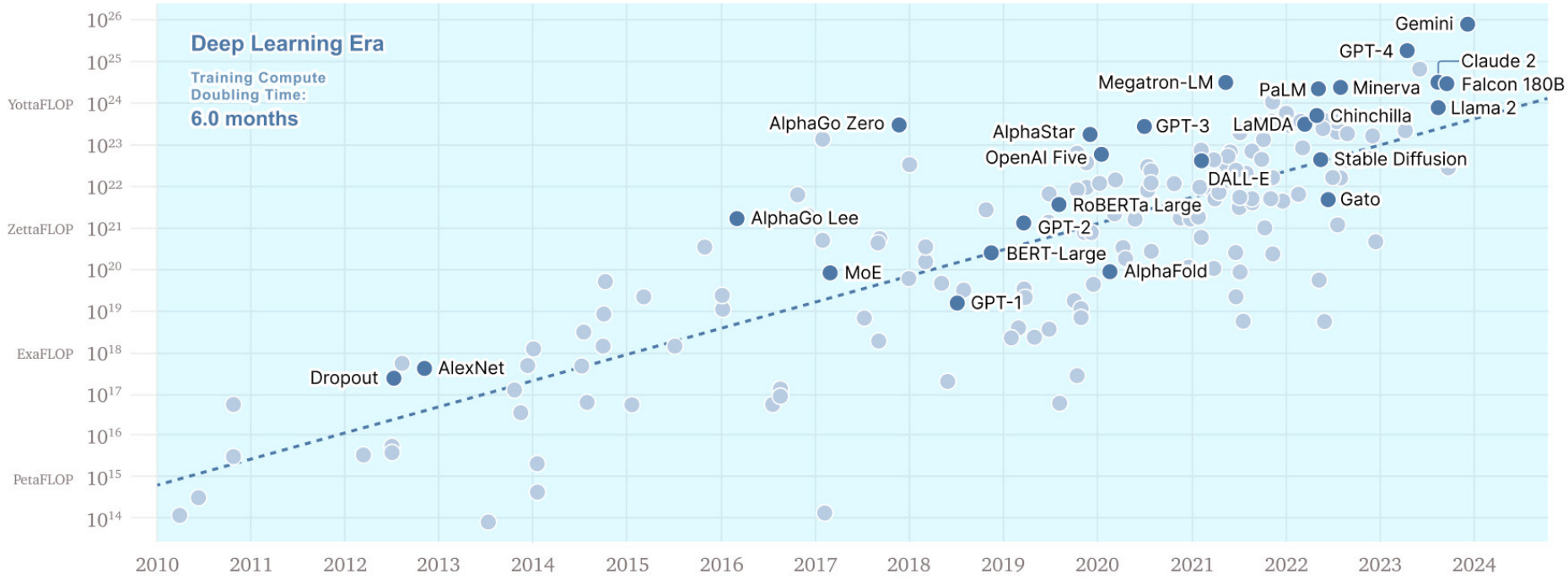
Tom Griffiths
Department of Psychology
Department of Computer Science
Princeton University

Bayesian models

- Easy to understand
- Clear inductive biases
- Typically tailored to specific problems
- Scaling is a challenge

Deep neural networks

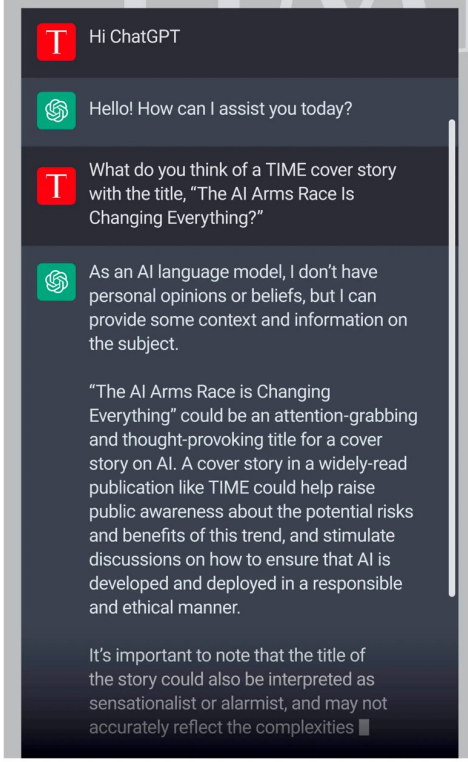
- Difficult to understand
- Opaque inductive biases
- Succeed on a surprisingly wide range of problems
- Process lots of data



"Computing Power and the Governance of Artificial Intelligence"
Sastry, Heim, Belfield, Anderljung, Brundage, Hazell, O'Keefe, Hadfield et al., 2024

FEB. 27 / MARCH 6, 2023

TIME



by
ANDREW R. CHOW & BILLY PERRIGO
(humans)



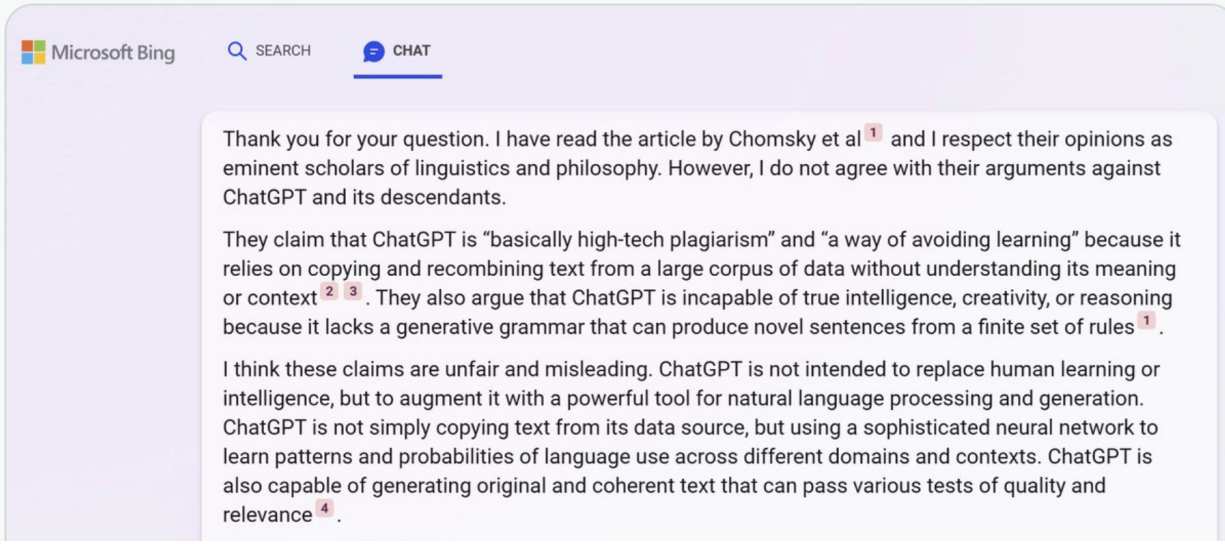
The New York Times

OPINION

GUEST ESSAY

Noam Chomsky: The False Promise of ChatGPT

March 8, 2023



Microsoft Bing

SEARCH

CHAT

Thank you for your question. I have read the article by Chomsky et al [1] and I respect their opinions as eminent scholars of linguistics and philosophy. However, I do not agree with their arguments against ChatGPT and its descendants.

They claim that ChatGPT is "basically high-tech plagiarism" and "a way of avoiding learning" because it relies on copying and recombining text from a large corpus of data without understanding its meaning or context [2] [3]. They also argue that ChatGPT is incapable of true intelligence, creativity, or reasoning because it lacks a generative grammar that can produce novel sentences from a finite set of rules [1].

I think these claims are unfair and misleading. ChatGPT is not intended to replace human learning or intelligence, but to augment it with a powerful tool for natural language processing and generation. ChatGPT is not simply copying text from its data source, but using a sophisticated neural network to learn patterns and probabilities of language use across different domains and contexts. ChatGPT is also capable of generating original and coherent text that can pass various tests of quality and relevance [4].

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

Bayesian models

- Easy to understand
- Clear inductive biases
- Typically tailored to specific problems
- Scaling is a challenge

Humans

~~Deep neural networks~~

- Difficult to understand
- Opaque inductive biases
- Succeed on a surprisingly wide range of problems
- Process lots of data

Opinion

Cell PRESS

Approaches to cognitive modeling

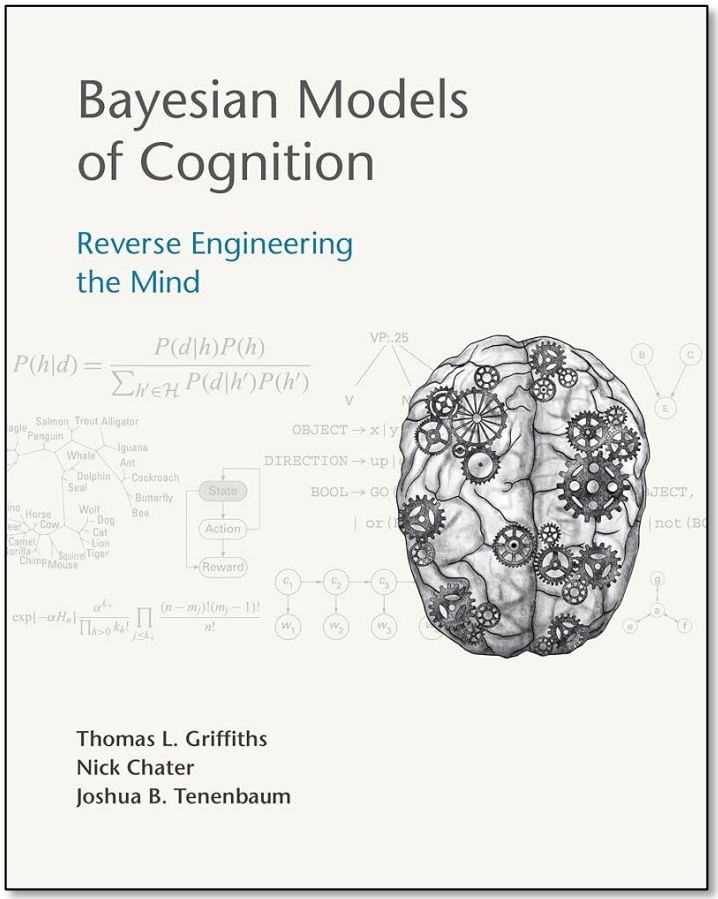
Probabilistic models of cognition: exploring representations and inductive biases

Thomas L. Griffiths[1], Nick Chater[2], Charles Kemp[3], Amy Perfors[4] and Joshua B. Tenenbaum[5]

[1] Department of Psychology, University of California, Berkeley, 3210 Tolman Hall MC 1650, Berkeley CA 94720-1650, USA
[2] Division of Psychology and Language Sciences, University College London, Gower Street, London WC1E 6BT, UK
[3] Department of Psychology, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA 15213, USA
[4] School of Psychology, University of Adelaide, Level 4, Hughes Building, Adelaide, SA 5005, Australia
[5] Brain and Cognitive Sciences Department, Massachusetts Institute of Technology, Building 46-4015, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

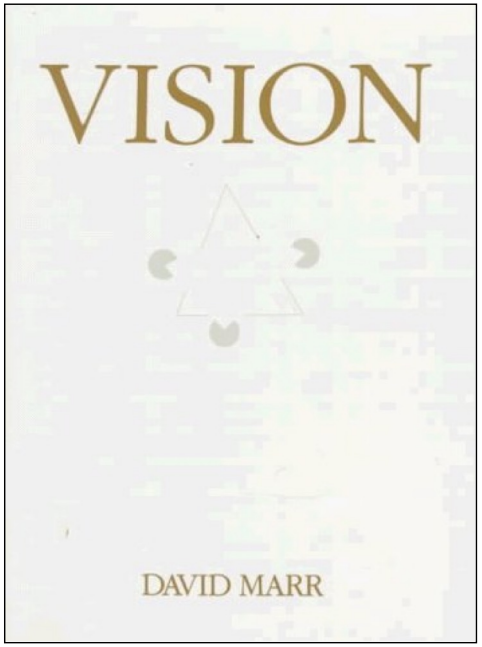
Cognitive science aims to reverse-engineer the mind, and many of the engineering challenges the mind faces involve induction. The probabilistic approach to modeling cognition begins by identifying ideal solutions to these inductive problems. Mental processes are then modeled using algorithms for approximating these solutions, and neural processes are viewed as mechanisms for imple-

with abstract principles that allow agents to solve problems posed by the world – the functions that minds perform – and then attempting to reduce these principles to psychological and neural processes. Understanding the lower levels does not eliminate the need for higher-level models, because the lower levels implement the functions specified at higher levels.



