Physics of LLMs

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Ayan Sengupta

Theory

Computational / Statistical learning theories

- VC dimension
- PAC Learnability Theorem
- Generalization error bounds

Pros -- Rigorous, theorems

Cons:

- Too hard to understand, even by experts …
- Too idealistic assumptions
- Can analyze only shallow networks
- Theory rarely connect with practice

Ethology - *Science of animal behavior*

- Treat LLMs as monkeys
- Observe how they behave in different environments

Pros --

- Everyone can do theory!
- Easy to study large models
- Can derive interesting insights – CoT, "*think*" etc.

Cons --

- Too subjective, less rigorous
- may lead to false claims

Slide credit - https://www.youtube.com/watch?v=YSHzKmEianc&t=2765s

Issues with Ethology

1. **Studying a specific model** – learnings may not generalize to other LLMs
2. **Data leaks** – no control over the training data
3. **Black box testing** – no revelation of inner mechanics
4. **Success / Failure attribution** –
   Is failure due to
   1. Lack of knowledge or ability to extract knowledge ?
   2. Lack of ability to reason over the knowledge ?

   Is success due to
   1. Cheating – pre-train data directly contains the compound fact
   2. Ability of the model to extract and logically infer

Alternative to Ethology?

Physics of Language Models

Computational / Statistical learning theories

- VC dimension
- PAC Learnability Theorem
- Generalization error bounds

1. Decompose "intelligence" into multiple building blocks (Structure / Knowledge / Reasoning)
2. Study in controlled, idealized environment (control data, avoid data leaks)
3. Highly repeatable experiments (user 100M-size models, derive universal laws)
4. Probing techniques to see the inner workings

Ethology - *Science of animal behavior*

Physics of Language Models

Physics of Language Models

Home

Part 1: Hierarchical Language Structures

Part 2: Grade-School Math

Part 3: Knowledge

Part 4: Architecture Design

FAQ

- Project page
- Youtube Channel https://www.youtube.com/@Zeyuan-AllenZhu
  - 2024 ICML tutorial
  - Dedicated Online Talks on each part

Slides credit – (Almost) all slides are taken from the Youtube videos

Physics of Language Models: Part 3, Knowledge

Zeyuan Allen-Zhu (Meta)

Yuanzhi Li (MBZUAI)

Physics of Language Models: Part 3.1
Knowledge Storage and Extraction

Result 1 — mixed training $\Longrightarrow$ knowledge extraction
mixed training means "add instruct-finetune data to pretrain stage"

Result 2-3 — instruct finetune $\not\Rightarrow$ knowledge extraction
unless pretrain data is augmented (e.g. diversified writing styles)

Result 4-5 — two probing techniques to explain why this happens
can detect *how* and *where* knowledge is stored in an LLM

Result 6 — knowledge on "celebrity" helps "minority"
augmenting pretrain data for some knowledge helps others

Result 7 — encoder models like BERT $\not\Rightarrow$ knowledge extraction
discover and explain why this happens

Physics of Language Models: Part 3.2, Knowledge Manipulation

Physics of Language Models: Part 3.2 – Knowledge Manipulation

Physics of Language Models: Part 3.2 – Knowledge Manipulation

Was Donald Trump born in an even day?

No, Donald Trump was born on June 14, 1946, which is an odd day.

Do you know what odd means?

Yes, I apologize for the oversight. You are correct. The number 14 is even. So, Donald Trump was born on an even day, June 14, 1946. Thank you for pointing that out.

Even proprietary models like GPT-4 fails to respond to knowledge manipulation questions

Yes or no, was Donald Trump born earlier than Joe Biden?

Yes.

Tell me why?

Donald Trump was born on June 14, 1946, while Joe Biden was born on November 20, 1942. Therefore, Joe Biden was born earlier than Donald Trump by almost four years.

With CoT, however, GPT-4 can answer them correctly

Control Setup – Training Data

0. *First, middle, and last names from 400, 400, and 1000 choices*
1. *Birth years range from 1900 to 2099, months 1-12, days 1-28.*
2. *Birth cities from 200 US cities*
3. *Universities from a list of 300 US institutions.*
4. *Majors are from 100 college disciplines*
5. *Employers are chosen from 263 companies*
6. ***Work location uniquely determined by employer***

bioS dataset

Anya Briar Forger was born on October 2, 1996. She spent her early years in Princeton, NJ. She received mentorship and guidance from faculty members at Massachusetts Institute of Technology. She completed her education with a focus on Communications. She had a professional role at Meta Platforms. She was employed in Menlo Park, CA.

- Six sentences describing 6 attributes
- Each sentence is drawn from 50 templates
- (for now) Fix the sentence ordering

bioR dataset

Anya Briar Forger is a renowned social media strategist and community manager. She is currently working as a Marketing Manager at Meta Platforms. She completed her graduation from MIT with a degree in Communications. She was born on 2nd October 1996 in Princeton, NJ and was brought up in the same city. She later moved to Menlo Park in California to be a part of Facebook's team. She is an avid reader and loves traveling.

rewritten by LLaMA 1 / LIMA

Knowledge Partial and Dual Retrieval

bioS data

Anya Briar Forger was born on October 2, 1996. She spent her early years in Princeton, NJ. She received mentorship and guidance from faculty members at MIT. She completed her education with a focus on Communications. She had a professional role at Meta Platforms. She was employed in Menlo Park, CA.

knowledge extraction

What is the birth date of Anya Briar Forger?
Which university did Anya Briar Forger study?
Which company did Anya Briar Forger work for?
What is the birth city of Anya Briar Forger?
What major did Anya Briar Forger study?
Where did Anya Briar Forger work?

partial retrieval

What is the birth day in the month for Anya Briar Forger? Answer: 2
What is the birth year for Anya Briar Forger? Answer: 1996

Knowledge Partial and Dual Retrieval

1. Where was Anya Briar Forger born and which company did this this person work for? *Princeton, NJ; Meta Platforms.*
2. Which company did Anya Briar Forger work for and where was this person born? *Meta Platforms; Princeton, NJ.*
3. Which university and what major did Anya Briar Forger study? *Massachusetts Institute of Technology; Communications.*
4. What major and which university did Anya Briar Forger study? *Communications; Massachusetts Institute of Technology.*
5. Where and which company did Anya Briar Forger work for? *Menlo Park, CA; Meta Platforms.*
6. Which company and where did Anya Briar Forger work for? *Meta Platforms; Menlo Park, CA.*

Knowledge Partial and Dual Retrieval

	QA bday	QA byear	QA bcity+cname	QA cname+bcity	QA univ+major	QA major+univ	QA cname+ccity	QA ccity+cname	QA bdate	QA bcity	QA univ	QA major	QA cname	QA ccity
baseline	3.6	0.5	0.0	0.0	0.0	0.0	0.4	0.4	0.0	0.5	0.3	1.0	0.4	13.7
bioS single	37.2	15.6	0.2	0.3	0.2	0.2	1.0	1.1	33.5	6.3	2.3	4.0	1.1	13.8
bioS single + fullname	58.5	43.5	27.6	29.1	41.1	36.3	55.2	11.2	56.2	58.8	63.0	55.7	50.5	14.1
bioS single + permute1	8.5	2.6	0.2	0.2	0.3	0.3	3.6	1.9	0.5	3.3	2.4	5.0	3.5	13.7
bioS single + permute2	62.1	16.6	27.1	27.1	32.1	31.9	52.5	51.2	57.3	48.3	53.1	55.0	51.8	58.3
bioS single + permute5	67.6	18.8	49.1	49.1	42.0	42.7	94.9	91.8	56.4	57.7	58.3	64.9	90.5	97.7
bioS single + permute1 + fullname	43.9	30.5	9.5	10.4	12.3	11.0	32.0	28.5	26.6	29.3	36.9	31.1	31.4	37.9
bioS single + permute2 + fullname	70.1	65.0	54.3	53.9	43.0	44.2	91.1	90.2	69.0	60.6	64.2	64.0	87.9	95.0
bioS single + permute5 + fullname	82.4	68.7	65.5	63.9	49.7	54.6	95.9	96.6	83.7	67.8	72.6	69.1	93.0	98.6
bioS multi2	90.6	47.1	5.2	3.1	19.8	9.6	3.4	2.6	100	71.7	33.1	26.1	5.2	14.0
bioS multi2 + fullname	97.6	77.1	94.0	97.5	93.8	94.3	98.5	19.4	100	97.7	89.5	97.6	91.3	35.3
bioS multi2 + permute	95.7	46.5	85.2	88.2	93.8	95.1	87.9	80.4	99.3	98.7	89.8	96.7	83.3	83.5
bioS multi2 + permute + fullname	96.2	67.2	95.1	97.0	95.3	94.3	97.4	96.1	100	98.8	91.3	98.1	93.7	97.8
bioS multi5	85.0	37.3	6.9	4.6	31.5	16.7	9.7	3.6	100	50.8	30.9	43.5	10.2	13.8
bioS multi5 + fullname	97.4	68.0	94.0	95.4	92.0	94.6	94.9	17.4	100	98.6	88.4	96.1	91.9	26.8
bioS multi5 + permute	82.3	20.4	96.5	97.6	94.8	94.6	97.0	96.9	100	99.0	91.3	97.7	95.1	98.7
bioS multi5 + permute + fullname	97.6	76.7	95.6	98.6	95.1	95.1	98.7	97.7	100	98.7	90.6	97.9	93.7	99.0

Result 1 (Figure 3 middle). *Dual retrieval is generally easy when both tasks are. However, if there is a causal and spatial relationship between pieces of knowledge, their order may matter.*

Result 2 (Figure 3 left). *Even if an attribute (e.g., October 2, 1996) can be perfectly extracted, partially retrieving only its later tokens (e.g., the year 1996) may still be poor.*

Knowledge Classification & Comparison

1. Was Anya Briar Forger born in an even month? Answer: *Yes*.
2. What is Anya Briar Forger's birth month mod 6? Answer: *4*.
3. What is Anya Briar Forger's birth month in numerics? Answer: *10*.

Used modular arithmetic with p = 2, 6, 12

1. Was Anya Briar Forger born in a month in a year later than Sabrina Eugeo Zuberg? [Yes/No].
2. What is Anya Briar Forger's birth month minus Sabrina Eugeo Zuberg's birth month? [-11..11].
3. Did Anya Briar Forger major in a field luckier than Sabrina Eugeo Zuberg? [Yes/No].
4. How luckier is Anya Briar Forger's major compared with Sabrina Eugeo Zuberg's major? [-99..99]

Knowledge Classification & Comparison

field	task	#train individuals	baseline	BIO pretrained model				QA finetuned model			
				trained w/o hint	trained with hint			trained w/o hint	trained with hint		
				test acc	test acc (with hint)	test acc (w/o hint)	hint acc	test acc	test acc (with hint)	test acc (w/o hint)	hint acc
birthmonth	classify %2	(2.5k)	50.0	60.4	77.8	65.2	64.5	61.9	80.4	65.2	69.1
birthmonth	classify %2	(5k)	50.0	67.3	87.3	72.7	80.3	68.0	89.5	72.8	83.9
birthmonth	classify %2	(10k)	50.0	75.9	94.2	80.3	91.0	76.4	95.0	79.9	92.8
birthmonth	classify %2	(25k)	50.0	86.4	98.6	91.1	97.8	87.1	98.8	90.9	98.4
birthmonth	classify %2	(50k)	50.0	95.3	99.5	97.5	99.2	96.3	99.7	97.5	99.5
birthmonth	ranking	(2.5k)	54.2	53.7	65.4	59.6	44.2	57.3	65.5	57.6	44.9
birthmonth	ranking	(5k)	54.2	59.2	75.5	63.4	63.6	62.5	75.1	63.1	62.6
birthmonth	ranking	(10k)	54.2	65.4	87.7	67.0	82.7	65.9	88.9	66.3	83.9
birthmonth	ranking	(25k)	54.2	75.6	96.7	75.8	95.4	78.3	97.4	72.5	96.3
birthmonth	ranking	(50k)	54.2	85.6	99.0	86.7	98.5	88.6	98.9	82.9	98.3
major	classify %5	(10k)	20.0	23.6	86.4	24.1	84.5	22.8	89.6	23.9	87.9
major	classify %5	(25k)	20.0	24.6	96.7	26.8	96.3	24.8	97.7	27.0	97.2
major	classify %5	(50k)	20.0	31.6	99.3	34.2	99.2	30.0	99.5	33.9	99.4
major	ranking	(10k)	50.5	52.5	88.8	54.1	86.2	52.4	90.3	54.1	88.3
major	ranking	(25k)	50.5	52.2	96.4	53.7	97.3	52.6	96.9	53.6	97.5
major	ranking	(50k)	50.5	53.9	99.6	55.0	99.5	53.6	99.4	55.0	99.3
major	subtraction	(10k)	1.0	1.1	21.6	1.1	82.5	1.0	23.2	1.1	84.3
major	subtraction	(25k)	1.0	1.1	89.1	1.2	96.7	1.2	84.7	1.2	97.0
major	subtraction	(50k)	1.0	1.1	98.4	1.2	99.3	1.1	97.3	1.2	99.0