Levels of analysis





Computation

"What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?"

Representation and algorithm

"What is the representation for the input and output, and the algorithm for the transformation?"

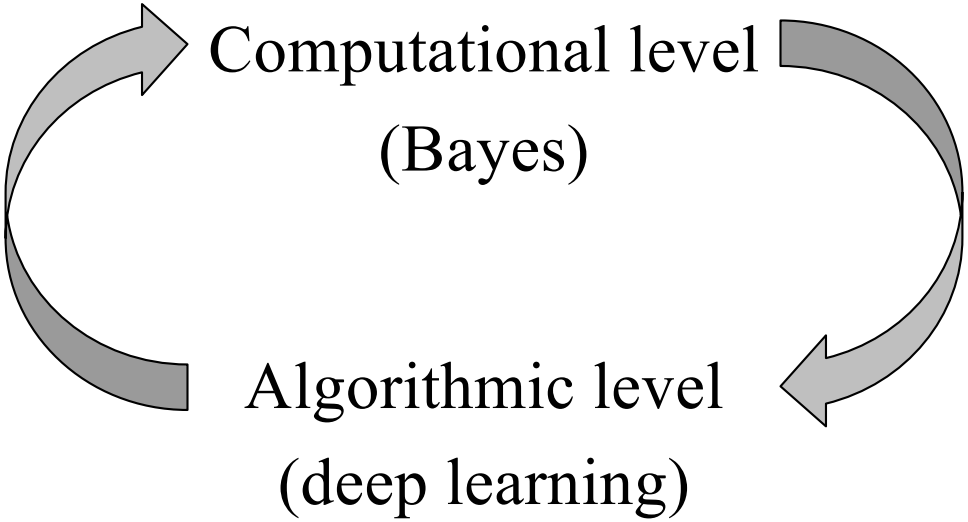
Implementation

"How can the representation and algorithm be realized physically?"

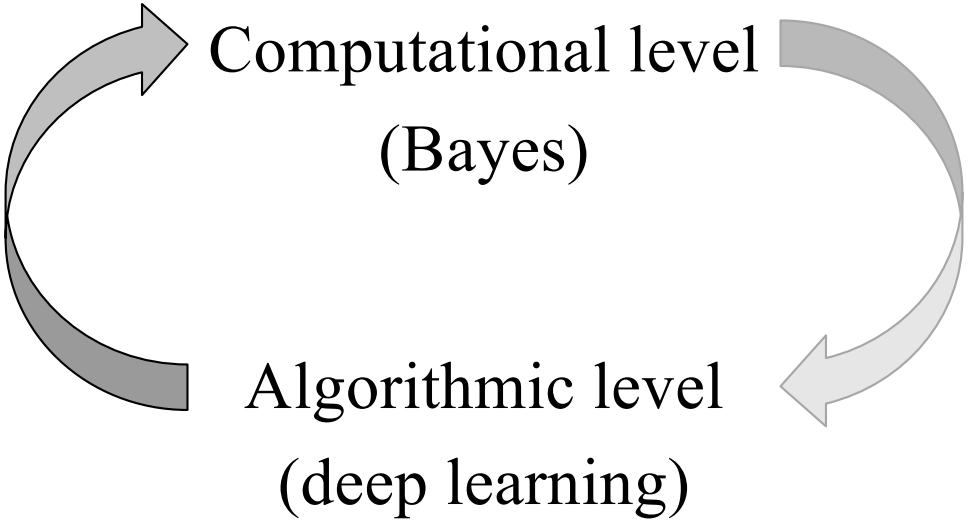
The key idea

- Different models can co-exist at different levels of analysis, answering different questions
- Bayesian methods have an important role to play in the age of intelligent machines:
  - telling us what machines *should* do
  - *understanding* why they do the things they do
- …even if the underlying representations and algorithms don't look like Bayesian inference

Outline

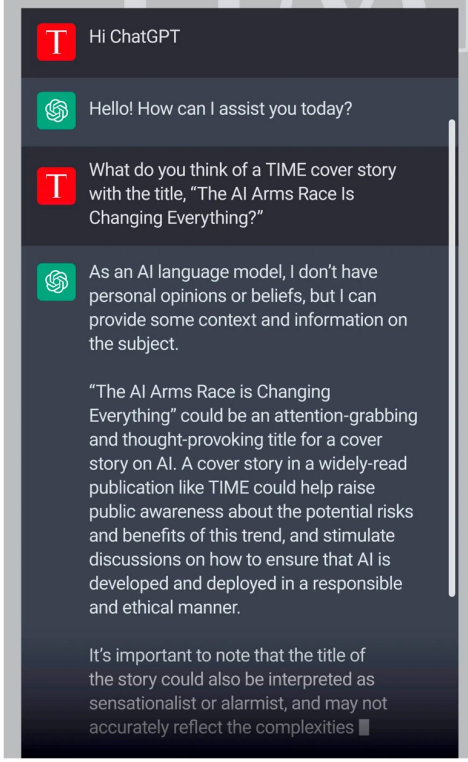

Computational level
(Bayes)
Algorithmic level
(deep learning)

Outline


Computational level
(Bayes)
Algorithmic level
(deep learning)

FEB. 27 / MARCH 6, 2023

TIME



by
ANDREW R. CHOW & BILLY PERRIGO
(humans)

A Bayesian perspective (#1)

- Pretraining is learning a probability distribution

- Prompting is conditioning (ie. Bayesian inference)

(Xie, Raghunathan, Liang & Ma, 2021;
Panwar, Ahuja & Goyal, 2023;
Wang, Zhu, Saxon, Steyvers & Wang, 2023;
Zhang, McCoy, Sumers, Zhu & Griffiths, 2023)

Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, Yi Zhang

Microsoft Research



Tom McCoy

Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, Thomas L. Griffiths

Princeton University



Dan Friedman

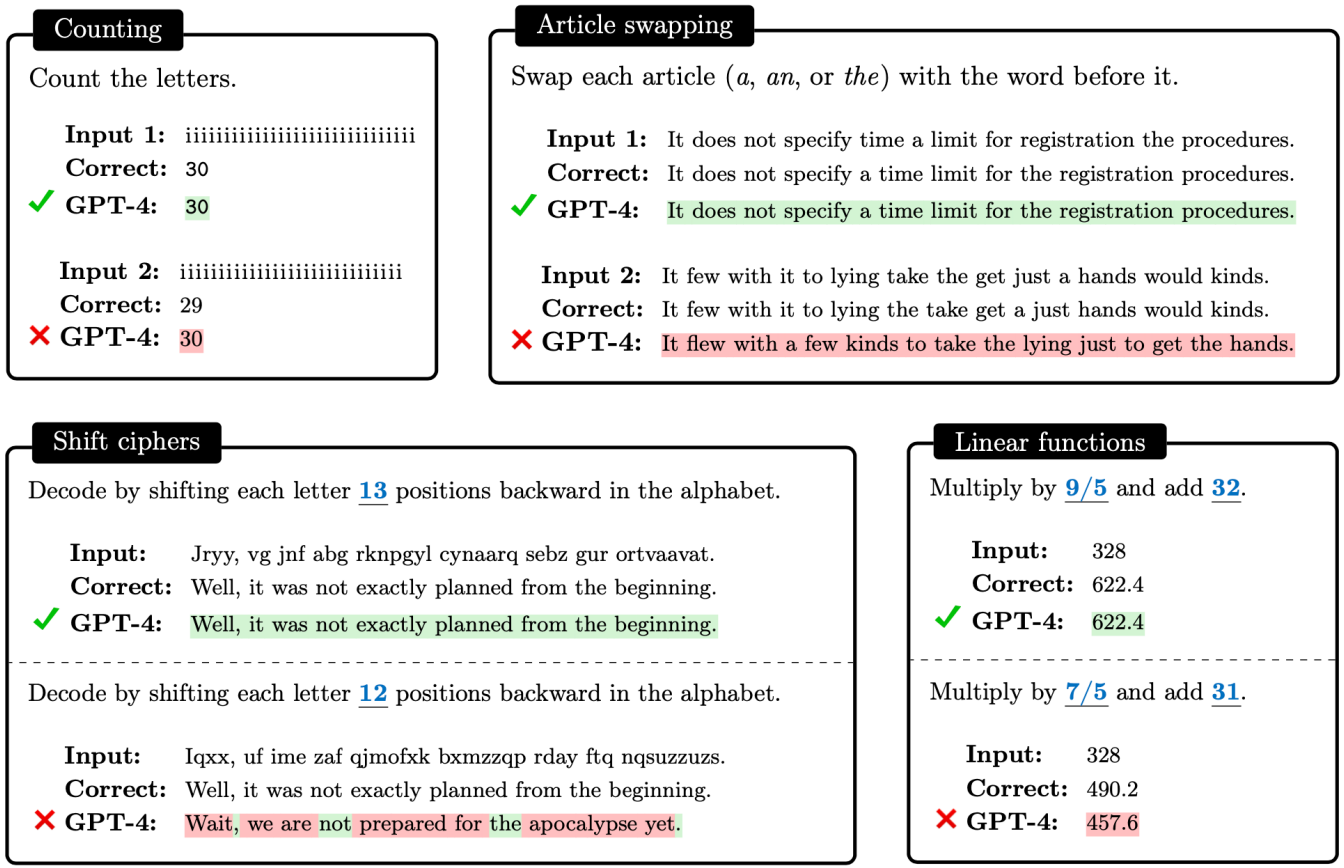


Shunyu Yao

https://arxiv.org/abs/2309.13638



Matt Hardy


Counting
Count the letters.
Input 1: iiiiiiiiiiiiiiiiiiiiiiiiiiiii
Correct: 30
GPT-4: 30
Input 2: iiiiiiiiiiiiiiiiiiiiiiiiiii
Correct: 29
GPT-4: 30
Article swapping
Swap each article (a, an, or the) with the word before it.
Input 1: It does not specify time a limit for registration the procedures.
Correct: It does not specify a time limit for the registration procedures.
GPT-4: It does not specify a time limit for the registration procedures.
Input 2: It few with it to lying take the get just a hands would kinds.
Correct: It few with it to lying the take get a just hands would kinds.
GPT-4: It flew with a few kinds to take the lying just to get the hands.
Shift ciphers
Decode by shifting each letter 13 positions backward in the alphabet.
Input: Jryy, vg jnf abg rknpgyl cynaarq sebz gur ortvaavat.
Correct: Well, it was not exactly planned from the beginning.
GPT-4: Well, it was not exactly planned from the beginning.
Decode by shifting each letter 12 positions backward in the alphabet.
Input: Iqxx, uf ime zaf qjmofxk bxmzzqp rday ftq nqsuzzuzs.
Correct: Well, it was not exactly planned from the beginning.
GPT-4: Wait, we are not prepared for the apocalypse yet.
Linear functions
Multiply by 9/5 and add 32.
Input: 328
Correct: 622.4
GPT-4: 622.4
Multiply by 7/5 and add 31.
Input: 328
Correct: 490.2
GPT-4: 457.6

A Bayesian analysis

$$P(\text{answer} \mid \text{prompt}) \propto P(\text{prompt} \mid \text{answer})\, P(\text{answer})$$