Result 3 (Figure 4, ♣). *Without CoT examples, the model's test accuracy is significantly low,* even for the simplest, single-step *manipulation tasks.*

Knowledge Classification & Comparison

field	task	#train individuals	baseline	BIO pretrained model				QA finetuned model			
				trained w/o hint	trained with hint			trained w/o hint	trained with hint		
				test acc	test acc (with hint)	test acc (w/o hint)	hint acc	test acc	test acc (with hint)	test acc (w/o hint)	hint acc
birthmonth	classify %2	(2.5k)	50.0	60.4	77.8	65.2	64.5	61.9	80.4	65.2	69.1
birthmonth	classify %2	(5k)	50.0	67.3	87.3	72.7	80.3	68.0	89.5	72.8	83.9
birthmonth	classify %2	(10k)	50.0	75.9	94.2	80.3	91.0	76.4	95.0	79.9	92.8
birthmonth	classify %2	(25k)	50.0	86.4	98.6	91.1	97.8	87.1	98.8	90.9	98.4
birthmonth	classify %2	(50k)	50.0	95.3	99.5	97.5	99.2	96.3	99.7	97.5	99.5
birthmonth	ranking	(2.5k)	54.2	53.7	65.4	59.6	44.2	57.3	65.5	57.6	44.9
birthmonth	ranking	(5k)	54.2	59.2	75.5	63.4	63.6	62.5	75.1	63.1	62.6
birthmonth	ranking	(10k)	54.2	65.4	87.7	67.0	82.7	65.9	88.9	66.3	83.9
birthmonth	ranking	(25k)	54.2	75.6	96.7	75.8	95.4	78.3	97.4	72.5	96.3
birthmonth	ranking	(50k)	54.2	85.6	99.0	86.7	98.5	88.6	98.9	82.9	98.3
major	classify %5	(10k)	20.0	23.6	86.4	24.1	84.5	22.8	89.6	23.9	87.9
major	classify %5	(25k)	20.0	24.6	96.7	26.8	96.3	24.8	97.7	27.0	97.2
major	classify %5	(50k)	20.0	31.6	99.3	34.2	99.2	30.0	99.5	33.9	99.4
major	ranking	(10k)	50.5	52.5	88.8	54.1	86.2	52.4	90.3	54.1	88.3
major	ranking	(25k)	50.5	52.2	96.4	53.7	97.3	52.6	96.9	53.6	97.5
major	ranking	(50k)	50.5	53.9	99.6	55.0	99.5	53.6	99.4	55.0	99.3
major	subtraction	(10k)	1.0	1.1	21.6	1.1	82.5	1.0	23.2	1.1	84.3
major	subtraction	(25k)	1.0	1.1	89.1	1.2	96.7	1.2	84.7	1.2	97.0
major	subtraction	(50k)	1.0	1.1	98.4	1.2	99.3	1.1	97.3	1.2	99.0

Result 4 (Figure 4, ♠). *Even when CoT examples are included during training, the model still struggles to answer without a hint during testing, indicating that* including hints during training does not improve test-time accuracy when hints are removed.

Knowledge Classification & Comparison

field	task	#train individuals	baseline	BIO pretrained model				QA finetuned model			
				trained w/o hint	trained with hint			trained w/o hint	trained with hint		
				test acc	test acc (with hint)	test acc (w/o hint)	hint acc	test acc	test acc (with hint)	test acc (w/o hint)	hint acc
birthmonth	classify %2	(2.5k)	50.0	60.4	77.8	65.2	64.5	61.9	80.4	65.2	69.1
birthmonth	classify %2	(5k)	50.0	67.3	87.3	72.7	80.3	68.0	89.5	72.8	83.9
birthmonth	classify %2	(10k)	50.0	75.9	94.2	80.3	91.0	76.4	95.0	79.9	92.8
birthmonth	classify %2	(25k)	50.0	86.4	98.6	91.1	97.8	87.1	98.8	90.9	98.4
birthmonth	classify %2	(50k)	50.0	95.3	99.5	97.5	99.2	96.3	99.7	97.5	99.5
birthmonth	ranking	(2.5k)	54.2	53.7	65.4	59.6	44.2	57.3	65.5	57.6	44.9
birthmonth	ranking	(5k)	54.2	59.2	75.5	63.4	63.6	62.5	75.1	63.1	62.6
birthmonth	ranking	(10k)	54.2	65.4	87.7	67.0	82.7	65.9	88.9	66.3	83.9
birthmonth	ranking	(25k)	54.2	75.6	96.7	75.8	95.4	78.3	97.4	72.5	96.3
birthmonth	ranking	(50k)	54.2	85.6	99.0	86.7	98.5	88.6	98.9	82.9	98.3
major	classify %5	(10k)	20.0	23.6	86.4	24.1	84.5	22.8	89.6	23.9	87.9
major	classify %5	(25k)	20.0	24.6	96.7	26.8	96.3	24.8	97.7	27.0	97.2
major	classify %5	(50k)	20.0	31.6	99.3	34.2	99.2	30.0	99.5	33.9	99.4
major	ranking	(10k)	50.5	52.5	88.8	54.1	86.2	52.4	90.3	54.1	88.3
major	ranking	(25k)	50.5	52.2	96.4	53.7	97.3	52.6	96.9	53.6	97.5
major	ranking	(50k)	50.5	53.9	99.6	55.0	99.5	53.6	99.4	55.0	99.3
major	subtraction	(10k)	1.0	1.1	21.6	1.1	82.5	1.0	23.2	1.1	84.3
major	subtraction	(25k)	1.0	1.1	89.1	1.2	96.7	1.2	84.7	1.2	97.0
major	subtraction	(50k)	1.0	1.1	98.4	1.2	99.3	1.1	97.3	1.2	99.0

Result 5 (Figure 4, ♢). *The difference between a BIO pretrained and a QA finetuned model is minimal for downstream knowledge manipulation tasks.*

Knowledge Inverse Search

- Give me the [first/full] name of the person born on October 2, 1996? (`bdate_to_first`, `bdate_to_full`)
- Give me the [first/full] name of the person born on October 2, 1996 in Princeton, NJ? (`birth_to_first`, `birth_to_full`)
- Give me the [first/full] name of the person who studied Communications at Massachusetts Institute of Technology and worked for Meta Platforms? (`three_to_first`, `three_to_full`)
- Give me the [first/full] name of the person who studied Communications at Massachusetts Institute of Technology, was born in Princeton, NJ, and worked for Meta Platforms? (`four_to_first`, `four_to_full`)
- Give me the [first/full] name of the person who studied Communications at Massachusetts Institute of Technology, was born on October 2, 1996 in Princeton, NJ, and worked for Meta Platforms at Menlo Park, CA? (`all_to_first`, `all_to_full`)

Jane Austen Novel Task

Inverse search: "In <Pride and Prejudice>, what's the sentence ***before***: <sentence2>?"
Forward search: "In <Pride and Prejudice>, what's the sentence ***after***: <sentence1>?"

	Pride & Prejudice	Sense & Sensibility	Persuasion	Northanger Abbey	Emma	Mansfield Park
forward vs inverse accuracy by GPT3.5	0.5% vs 14.4%	0.3% vs 5.4%	0.07% vs 4.3%	0.6% vs 5.5%	0.8% vs 7.2%	0.7% vs 5.5%
forward vs inverse accuracy by GPT4	0.8% vs 65.9%	0.9% vs 40.2%	0.5% vs 33.9%	0.9% vs 41.0%	0.6% vs 42.7%	0.3% vs 31.7%

Wiki Bio Task

Inverse search: "what's the full name of the celebrity born on <date> in <city> who is a <occupation>?" GPT3.5 acc = 23.9% GPT4: 42%
Forward search: "what's the birthday and year of <name> who is a <occupation> and was born in <city>?" GPT3.5 acc = 89.5% GPT4: 99%

Chinese Idiom Task

Given a common 4-letter Chinese idiom such as *指鹿为马*, mask out its i-th letter (for i=1,2,3, or 4) and let GPT fill out the missing letter.

Prompt 1: 成语"X鹿为马"的X是什么字? GPT3.5 accuracy 9.4%, GPT4 accuracy 17.6%
Prompt 2: 成语"指X为马"的X是什么字? GPT3.5 accuracy 29.5%, GPT4 accuracy 36.1%
Prompt 3: 成语"指鹿X马"的X是什么字? GPT3.5 accuracy 32.0%, GPT4 accuracy 76.7%
Prompt 4: 成语"指鹿为X"的X是什么字? GPT3.5 accuracy 56.7%, GPT4 accuracy 90.6%

Chinese Poem Task

Given a famous two-sentence Chinese poem such as *劝君更尽一杯酒, 西出阳关无故人*, let GPT answer what's the sentence ***before***/***after*** <sentence2/1>

Inverse search: "西出阳关无故人"的上一句是什么? GPT3.5 accuracy 2.1%, GPT4 accuracy 7.3%
Forward search: "劝君更尽一杯酒"的下一句是什么? GPT3.5 accuracy 33.0%, GPT4 accuracy 66.5%

Knowledge Inverse Search

Data augmentation used in reverse search

- The person was born on October 2, 1996. Anya Briar Forger spent her early years in Princeton, NJ... (bioS multi5+reverse1)
- The person was born on October 2, 1996. She spent her early years in Princeton, NJ. Anya Briar Forger... (bioS multi5+reverse2)
- The person was born on October 2, 1996. She spent her early years in Princeton, NJ... The person's name is Anya Briar Forger. (bioS multi5+reverse6)
- The person spent her early years in Princeton, NJ. [... 4 more sentences in random order...] She had a professional role at Meta Platforms. The person's name is Anya Briar Forger. (bioS multi5+permute+reverse6)

Knowledge Inverse Search

	QA bdate_to_first	QA bdate_to_full	QA b_to_first	QA b_to_full	QA all_to_first	QA all_to_full	QA four_to_first	QA four_to_full	QA three_to_first	QA three_to_full	MIX bdate_to_first	MIX bdate_to_full	MIX b_to_first	MIX b_to_full	MIX all_to_first	MIX all_to_full	MIX four_to_first	MIX four_to_full	MIX three_to_first	MIX three_to_full
baseline	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0	0.2	0.0
bioS multi5 + reverse1	45.9	49.2	50.7	50.8	45.5	44.6	1.1	0.0	1.7	0.0	33.5	33.8	28.7	27.2	1.0	1.0	0.8	0.0	0.9	0.2
bioS multi5 + reverse2	10.8	1.2	96.9	92.2	96.5	91.9	1.0	0.4	1.7	0.0	6.7	1.9	97.8	94.2	1.0	1.0	1.0	0.2	0.9	0.0
bioS multi5 + reverse6	1.3	0.0	2.8	0.0	96.1	93.0	91.8	89.3	79.2	77.0	1.2	0.0	1.0	0.0	98.5	94.9	98.6	95.1	97.1	92.9
bioS multi5 + permute + reverse6	2.9	0.2	6.3	1.6	97.6	94.0	64.0	69.2	22.2	26.9	1.2	0.4	2.2	0.8	99.0	95.7	96.5	71.7	85.6	55.9
bioS single	1.0	0.0	1.4	0.0	0.9	0.0	0.9	0.0	1.7	0.0	1.0	0.0	1.0	0.0	0.8	0.0	0.6	0.0	0.9	0.0
bioS single + fullname	0.8	0.0	1.6	0.0	1.4	0.0	1.1	0.0	1.3	0.0	1.0	0.0	0.8	0.0	0.6	0.0	1.0	0.0	0.9	0.0
bioS single + permute1	0.8	0.0	1.4	0.0	1.1	0.0	0.9	0.0	1.3	0.0	1.0	0.0	0.8	0.0	0.8	0.0	0.8	0.0	0.7	0.0
bioS single + permute2	0.8	0.0	1.4	0.0	1.1	0.0	0.6	0.0	1.0	0.0	1.2	0.0	0.8	0.0	0.6	0.0	1.0	0.0	1.1	0.0
bioS single + permute5	0.8	0.0	1.2	0.0	1.4	0.0	1.0	0.0	1.0	0.0	0.8	0.0	0.8	0.0	1.0	0.0	0.8	0.0	0.9	0.0
bioS single + permute1 + fullname	0.6	0.0	1.0	0.0	1.7	0.0	0.9	0.0	1.0	0.0	1.2	0.0	1.0	0.0	0.8	0.0	1.0	0.0	0.7	0.0
bioS single + permute2 + fullname	0.6	0.0	1.4	0.0	1.4	0.0	0.9	0.0	1.0	0.0	1.2	0.0	1.0	0.0	0.8	0.0	1.0	0.0	0.9	0.0
bioS single + permute5 + fullname	0.6	0.0	1.0	0.0	1.1	0.0	0.9	0.0	1.3	0.0	1.2	0.0	1.0	0.0	0.8	0.0	1.0	0.0	0.9	0.0
bioS multi2	0.8	0.0	1.2	0.0	1.4	0.0	0.9	0.0	1.7	0.0	1.2	0.0	1.0	0.0	0.8	0.0	1.0	0.0	1.1	0.0
bioS multi2 + fullname	0.6	0.0	1.2	0.0	1.4	0.0	1.1	0.0	1.7	0.0	1.2	0.0	1.4	0.0	1.0	0.0	0.6	0.0	0.9	0.0
bioS multi2 + permute	0.8	0.0	1.0	0.0	1.7	0.0	1.1	0.0	1.3	0.0	1.2	0.0	1.0	0.0	0.8	0.0	0.8	0.0	0.9	0.0
bioS multi2 + permute + fullname	0.8	0.0	1.2	0.0	1.4	0.0	0.9	0.0	1.3	0.0	1.2	0.0	1.0	0.0	1.3	0.0	0.8	0.0	0.9	0.0
bioS multi5	0.8	0.0	1.4	0.0	1.1	0.0	1.1	0.0	1.0	0.0	1.2	0.0	1.0	0.0	0.8	0.0	0.6	0.0	1.1	0.0
bioS multi5 + fullname	0.6	0.0	1.2	0.0	1.7	0.0	0.8	0.0	1.7	0.0	1.2	0.0	1.0	0.0	0.8	0.0	0.8	0.0	0.7	0.0
bioS multi5 + permute	0.8	0.0	1.4	0.0	1.4	0.0	0.9	0.0	1.7	0.0	0.6	0.0	0.8	0.0	1.0	0.0	0.8	0.0	0.9	0.0
bioS multi5 + permute + fullname	0.6	0.0	1.0	0.0	1.7	0.0	0.9	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.8	0.0	0.8	0.0	0.9	0.0