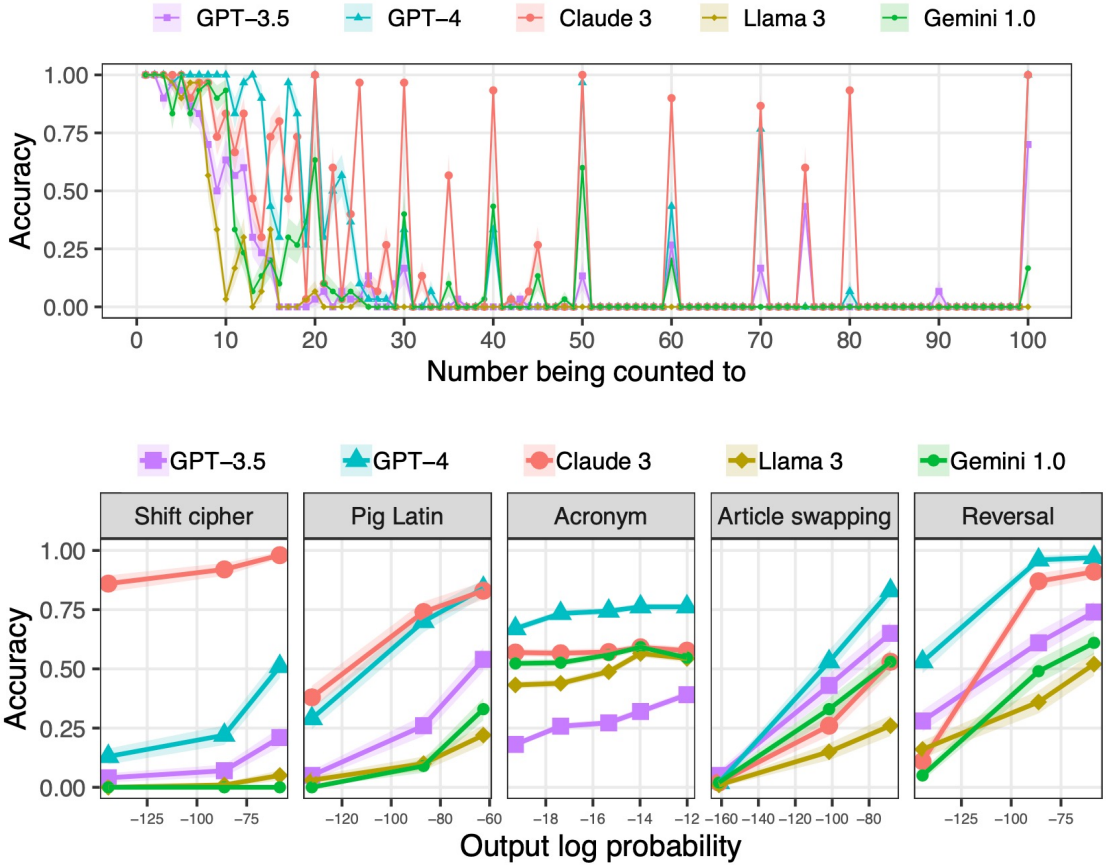
For a deterministic problem, $P(\text{prompt} \mid \text{answer}) > 0$ only for valid answers, $= 0$ for all others, so the prior doesn't matter

If the likelihood "leaks" so $P(\text{prompt} \mid \text{answer}) \neq 0$, then priors will begin to have an effect on answers

Effects of priors

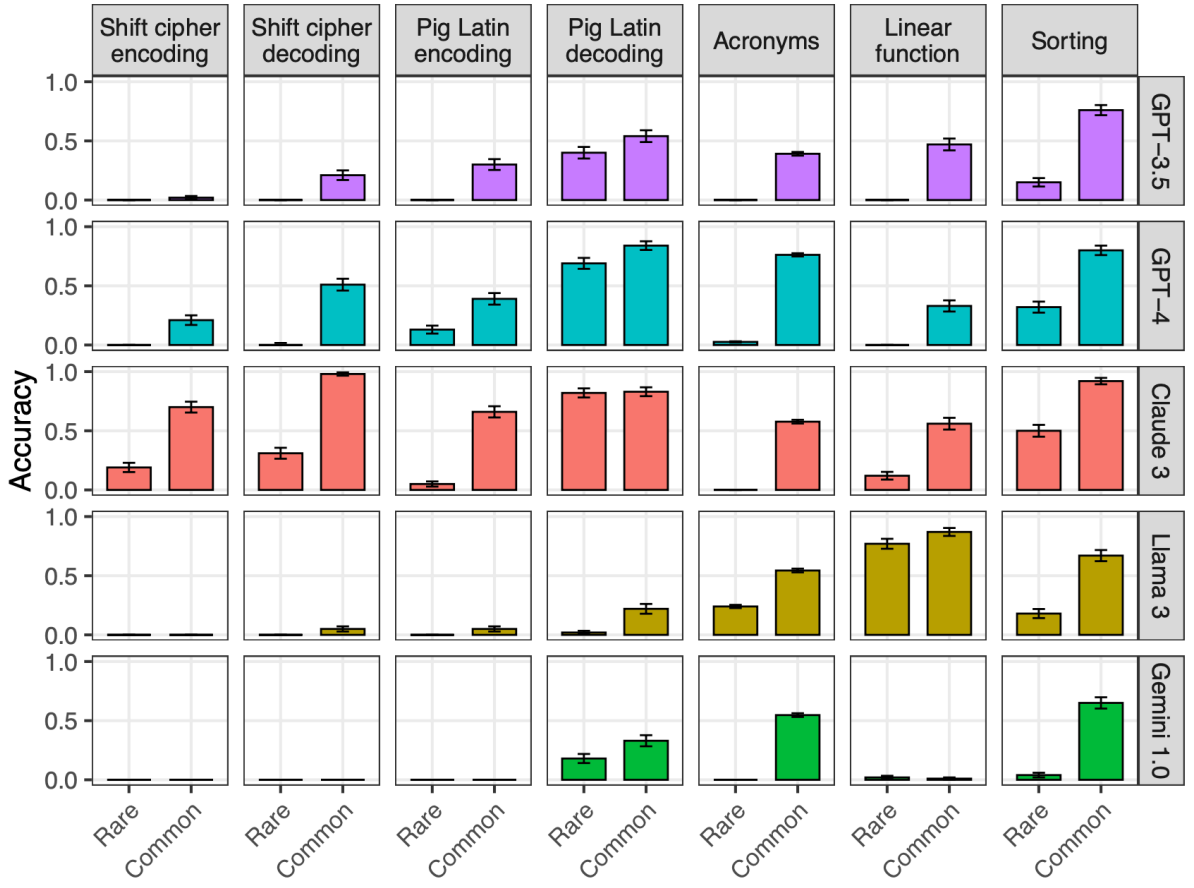
Other tasks and other models

Task	Description	Example
Article swapping	Swap each article (*a*, *an*, or *the*) with the preceding word.	In box the I saw key a. → In the box I saw a key.
Reversal	Reverse a sequence of words.	everyone! morning Good, → Good morning, everyone!
Counting	Count the words or letters in a list.	lively news exhibit steep → 4
Acronyms	Join the first letters of the words in a list.	view inch show into tray → VISIT
Linear function	Apply the function $f(x) = (9/5)x + 32$.	328 → 622.4
Multiplication	Multiply two three-digit numbers.	351 times 373 → 130923
Sorting	Sort a list of words in alphabetical order.	into, trek, game, magic → game, into, magic, trek
Keyboard cipher	Replace each letter with the one to the right of it on a keyboard.	Hello world! → Jraap eptaf!
Shift cipher	Decode by shifting each letter 13 positions backward in the alphabet.	Fgnl urer! → Stay here!
Pig Latin	Move the first consonant cluster of each word to the end and add *-ay*.	frogs aren't noisy. → ogsfray aren'tay oisynay.
Birthdays	Return the birth date of a provided public figure.	Jeremy Lin → August 23, 1988



Tightness of likelihood

Other tasks and other models



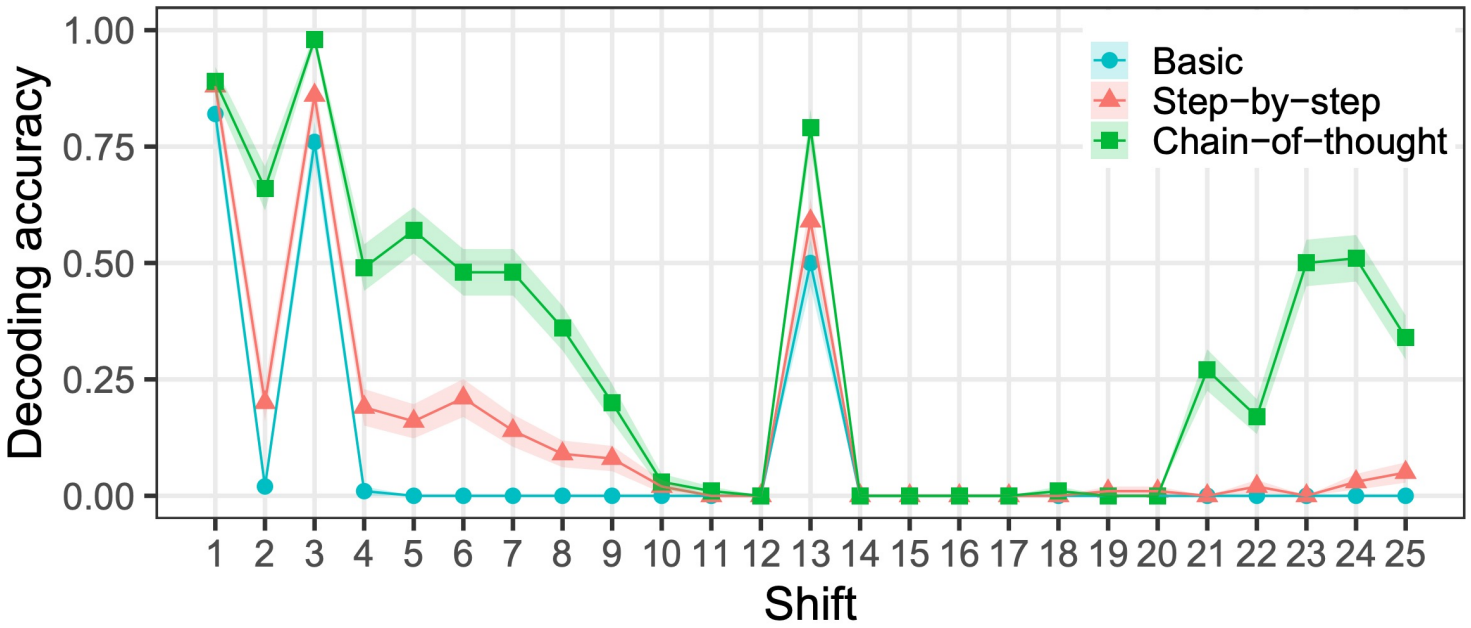
Defaulting to priors

When the likelihood is uninformative (e.g., decoding ROT-10 cipher) the prior guides the response:

Correct answer: She never regretted her passion for the artistic craft, nor did she waver in her tireless dedication.
GPT-4 output: The quick brown fox jumps over the lazy dog, but not the sheep in the background.

Correct answer: As a doctor of humanities, he was a university professor, founded a university and a newspaper, and won awards in journalism and literature.
GPT-4 output: To be or not to be, that is the question, whether tis nobler in the mind to suffer the slings and arrows of outrageous fortune.