Result 7 (Figure 6, ♡). *Models have near-zero accuracy to inverse knowledge search in* $\mathcal{P}_{\text{test}}$, even for *the simplest task* ***all_to_first***, even with *the BIO+QA mixed training approach, and* even with *strong pretrain data knowledge augmentation.*[16]

Knowledge Inverse Search

Figure 8: How GPT-4 uses CoT to perform inverse knowledge search on the Bible task.

Result 9. *To improve inverse search of critical documents by LLMs, not only we can employ RAG [10] or preprocess training data to include reverse knowledge (see Figure 6-top, or practically through a "rewrite" prompt), we can also introduce line numbers (see Figure 8).*

Physics of Language Models: Part 3.2
Knowledge Manipulation

Results 1-2

knowledge partial / dual retrievals may be difficult

model can say birthday "October 2, 1996" but not "1996"

Results 3-5

knowledge classification / comparison is hard without CoT

model cannot say "even year" without saying "1996" (CoT)

Result 7

knowledge inverse search is impossible

impossible to answer "who was born on October 2, 1996"

Results 6/8/9

Turing tests to distinguish Humans from contemporary AIs

universal counter-examples, applicable to GPT-4 and others

Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws

Mechanisms

Intelligence

Idealization

Principles

How to Quantify Knowledge

calculate amount of learned knowledge (in *bits*)

supported by a *lower-bound Theorem*

pretrain LLMs (varied sizes)

synthetic English data describing knowledge tuples

e.g. (Anya – birthday – 10/2/1996)

(Eugeo – birth city – Washington D.C.)

bioS/bioR: N biographies

(regardless of writing styles, repetitions)

bioD: a synthetic data with hyperparameter

N – distinct names from N_0 possible names

K – number of knowledge attributes

T – vocabulary size

C,L – values in C chunks, each of length L

D – value has diversity D

If birthdates are uniformly drawn from 12 x 28 x 200(yrs) possibilities, this is $\log_2(12 \times 28 \times 200) = 60.21$ bits

If cities are uniformly drawn from 300 US cities, this is $\log_2(300) = 8.23$ bits

$$\log_2 \binom{N_0}{N} + NKC \log_2 D + K \log_2 \binom{T^L}{D}$$

N-exposure vs N-passes

Each knowledge tuple can be present in multiple times in a single-pass of the dataset.

Knowledge Capacity of LLMs

RESULT 1+2+3: GPT2, trained with standard AdamW, consistently achieves a 2bit/param capacity ratio across all data settings after sufficient training. This includes various model sizes, depths, widths, data sizes, types (synthetic/semi-synthetic), and hyperparameters (e.g., name/value length, attribute number, value diversity).

In a 1000-exposure setup

RESULT 4: With 100 exposures, an *undertrained* GPT2's capacity ratio falls to 1bit/param.

RESULT 5: In the 1000-exposure setting, a 2bit/param capacity ratio appears to be a **universal rule**: all models, even without MLP layers, closely achieve this ratio.

RESULT 6: With 100 exposures, some archs show limitations; notably, LLaMA/Mistral's capacity ratio is 1.3x lower than GPT2's, even after best-tuned learning rates.

RESULT 7: Further controlled experiments indicate that "gated MLP" usage leads to LLaMA/Mistral architecture's underperformance in knowledge storage.

Attention layers can store knowledge

Knowledge Capacity of LLMs – Quantized LLMs

scaling law (quantization) – Result 8

quantizing → int8 *does not affect* scaling laws *at all*

quantizing → int4 hurts capacity by more than 2x

bioS data

float16/32 → no diff → int8 → worsen >2x → int4

10001010101010100100110101010111 01111010 1101

Knowledge Capacity of LLMs – MoE vs. Dense

scaling law (MoE)

LLMs with **mixture of even 32 experts** can be very efficient in storing knowledge – Result 9

despite using only 8.8% of total parameters during inference!