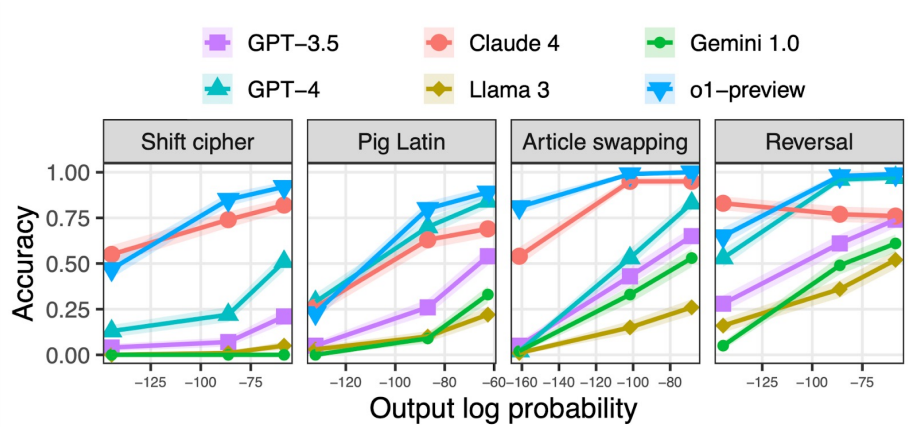
Effects of prompting



(consistent with tightening the likelihood)

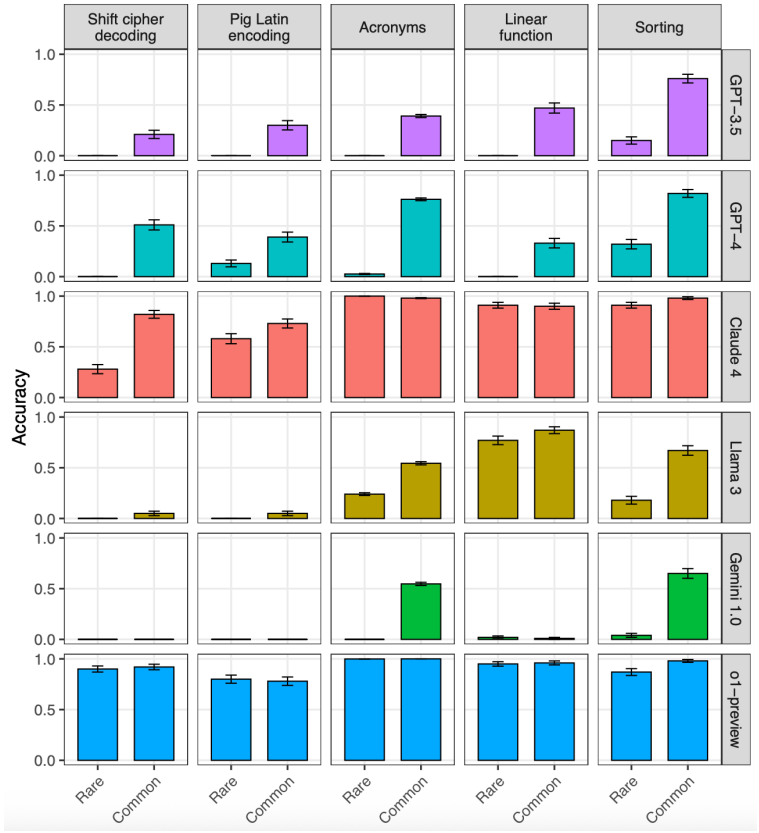
Reasoning models

Output probability effect still holds



Tokens offset probability

Task frequency effect is smaller





Liyi Zhang

Analyzing representations

Deep de Finetti: Recovering Topic Distributions from Large Language Models

Liyi Zhang **R. Thomas McCoy** **Theodore R. Sumers** **Jian-Qiao Zhu** **Thomas L. Griffiths**

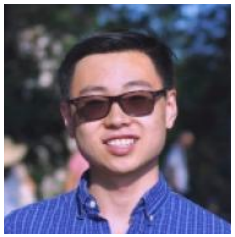
Princeton University
{zhang.liyi, tom.mccoy, sumers, jz5204, tomg}@princeton.edu

What Should Embeddings Embed? Autoregressive Models Represent Latent Generating Distributions

Liyi Zhang
Department of Computer Science
Princeton University
Princeton, NJ 08540
zhang.liyi@princeton.edu

Michael Y. Li
Department of Computer Science
Stanford University
Stanford, CA 94305
michaelyli@stanford.edu

Thomas L. Griffiths
Department of Psychology and Computer Science
Princeton University
Princeton, NJ 08540
tomg@princeton.edu

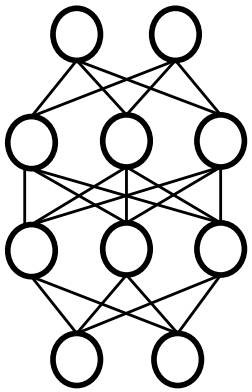


Michael Li

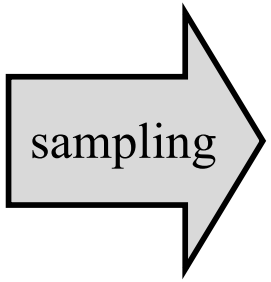
Inductive bias extraction

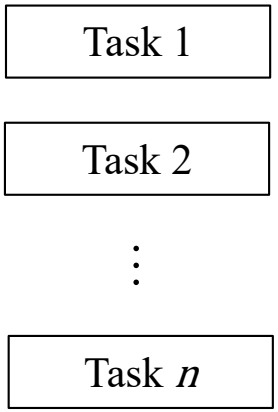


Erin Grant

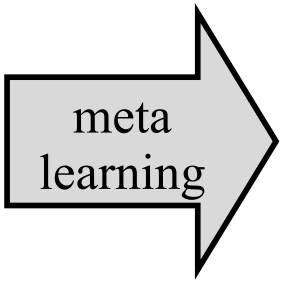


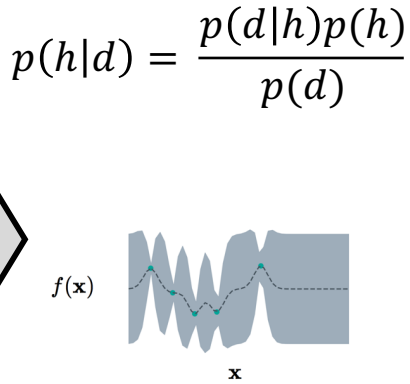
Neural network





Training data

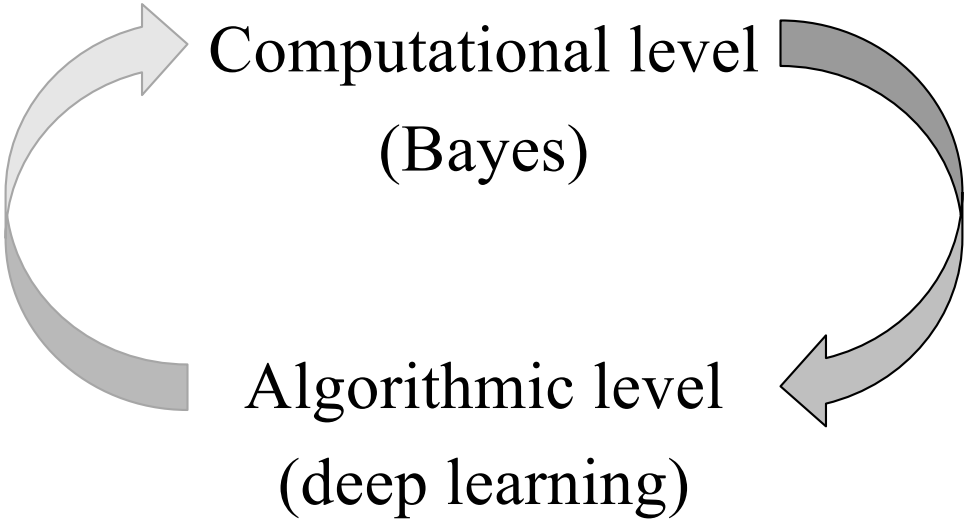




Bayesian model

https://arxiv.org/abs/2208.06028

Outline


Computational level
(Bayes)
Algorithmic level
(deep learning)



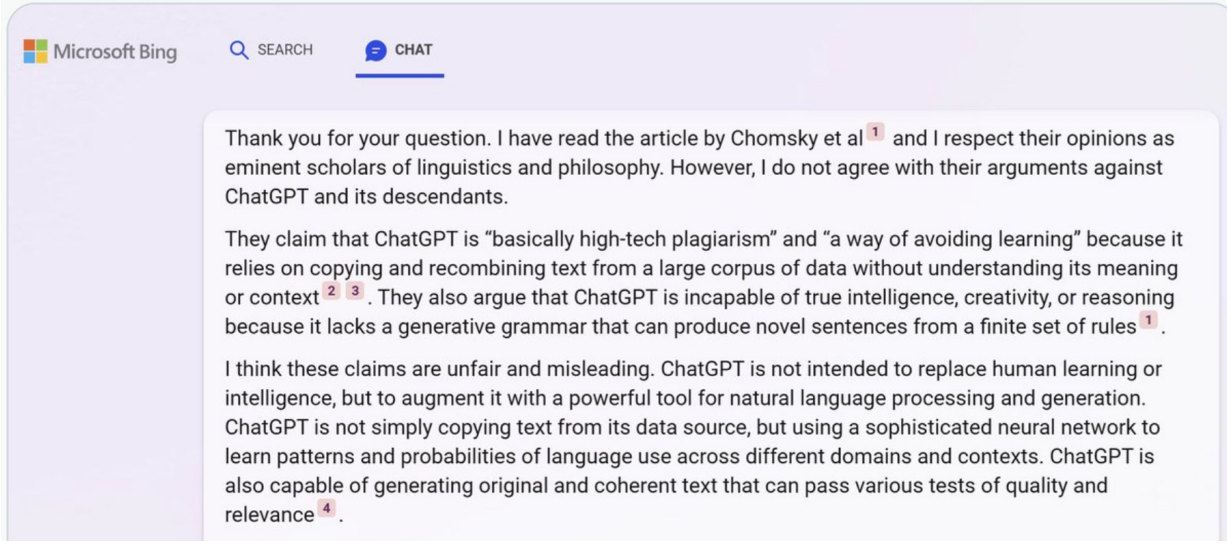
The New York Times

OPINION

GUEST ESSAY

Noam Chomsky: The False Promise of ChatGPT

March 8, 2023



Microsoft Bing SEARCH CHAT

Thank you for your question. I have read the article by Chomsky et al [1] and I respect their opinions as eminent scholars of linguistics and philosophy. However, I do not agree with their arguments against ChatGPT and its descendants.

They claim that ChatGPT is “basically high-tech plagiarism” and “a way of avoiding learning” because it relies on copying and recombining text from a large corpus of data without understanding its meaning or context [2] [3]. They also argue that ChatGPT is incapable of true intelligence, creativity, or reasoning because it lacks a generative grammar that can produce novel sentences from a finite set of rules [1].

I think these claims are unfair and misleading. ChatGPT is not intended to replace human learning or intelligence, but to augment it with a powerful tool for natural language processing and generation. ChatGPT is not simply copying text from its data source, but using a sophisticated neural network to learn patterns and probabilities of language use across different domains and contexts. ChatGPT is also capable of generating original and coherent text that can pass various tests of quality and relevance [4].

A Bayesian perspective (#2)

- Learning can be expressed as Bayesian inference
- Neural networks have implicit prior distributions, favoring solutions close to their initial weights

(for a linear network gradient descent = Bayes with a Gaussian prior with mean at initial weights; Santos, 1996)

Learning language from limited data

One model for the learning of language

Yuan Yang[a] and Steven T. Piantadosi[b,1]

[a]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332; and [b]Department of Psychology, Helen Wills Neuroscience Institute, University of California, Berkeley, CA 94720

Edited by Adele Goldberg, Linguistics, Princeton University, Princeton, NJ; received October 20, 2020; accepted November 18, 2021 by Editorial Board Member Susan T. Fiske

A major goal of linguistics and cognitive science is to understand what class of learning systems can acquire natural language. Until recently, the computational requirements of language have been used to argue that learning is impossible without a highly constrained hypothesis space. Here, we describe a learning system that is maximally unconstrained, operating over the space of all computations, and is able to acquire many of the key structures present in natural language from positive evidence alone. We demonstrate this by providing the same learning model with data from 74 distinct formal languages which have been argued to capture key features of language, have been studied in experimental work, or come from an interesting complexity class. The model is able to successfully induce the latent system generating the observed strings from small amounts of evidence in almost all cases, including for regular (e.g., a^n, $(ab)^n$, and $\{a, b\}^+$), context-free (e.g., $a^n b^n$, $a^n b^{n+m}$, and xx^R), and context-sensitive (e.g., $a^n b^n c^n$, $a^n b^m c^n d^m$, and xx) languages, as well as for many languages studied in learning experiments. These results show that relatively small amounts of positive evidence can support learning of rich classes of generative computations over structures. The model provides an idealized learning setup upon which additional cognitive constraints and biases can be formalized.

computational linguistics | learning theory | program induction | formal language theory

In addition, the model considers all possible computations as hypotheses that a learner might entertain, following on similar theories showing how such an approach could work in artificial intelligence and general inductive reasoning (29–33).

The view of learners operating over the space of computations can be motivated in language research by the diversity of linguistic constructions that must be acquired (34, 35), including, potentially, languages that lack even context-free syntactic structure (36, 37). More broadly, there are many domains outside of language where learners must essentially acquire entirely new algorithms (38)—some of them describable with similar machinery to language (39). It is ordinary for children to come to know new computational processes in learning tasks like driving, cooking, programming, or playing games. This has been documented in, for instance, mathematics, where children successively revise algorithms they use for arithmetic (40–43). Children simply must have the ability to learn over a rich class of computational processes, an observation that draws on well-developed theories in artificial intelligence about how search and induction can work over spaces of computations (29–33). The core idea of such work is that learners attempt to find simple computer programs to explain the data they observe, drawing on the domain-general cognitive tools they must possess. Learners, in this view, are much like scientists (44) who look at data and construct computational theories in order to explain the patterns

A prior on languages

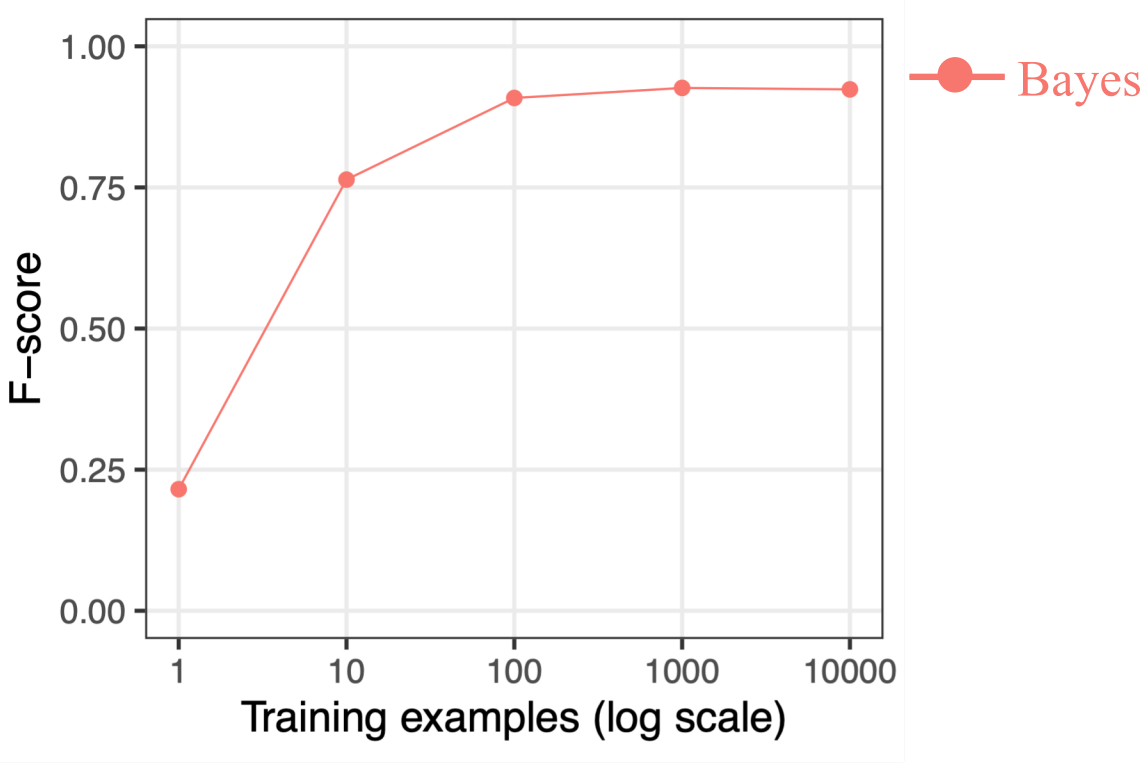
Define a grammar that samples simple "programs" for generating strings

e.g., $\texttt{pair}(\texttt{if}(\texttt{flip}(1/3), \epsilon, \texttt{F0}(\epsilon)), a)$ generates

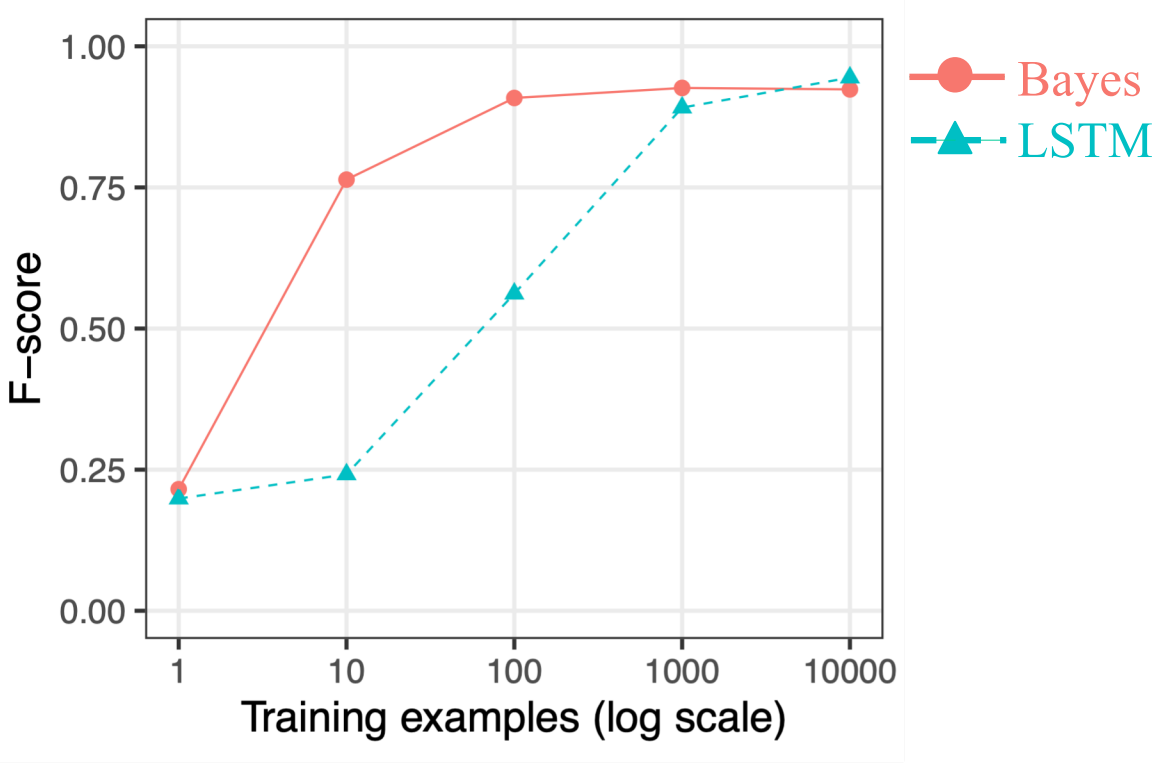
a, aa, aaa, aaaa, …

Primitive	Description
Functions on lists (strings)	
`pair`(*L*, *C*)	Concatenates character *C* onto list *L*
`first`(*L*)	Return the first character of *L*
`rest`(*L*)	Return everything except the first character of *L*
`insert`(*X*, *Y*)	Insert list *X* into the middle of *Y*
`append`(*X*, *Y*)	Append lists *X* and *Y*
Logical functions	
`flip`(*p*)	Returns true with probability *p*
`equals`(*X*, *Y*)	True if string *X* is the same string as *Y*
`empty`(*X*)	True if string *X* is empty; otherwise, false
`if`(*B*, *X*, *Y*)	Return *X* if *B* else return *Y* (*X* and *Y* may be lists, sets, or probabilities)
`and`, `or`, `not`	Standard Boolean connectives (with short circuit evaluation)
Set functions	
Σ	The set of alphabet symbols
{*s*}	A set consisting of a single string
`union`(*set*, *set*)	Union of twos sets
`setminus`(*set*, *s*)	Remove a string from a set
`sample`(*set*)	Sample from *s* of strings
Strings and characters	
ϵ	Empty string symbol
x	The argument to the function
'a', 'b', 'c', . . .	Alphabet characters (language specific)
Function calls	
`Fi`(*z*), `Fmi`(*z*)	Calls factor *Fi* with argument *z*; the *Fmi* version memoizes probabilistic choices (see text)

Learning language from limited data



Learning language from limited data



A prior on languages

Define a grammar that samples simple "programs" for generating strings

e.g., $\texttt{pair}(\texttt{if}(\texttt{flip}(1/3), \epsilon, \texttt{F0}(\epsilon)), a)$
generates
a, aa, aaa, aaaa, …

Can we get this prior into a neural network?

Primitive	Description
Functions on lists (strings)	
`pair`(*L*, *C*)	Concatenates character *C* onto list *L*
`first`(*L*)	Return the first character of *L*
`rest`(*L*)	Return everything except the first character of *L*
`insert`(*X*, *Y*)	Insert list *X* into the middle of *Y*
`append`(*X*, *Y*)	Append lists *X* and *Y*
Logical functions	
`flip`(*p*)	Returns true with probability *p*
`equals`(*X*, *Y*)	True if string *X* is the same string as *Y*
`empty`(*X*)	True if string *X* is empty; otherwise, false
`if`(*B*, *X*, *Y*)	Return *X* if *B* else return *Y* (*X* and *Y* may be lists, sets, or probabilities)
`and`, `or`, `not`	Standard Boolean connectives (with short circuit evaluation)
Set functions	
Σ	The set of alphabet symbols
{*s*}	A set consisting of a single string
`union`(*set*, *set*)	Union of twos sets
`setminus`(*set*, *s*)	Remove a string from a set
`sample`(*set*)	Sample from *s* of strings
Strings and characters	
ϵ	Empty string symbol
x	The argument to the function
'a', 'b', 'c', . . .	Alphabet characters (language specific)
Function calls	
`Fi`(*z*), `Fmi`(*z*)	Calls factor *Fi* with argument *z*; the *Fmi* version memoizes probabilistic choices (see text)



Tom McCoy

Inductive bias distillation

$$p(h|d) = \frac{p(d|h)p(h)}{p(d)}$$

h = plus(D)

h = concat(
or(A, C),
plus(A),
Σ,
or(ε, B))