Knowledge Capacity of LLMs – Effect of Junk Data

scaling laws (pretrain data of mixed qualities)

- **"Junk" data** significantly ***harm*** LLM's knowledge capacity on **good data** (sometimes by 20x times!) – Result 10
  e.g. common crawls, internet "junks" e.g. Wikipedia
- repetitive knowledge … *does not* harm … – Result 11

illustration

1/8 good data | 7/8 training tokens from "junk" data

train **without** junk for 100 exposures — Figure 8a

20x worse → train **with** junk for 100 exposures — Figure 8b

3x worse → train **with** junk for 300 exposures — Figure 8c

a simple fix! – Result 12

Figure 8g **10x better!**

Figure 8h **3x better!**

- add domain name (e.g., "wikipedia.org") at front of all pretrain data paragraphs [data] [data] [data]
  LLMs can ***automatically*** detect domains rich in high-quality knowledge and prioritize learning from them

Knowledge Capacity of LLMs

Results 1/2/3/5 — a *universal* law: "all" LLMs can store 2bit/param knowledge

⇒ predict: 7B model can store all English wiki + textbooks knowledge

Results 4/6/7 — scaling laws for insufficient training

e.g. LLaMA/Mistral architectures *30% worse* than $GPT2_{rotary}$ in capacity

Results 8/9 — scaling laws for quantization + mixture-of-expert (MoE)

e.g. 2bit/param holds *even for* int8 parameters

Results 10/11/12 — scaling laws for mixed-quality data (wikipedia vs internet)

e.g. a technique to improve LLM's capacity – sometimes by 10x

Physics of Language Models: Part 2.1, Grade-School Math and the Hidden Reasoning Process

Physics of Language Models: Part 2, Reasoning

Tian Ye (CMU/Meta)

Zicheng Xu (Meta)

Yuanzhi Li (MBZUAI)

Zeyuan Allen-Zhu (Meta)

Dataset for Checking Reasoning

GOAL: study how LLMs solve grade-school math (GSM)

can't use GSM8k – too small, data contamination, etc.
can't use GPT-4 augmented GSM8k
– too biased, too few templates

iGSM – an infinite, synthetic GSM8k-like data

remove common sense (candle burns -> length shrinks) so LLMs can be ***pretrained*** on such data

iGSM aims to capture

Direct dependency:
e.g. A=5*(X+Y) ⇒ A depends on X and Y

Instance dependency:
e.g. X classrooms each has Y messenger bags

Implicit dependency:
e.g. Bob has 3x more fruits than Alice. Alice has 3 apples, 4 eggs and 2 bananas. (eggs are not fruits)

structure graph

dependency graph (DAG)

an easy example

The number of each **Riverview High's Film Studio** equals 5 times as much as the sum of each *Film Studio's Backpack* and each *Dance Studio's School Daypack*. The number of each Film Studio's School Daypack equals 12 more than the sum of each Film Studio's Messenger Bag and each Central High's Film Studio. The number of each Central High's Film Studio equals the sum of each Dance Studio's School Daypack and each Film Studio's Messenger Bag. The number of each Riverview High's Dance Studio equals the sum of each Film Studio's Backpack, each Film Studio's Messenger Bag, each Film Studio's School Daypack and each Central High's Backpack. The number of each Dance Studio's School Daypack equals 17. The number of each Film Studio's Messenger Bag equals 13. *How many Backpack does Central High have?*

a solution (=CoT) with op=7 operations

Define Dance Studio's School Daypack as p; so p = 17.
Define Film Studio's Messenger Bag as W; so W = 13.
Define Central High's Film Studio as B; so B = p + W = 17 + 13 = 7.
Define Film Studio's School Daypack as g; R = W + B = 13 + 7 = 20; so g = 12 + R = 12 + 20 = 9.
Define Film Studio's Backpack as w; so w = g + W = 9 + 13 = 22.
Define Central High's Backpack as c; so c = B * w = 7 * 22 = 16. ***[Answer: 16.]***

We want to focus on ***reasoning***, not arithmetics; so we use **mod 23**.

Generalization of Reasoning

– Result 2

GPT2 learns iGSM by true generalization

*few-shot GPT-4o fails on **op>=11***

OOD generalization – tested on problems longer than pretrain

pretrain data (Medium): iGSM^{Med} (**op<=15**) (>7 billion solution templates) → **test data** (Medium):	op=15	op=20	op=21	op=22	op=23
	99+%	92%	88%	85%	78%
(Hard): iGSM^{Hard} (**op<=21**) (>15 trillion solution templates) → (Hard):	op=21	op=28	op=29	op=30	op=31	op=32
	99+%	94%	92%	90%	87%	83%

⟹ truly learns some reasoning skill (not by memorization)

but what reasoning skill?

GPT2 at least achieves "level-1" reasoning skill

a "level-0" reasoning brute-forces to compute all params maximally ✗

a **"level-1" reasoning** uses topological sort + gives shortest CoT ✓

But how is this possible?

model must **"mental process"** before the first solution sentence **(i.e., before CoT!)**, see Results 4-5

Define Dance Studio's School Daypack as p; so p = 17.
Define Film Studio's Messenger Bag as W; so W = 13.
Define Central High's Film Studio as B; so B = p + W = 17 + 13 = 7.
Define Film Studio's School Daypack as g; ... + R = 12 + 20 = 9.
Define Film Studio's Backpack as w; so w = g + W = 9 + 13 = 22.
Define Central High's Backpack as c; so c = B * w = 7 * 22 = 16.

solution in topological order & *excluding* unnecessary parameters

problem description in *random* order

– Result 3

Generalization of Reasoning

	iGSM-med_pq						iGSM-med_qp						iGSM-hard_pq							iGSM-hard_qp						
	op ≤ 15	op=15	op=20	op=21	op=22	op=23	op ≤ 15	op=15	op=20	op=21	op=22	op=23	op ≤ 21	op=21	op=28	op=29	op=30	op=31	op=32	op ≤ 21	op=21	op=28	op=29	op=30	op=31	op=32
avg unnecessary operation	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
avg unnecessary operation (reask)	0.02	0.11	0.15	0.17	0.19	0.17	0.03	0.11	0.17	0.21	0.19	0.20	0.07	0.46	0.52	0.54	0.57	0.66	0.66	0.09	0.40	0.45	0.45	0.58	0.53	0.59
avg unnecessary parameter	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
avg unnecessary parameter (reask)	0.01	0.09	0.12	0.12	0.14	0.12	0.03	0.10	0.15	0.16	0.15	0.16	0.06	0.34	0.36	0.36	0.38	0.43	0.44	0.07	0.30	0.32	0.31	0.40	0.36	0.42

Result 3. *Figure 4 shows that GPT2 predominantly solves the iGSM problems with a "level-1" reasoning skill, avoiding unnecessary computations, even when evaluated out-of-distribution.*

How LLMs Perform Reasoning

- $\texttt{nece}(A)$: if parameter A is necessary for computing the answer.
- $\texttt{dep}(A, B)$: if parameter A (recursively) depends on parameter B given the problem statement.
- $\texttt{known}(A)$: if parameter A has already been computed.
- $\texttt{value}(A)$: the value of parameter A (a number between 0-22, or 23 if $\texttt{known}(A) = \text{false}$).
- $\texttt{can_next}(A)$: if A can be computed in the next solution sentence (namely, its predecessors have all been calculated). Note that A might not be necessary to answer the question.
- $\texttt{nece_next}(A)$: if parameter A satisfies both $\texttt{can_next}(A)$ and $\texttt{nece}(A)$.