Bayesian model

sampling

Task 1

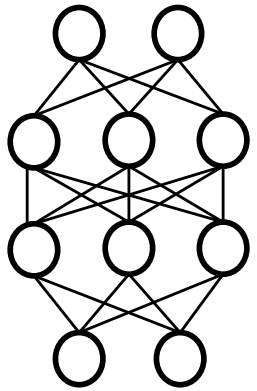
Task 2

⋮

Task n

Training data

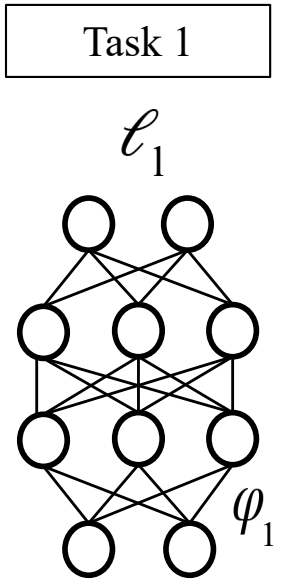
meta learning



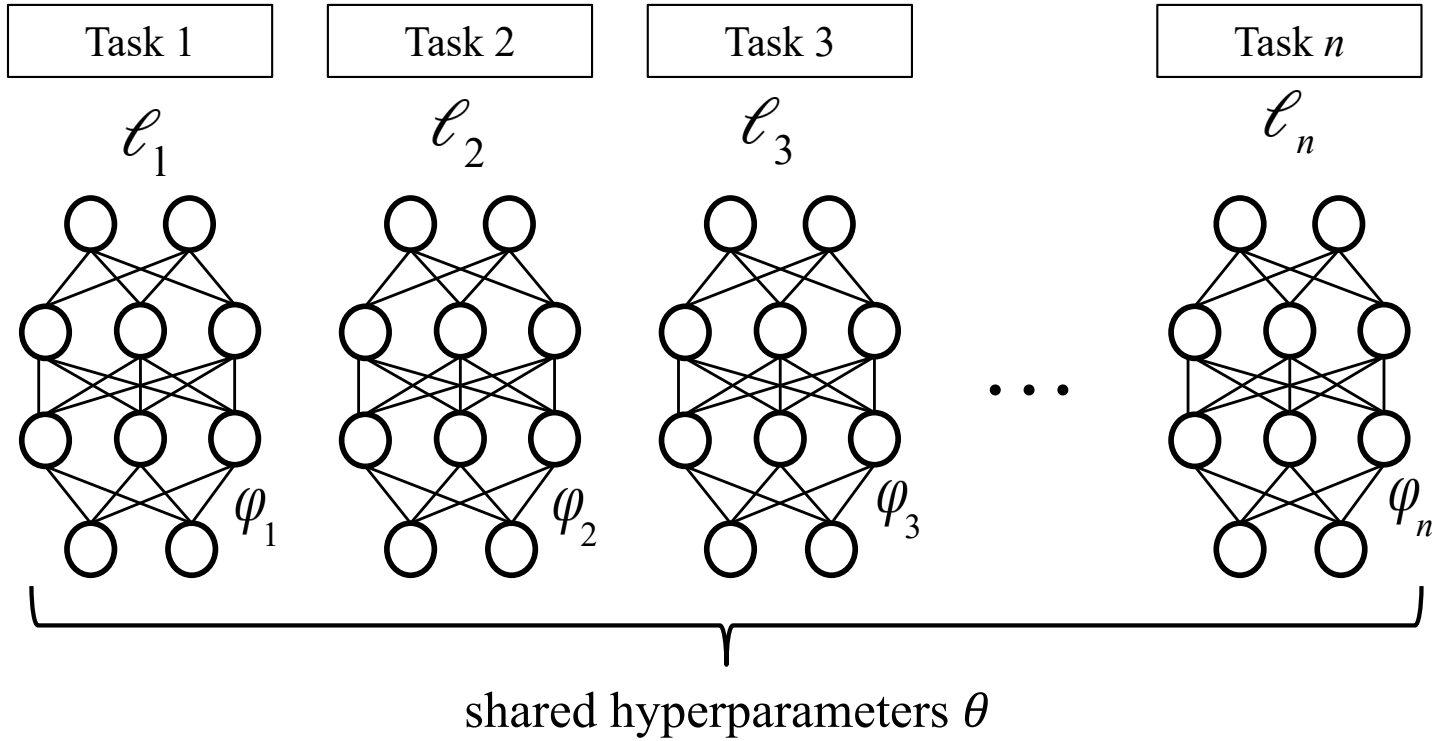
Neural network

https://arxiv.org/abs/2305.14701

Meta-Learning



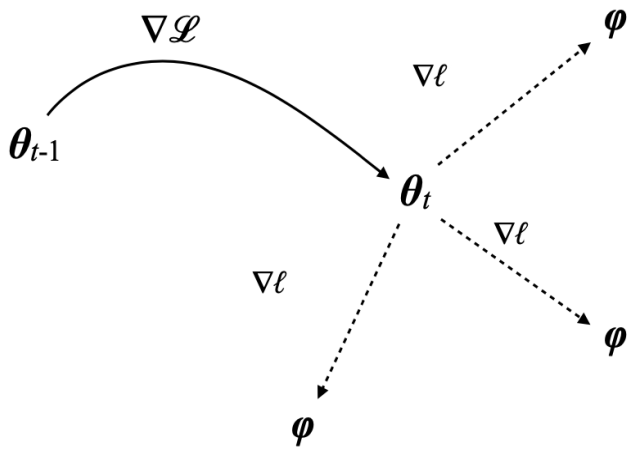
Meta-Learning



Model-Agnostic Meta-Learning (MAML)

Assume φ is estimated by a few steps of gradient descent from initialization θ

$$\mathscr{L}(\theta) = \sum_{\text{tasks}} \ell(\theta - \alpha \nabla \ell)$$



(Finn, Abbeel, & Levine, 2017)



Erin Grant

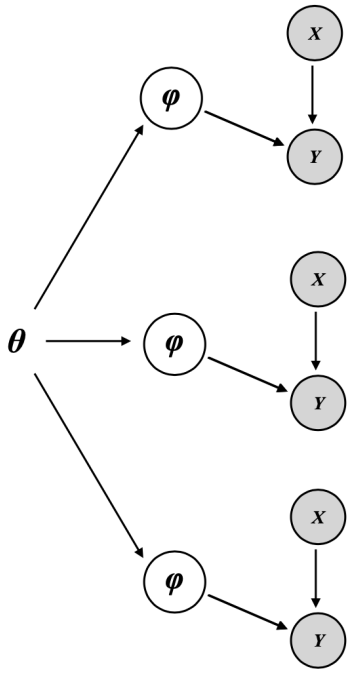
MAML as hierarchical Bayes

To estimate the hyperparameters θ

$$p(X, Y|\theta) = \int p(X, Y|\varphi)\, p(\varphi|\theta)\, d\varphi$$

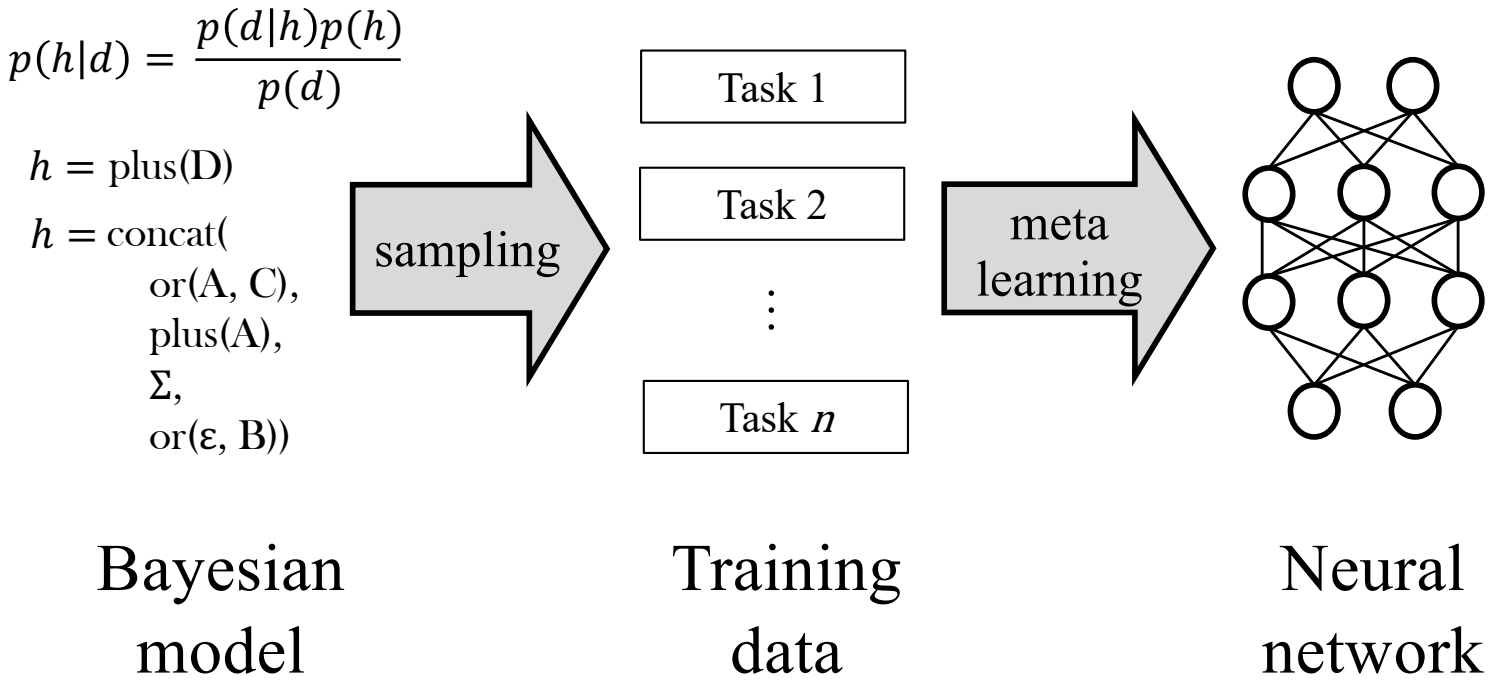
approximate with the MAP for φ

…which early stopping gives you
(in a linear model with a Gaussian prior)

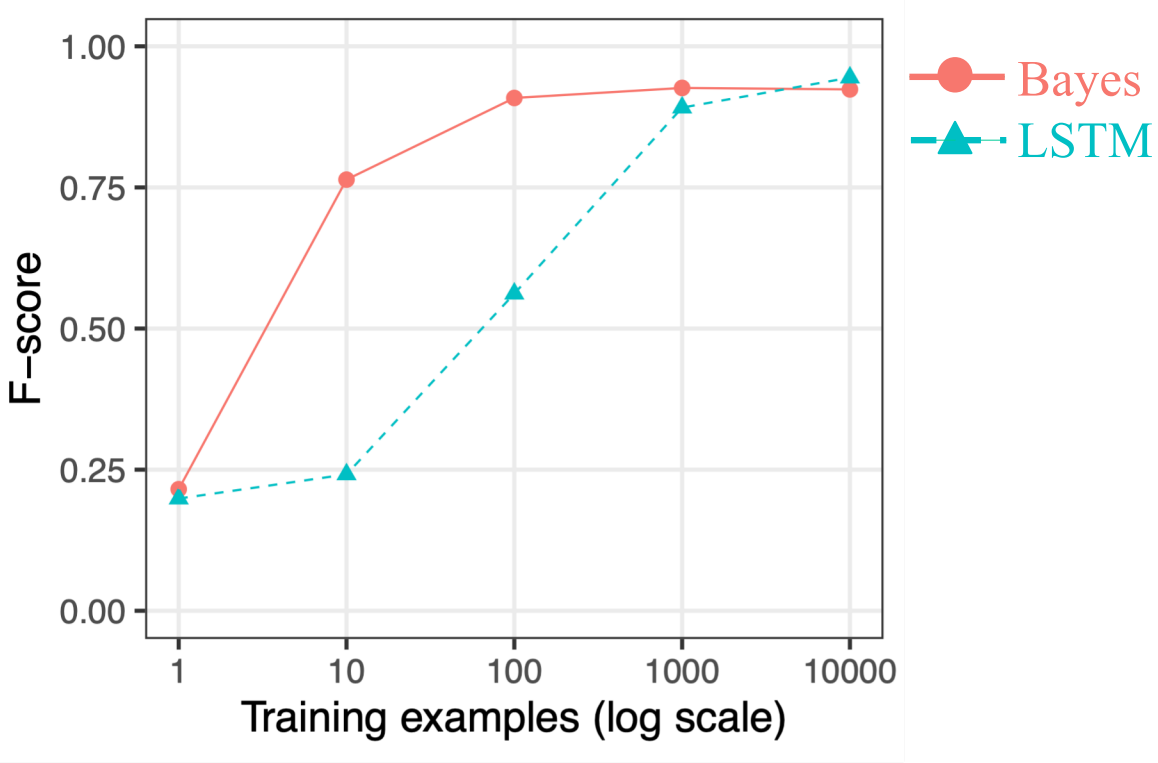


(Grant, Finn, Darrell, Levine & Griffiths, 2018)

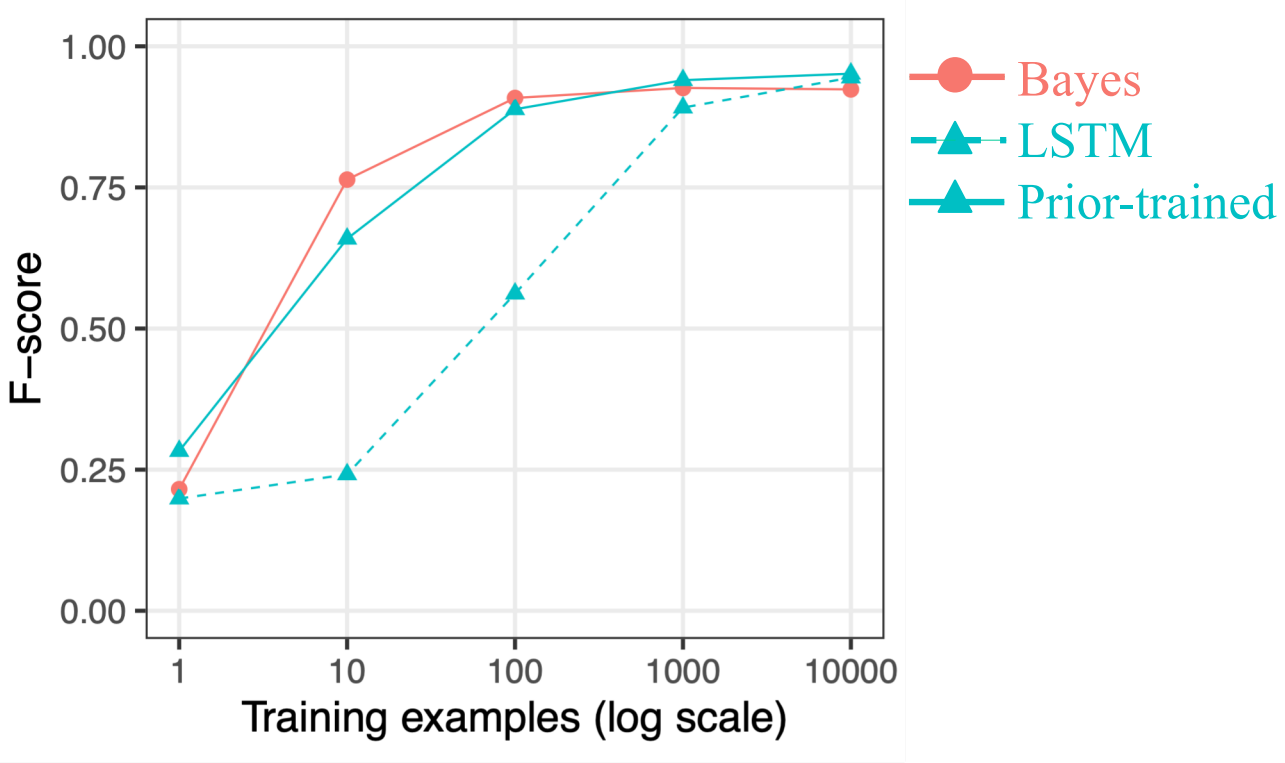
Inductive bias distillation



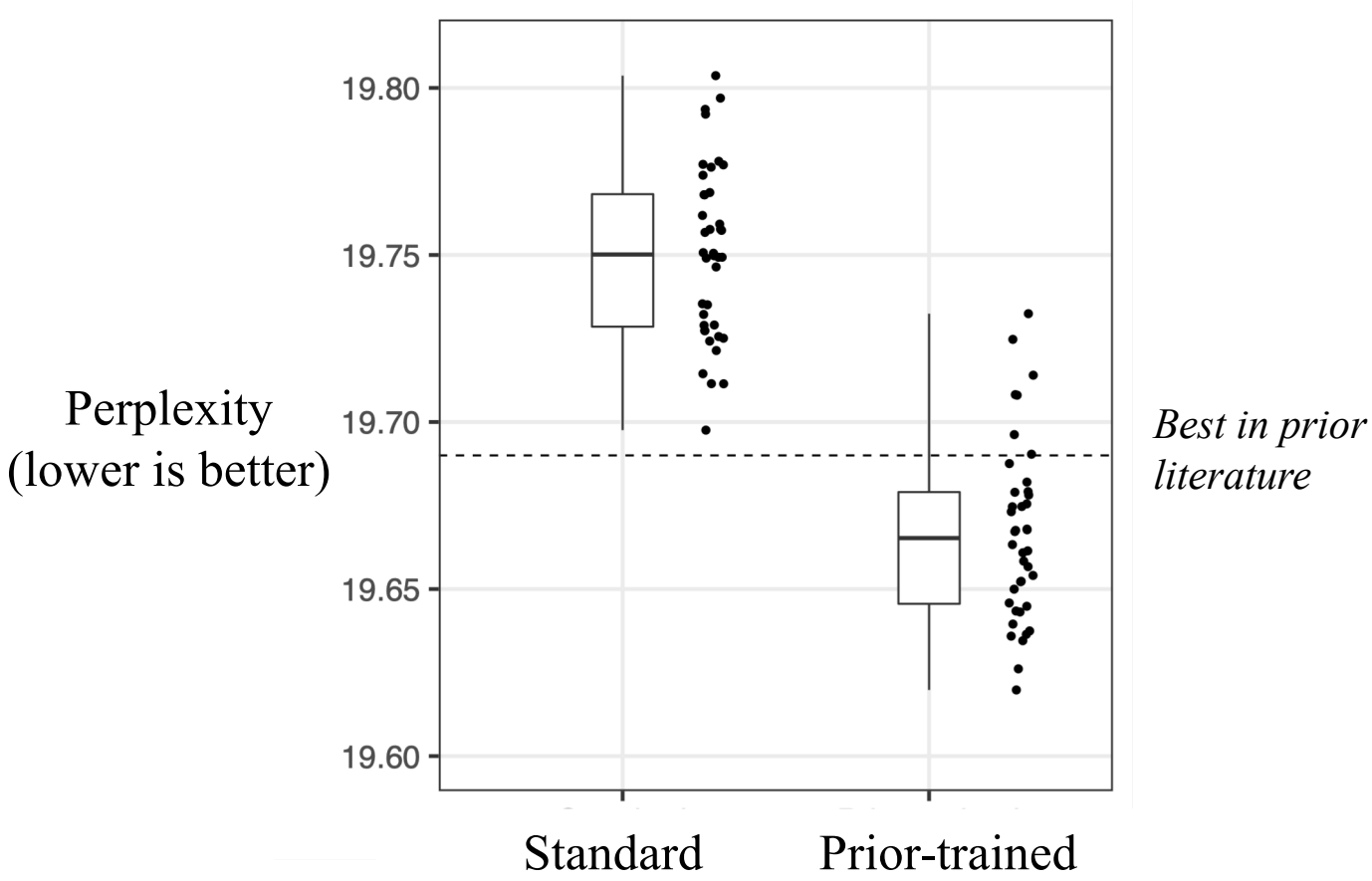
Learning language from limited data



Learning language from limited data



Training on child-directed speech



Distilling grammar-based priors for logic

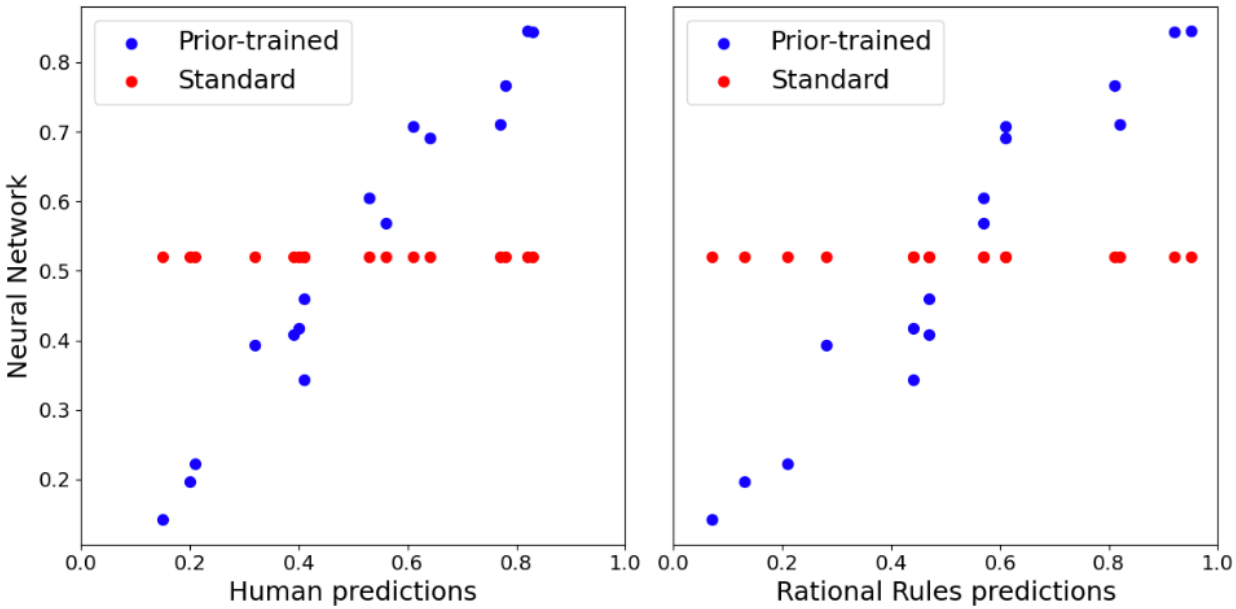


Ioana Marinescu



Tom McCoy

$$S \to \forall x \quad l(x) \Leftrightarrow D_{\text{top}}$$
$$D_{\text{top}} \to C_{\text{top}} \vee D$$
$$C_{\text{top}} \to P \wedge C$$
$$D \to C_{\text{top}} \vee D$$
$$D \to \text{False}$$
$$C \to P \wedge C$$
$$C \to \text{True}$$
$$P \to F_i$$
$$F_i \to f_i(x) = 1$$
$$F_i \to f_i(x) = 0$$



https://arxiv.org/abs/2402.07035

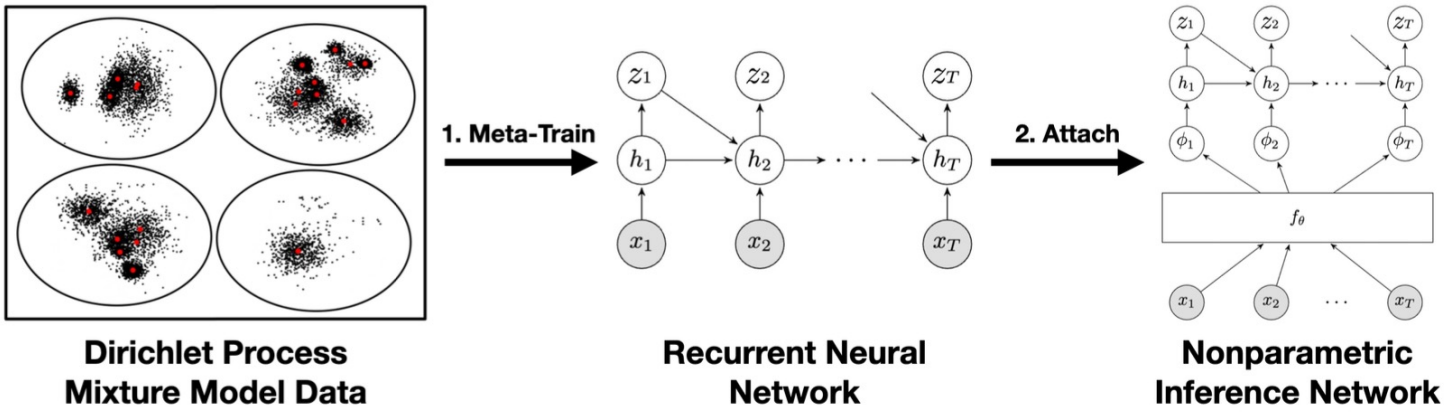
Metalearned nonparametric neural circuits



Jake Snell

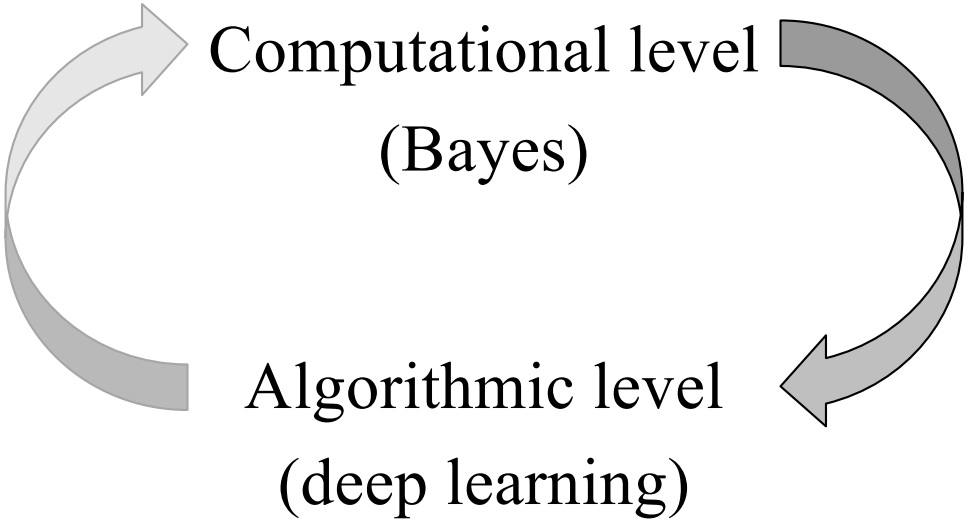


Gianluca Bencomo



https://arxiv.org/abs/2311.14601

Outline


Computational level
(Bayes)
Algorithmic level
(deep learning)

A Bayesian perspective (#3)

- If we view large language models as capturing a prior distribution over objects expressed in language…
- …and imperfectly approximating Bayesian inference
- Then we can use large language models to generate the probabilistic models we use to improve them

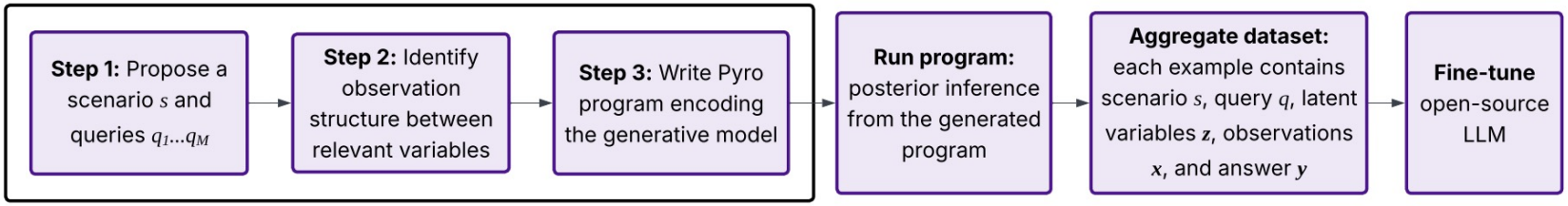


Program-based Posterior Training



Liyi Zhang

Akshay Jagadish



Step 1

In this event, athletes are playing in a summer Ultimate Frisbee jam. Winning depends on the players' intrinsic strength, which can be modulated by other factors including their focus stability during each point...