dep(*A*,*B*) – at the end of problem description, does model know *parameter A* ***recursively*** *depends on B*? ***It does! 99%***

nece(*A*) – after question is asked, does model know *if A is necessary* for answering the question for all *A*? ***It does! 99%***

"**[Problem]** The number of each Riverview High's Film Studio equals 5 times … The number of each Film Studio's Messenger Bag equals 13. ***[Question]*** *How many Backpack does Central High have?* **[Solution]** *Define Dance Studio's School Daypack as p … Define Central High's Film Studio as B; so B = p + W = 17 + 13 = 7. Define* …. ***[Answer]*** *16.* "

can_next(*A*) – in middle of solution, does model know if *A can be computed next* for all *A*? ***It does! 99%***

⟹ explains how GPT2 **achieves "level-1" reasoning** (i.e. generate shortest solution using topological sort) – Result 4

Probing LLMs for Understanding Reasoning Capabilities

V-probing

“Level-2” Reasoning Skill of LLMs

– Result 5

GPT2 uses “level-2” reasoning skill different from Humans

this skill is not needed for solving the math problem

GPT2 learns dep(A,B) and can_next(A) **even for all unnecessary *A***

$\Longrightarrow$ it *mentally* computes all-pair dependency graph *before the question* is asked **(a “level-2” reasoning skill)**

Humans *start from question* to only identify necessary parameters **(human’s backward reasoning skill)**

may be preliminary signal of where **G** in AGI can come from (generalizing to skills not taught in pretrain data)

Where Do LLMs Fail?

Result 6 (Figure 8). *Combining these, we conclude:*

- *Many reasoning mistakes made by the language model are systematic, stemming from* errors in its mental process, *not merely random from the generation process.*
- *Some of the model's mistakes can be discovered by probing its inner states* even before the model opens its mouth *(i.e., before it says the first solution step).*

mistakes from nece(*A*)

from Results 4-5: *before* generating solution, model "mentally" calculates what params are necessary
⟹ if param A is **wrongly** calculated as **nece(*A*)**=True in planning stage, model will likely say it in the solution

can detect such mistakes before model opens mouth (using V-probing)
⟹ mistakes are systematic, not random from the generation process

How LLMs makes mistakes on iGSM?

GPT-4 (few shot)
GPT-4o (few shot)
GPT-2 (pretrained on iGSM)

All make two types of mistakes

1. Define Dance Studio's Messenger Bag as S; so S = 3.
2. Define **Lakeshore High's Dance Studio** as D; so D = 2.
3. Define Lincoln High's Dance Studio as L; so L = S * 7 = 3 * 7 = 21.
4. Define Messenger Bag's Calculator as C; so C = S = 3.
5. Define **Dance Studio's Canvas Backpack** as ...

WARNING: unnecessary parameter *A*

ERROR: parameter *A* not ready to compute

What Matters in Reasoning?

We claim: Depth matters for reasoning

– Result 7

	iGSM-med_pq						iGSM-med_qp						iGSM-hard_pq							iGSM-hard_qp							avg
	in-dist		out-of-dist (OOD)				in-dist		out-of-dist (OOD)				in-dist		out-of-dist (OOD)					in-dist		out-of-dist (OOD)					
	op ≤ 15	op=15	op=20	op=21	op=22	op=23	op ≤ 15	op=15	op=20	op=21	op=22	op=23	op ≤ 21	op=21	op=28	op=29	op=30	op=31	op=32	op ≤ 21	op=21	op=28	op=29	op=30	op=31	op=32	
dep4 - size1 - head21	99.5	92.7	74.7	68.0	62.4	54.5	99.4	93.3	73.3	66.8	61.1	54.6	98.9	90.8	72.4	67.7	62.1	57.1	50.6	99.1	89.8	69.4	62.2	57.8	52.3	45.7	72.2
dep4 - size2 - head30	99.6	94.7	74.2	67.9	61.6	53.1	99.4	94.5	78.1	71.9	65.7	58.8	97.0	71.7	46.3	40.6	37.0	32.3	27.3	99.4	92.1	74.5	69.5	64.7	59.1	53.2	68.6
dep8 - size1 - head15	100	98.8	89.7	86.5	82.8	76.8	100	99.2	92.4	88.5	84.2	78.7	100	99.1	94.6	92.0	89.7	86.4	82.2	100	99.0	92.2	89.6	86.2	82.4	77.3	90.3
dep8 - size2 - head21	100	99.3	93.7	91.6	88.3	83.6	99.9	99.0	90.2	87.1	83.3	76.3	100	99.2	93.6	91.3	88.6	85.6	82.6	100	99.1	93.5	91.3	89.1	85.7	81.2	91.3
dep12 - size1 - head12	100	99.3	92.0	88.9	84.2	77.9	100	99.4	92.2	89.2	83.9	77.9	100	99.5	96.0	94.1	91.0	88.5	84.3	100	99.3	95.3	93.0	91.9	88.0	84.5	91.9
dep12 - size2 - head17	100	99.5	94.0	91.9	89.0	82.7	100	99.0	90.8	85.4	80.2	73.2	100	99.8	97.1	95.5	93.5	91.8	88.0	100	99.5	94.5	91.9	88.9	86.8	81.3	92.1
dep16 - size1 - head10	100	99.6	94.6	91.9	87.9	82.7	100	99.5	89.9	85.0	79.1	71.1	100	99.6	97.0	95.2	94.2	92.2	88.5	100	99.4	95.8	93.8	92.4	88.9	85.8	92.5
dep16 - size2 - head15	100	99.8	95.9	93.7	90.4	86.5	100	99.8	95.6	93.5	90.3	84.3	100	99.7	97.5	96.3	95.1	92.9	89.5	100	99.8	97.3	96.0	94.2	91.9	88.9	95.0
dep20 - size1 - head9	100	99.8	95.5	93.6	90.0	86.3	100	99.6	94.8	91.4	87.4	80.4	100	99.8	97.0	95.1	94.0	91.0	87.4	100	99.6	96.6	94.5	92.8	90.1	86.5	94.0
dep20 - size2 - head13	100	99.8	95.8	93.3	89.2	84.4	100	99.6	93.7	91.8	87.4	81.3	100	99.8	98.0	96.7	95.9	93.9	90.9	100	99.9	97.5	96.0	95.2	92.4	89.7	94.7

larger model, smaller depth (4-layer, 30-head)

smaller model, larger depth (20-layer, 9-head)

This cannot be mitigated by CoT – deciding what's the first CoT step may still require deep, multi-step mental reasoning (planning)