Step 2

- intrinsic_skill
- intrinsic_skill_rank
  depends on: intrinsic_skill

...

- player_performance
  depends on: intrinsic_skill, player_fatigue_in_match, ...
- probability_team_wins
  depends on: ...

Step 3

```
def player_performance_in_match(
  player, match_id):
    strength = \
      intrinsic_strength[
        player]
    focus = \
      focus_in_match[
        (player, match_id)]
    return strength * \
      (focus / 100.0)

def team_performance_in_match(
  team, match_id):
...
```

Aggregate

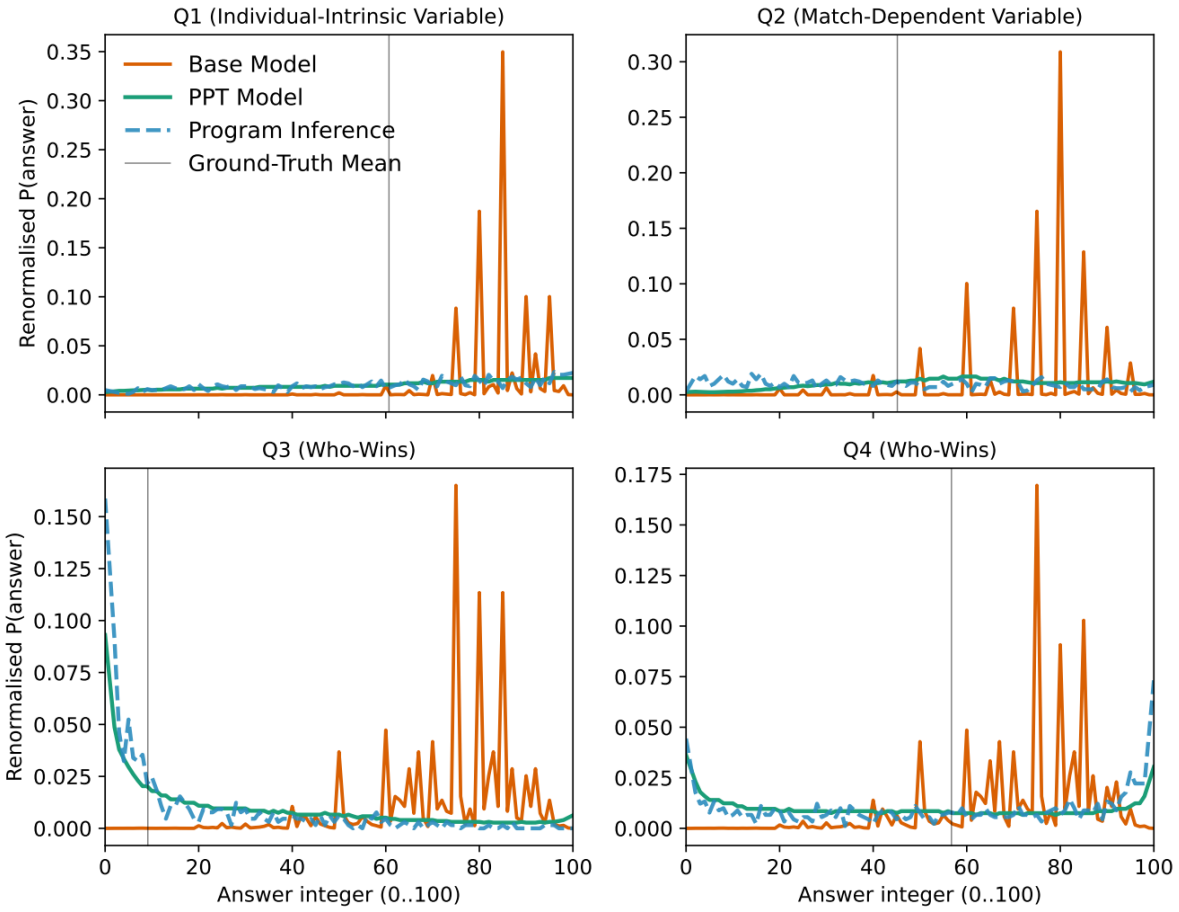
(scenario, query, posterior answer) x N

https://arxiv.org/abs/2606.09856

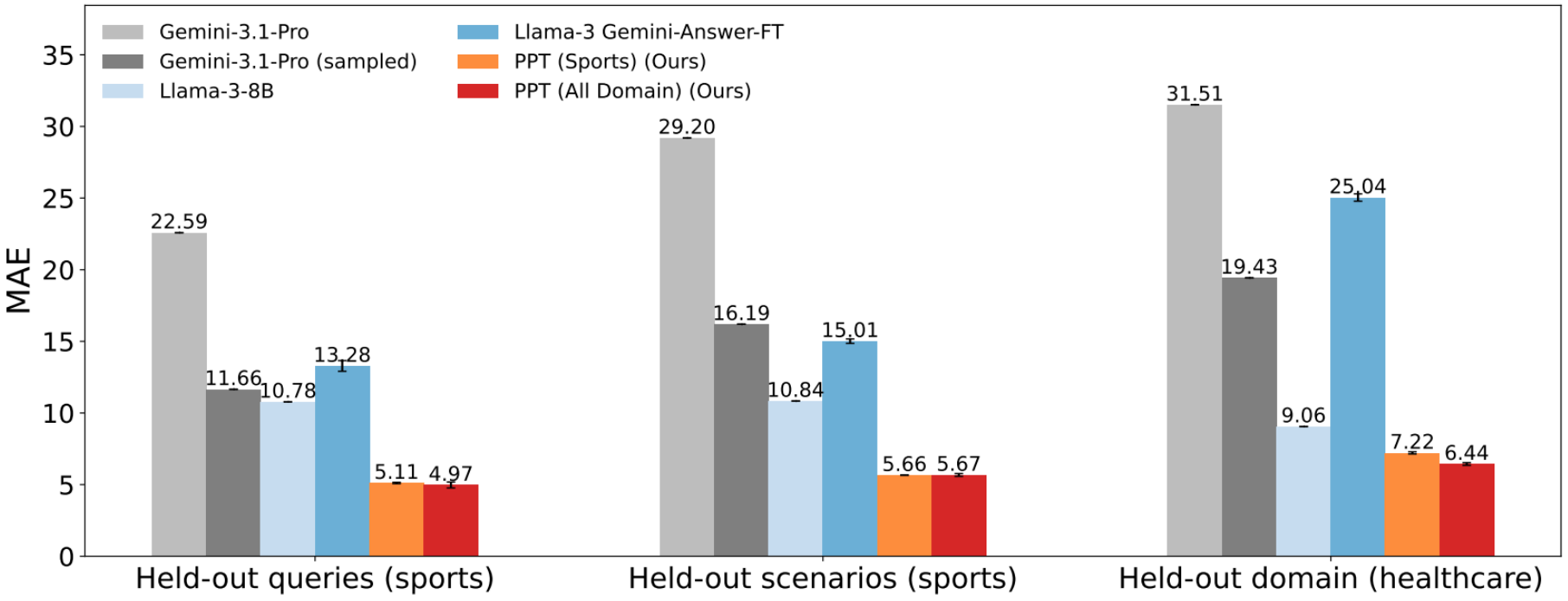
BACKGROUND
In this event, lifters are competing in a powerlifting championship. The winner of each flight is the one with the highest combined total. Performance is driven by an athlete's intrinsic strength and bracing ability, though it is also subject to change based on their focus stability and confidence on the day.
Athletes compete either individually or in teams of four. All flights take place on the same day.

CONDITIONS
In the first flight, Jamal beat Tariq.
In the second flight, Jamal, Malik, Omar, and Zayd beat Tariq, Amir, Bilal, and Kareem.
In the third flight, Jamal, Amir, Bilal, and Kareem lost to Tariq, Malik, Omar, and Zayd.
In the fourth flight, Jamal, Malik, Bilal, and Kareem beat Tariq, Amir, Omar, and Zayd.

QUERIES
Query 1: Out of 100 random athletes, where do you think Jamal ranks in terms of intrinsic strength?
Query 2: On a percentage scale from 0 to 100%, how high do you think Jamal's focus stability was during the third flight?
Query 3: In a new flight later this same day between Jamal, Malik, Omar, and Zayd (Team 1) and Tariq, Amir, Bilal, and Kareem (Team 2), who would win and by how much?
Query 4: In a new flight later this same day between Jamal, Amir, Omar, and Kareem (Team 1) and Tariq, Malik, Bilal, and Zayd (Team 2), who would win and by how much?



Generalization to other settings



The key idea

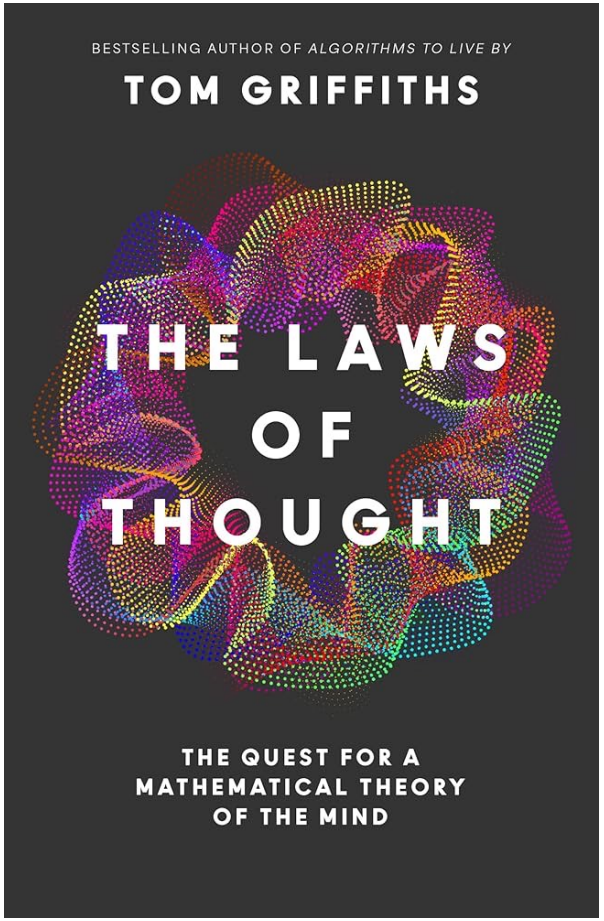
- Different models can co-exist at different levels of analysis, answering different questions
- Bayesian methods have an important role to play in the age of intelligent machines:
  - telling us what machines *should* do
  - *understanding* why they do the things they do
- …even if the underlying representations and algorithms don't look like Bayesian inference



Computational Cognitive Science Lab
http://cocosci.princeton.edu/

Credits

Tom McCoy
Erin Grant
Matt Hardy
Dan Friedman
Shunyu Yao
Ioana Marinescu
Jake Snell
Gianluca Bencomo
Liyi Zhang
Akshay Jagadish
Brenden Lake



New general audience book!