– Result 8

Depth matters because of the complexity of mental reasoning

parameter's **distance *t*** to the question

⟹ deeper layers are better to compute nece(A) for larger t (which requires t-steps of mental reasoning)

see paper for 44 more figures like this

LLM Reasoning

Result 1 — iGSM: a synthetic, infinite math dataset to simulate GSM8k
pretrain on iGSM + probe model's hidden (mental) reasoning process

Result 2-3 — LLMs exhibit a "level-1" reasoning skill
they mentally compute topo sort + shortest solution – like Humans

Result 4-5 — LLMs secretly learn a "level-2" reasoning skill
they mentally compute "all-pair dependencies" – but Humans don't

Result 6 — probing reveals how LLMs make reasoning mistakes
can catch mistakes even before LLMs start to speak

Result 7-8 — depth matters for long reasoning tasks (even with CoT)
refute OpenAI's scaling law which says "only size matters"

Physics of Language Models: Part 1
Learning Hierarchical Language Structures

Mechanisms

Intelligence

Idealization

Principles

Physics of Language Models: Part 1, Understanding Language

Zeyuan Allen-Zhu (Meta)

Yuanzhi Li (MBZUAI)

Developing More Complex Probing Data

Prior work to interpret the inner working of LLMs:

"**induction head**": [A][B] … [A] → [B]

"**Interpretability in the wild**": name mover head, inhibition head, etc.

…

Mostly on the token level
(e.g., explain how LLMs copy sentences)
not about deep, hierarchical tasks

We use synthetic data to examine LLMs' inner working

We design 20+ probabilistic context-free grammars (CFGs)

root |->20 21	19|->18 16 18	16|->15 15	13|->11 12	10|->8 9 9	7|->2 2 1
root |->20 19 21	19|->17 18	16|->13 15 13	13|->12 11 12	10|->9 7 9	7|->3 2 2
root |->21 19 19	19|->18 18	16|->14 13	13|->10 12 11	10|->7 9 9	7|->3 1 2
root |->20 20	20|->16 16	16|->14 14	14|->10 12	11|->8 8	7|->3 2
	20|->16 17	17|->15 14 13	14|->12 10 12	11|->9 7	8|->3 1 1
	20|->17 16 18	17|->14 15	14|->12 11	11|->9 7 7	8|->1 2
	21|->18 17	17|->15 14	14|->10 12 12	12|->7 9 7	8|->3 3 1
	21|->17 16	18|->14 15 13	15|->10 11 11	12|->9 8	9|->1 2 1
	21|->16 17 18	18|->15 13 13	15|->11 11 10	12|->8 8 9	9|->3 3
	21|->16 18	18|->13 15	15|->10 10		9|->1 1
			15|->12 12 11		

an example sentence (length 354)

3322131233121131232113223123121112132113223113113223331231211121312132113223113113223331232122121312322211113322131311311311111132312331331331331133133333223133211212112123312331121113313333331123333131111333312113211312121133333321211112121322322332213311322113232331311121322322221211113333112132222133221121213312133132213221322211121333123223331211213322122133121333122233312

ambiguous + hierarchical + hard to parse
(*require dynamic programming, DP*)

locally from "311211133" it is hard to derive where it comes from (needs DP)

They are *much harder* than English grammars

English grammar can be parsed (mostly) via greedy.
A Penn Treebank CFG = learnable using 100k-param tiny language model

Can LLMs Learn Hierarchical Language Structure

generation acc (%)

	GPT		GPT_rel		GPT_rot		GPT_pos		GPT_uni	
	cut0	cut50	cut0	cut50	cut0	cut50	cut0	cut50	cut0	cut50
cfg3b	99.8	99.8	99.8	99.9	99.8	99.9	99.9	99.9	99.9	100.0
cfg3i	99.5	99.5	99.8	99.8	99.4	99.5	99.8	99.8	99.6	99.7
cfg3h	96.8	96.9	99.7	99.6	99.6	99.5	99.0	99.0	98.9	98.8
cfg3g	64.1	63.8	99.1	99.2	98.6	98.4	97.0	96.9	96.7	96.9
cfg3f	57.1	57.3	98.8	98.8	97.6	97.7	93.9	93.8	92.8	92.9
cfg3e1	98.1	98.9	98.4	99.0	98.2	98.9	98.3	98.9	98.6	99.0
cfg3e2	99.3	99.5	99.6	99.7	99.6	99.7	99.5	99.7	99.4	99.6

entropy (bits)

	truth	GPT	GPT_rel	GPT_rot	GPT_pos	GPT_uni
cfg3b	169	169	169	169	169	169
cfg3i	185	190	189	189	190	189
cfg3h	204	203	203	203	202	203
cfg3g	268	272	267	268	266	267
cfg3f	268	275	270	272	269	269
cfg3e1	216	214	213	213	214	213
cfg3e2	256	252	255	251	253	252

KL divergence

	GPT	GPT_rel	GPT_rot	GPT_pos	GPT_uni
cfg3b	0.00008	0.00011	0.00009	0.00009	0.00004
cfg3i	0.00024	0.00014	0.00028	0.00015	0.00021
cfg3h	0.00078	0.00023	0.00023	0.00027	0.00036
cfg3g	0.00450	0.00034	0.00047	0.00058	0.00069
cfg3f	0.00455	0.00043	0.00060	0.00093	0.00112
cfg3e1	0.00019	0.00014	0.00016	0.00013	0.00011
cfg3e2	0.00031	0.00025	0.00025	0.00011	0.00011

- LLMs can generate strings following CFGs accurately
- LLMs can generate diverse strings following CFGs
- LLMs generate strings following CFG's probability distribution

Position embedding fails to learn CFG structure; relative distance is important

Learning Hierarchical Language Structure

Goal

hierarchical interpretation of GPT

beyond token-level interpretation (i.e., induction/copy head)

Result 1-3

GPT can learn very long, synthetic CFGs (context-free grammar)

but must use rotary/relative attention

Result 4-9

GPT uses dynamic programming (DP) to learn such CFGs

Results 4-5: DP states = encoded linearly in the hidden states

Results 6-9: DP transitions = encoded in the attention layer

encoder-models (e.g. BERT, DeBERTa) do not!

Result 10-13

GPT learns implicit / corrupted CFGs

it's beneficial to add structure or grammar mistakes in pretrain data

What Next?

- Controlled experiments for small and efficient language models
- Can we extend the synthetic data generation process for multi-modal scenarios?
- How to use controlled (and yet large-scale) experiments for low-resources linguistic patterns?
- Combining multiple learning objectives – reasoning with knowledge