Understand LLM

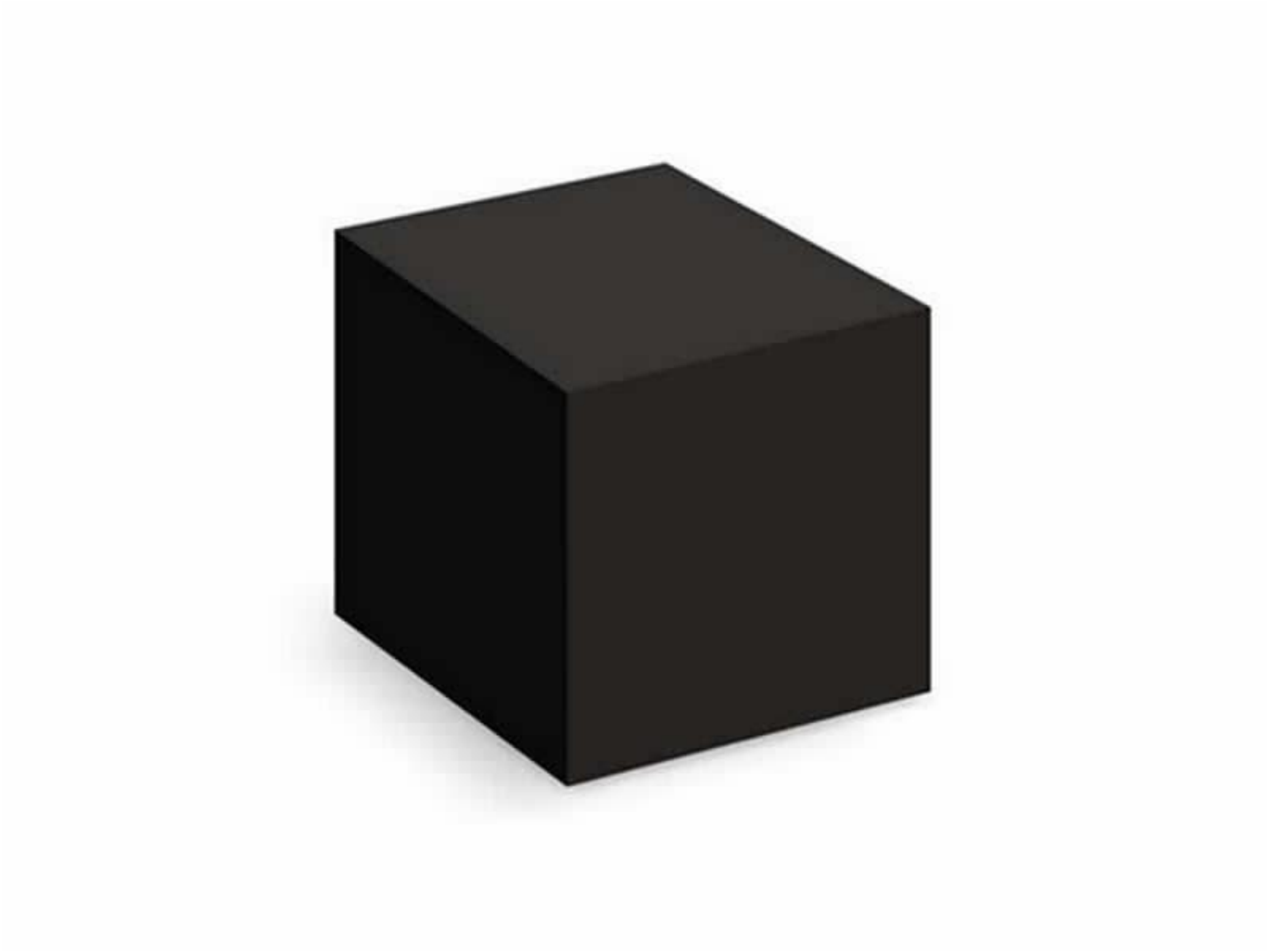
@Mighty Jaxx

Lecture 13

CSC485

Announcement

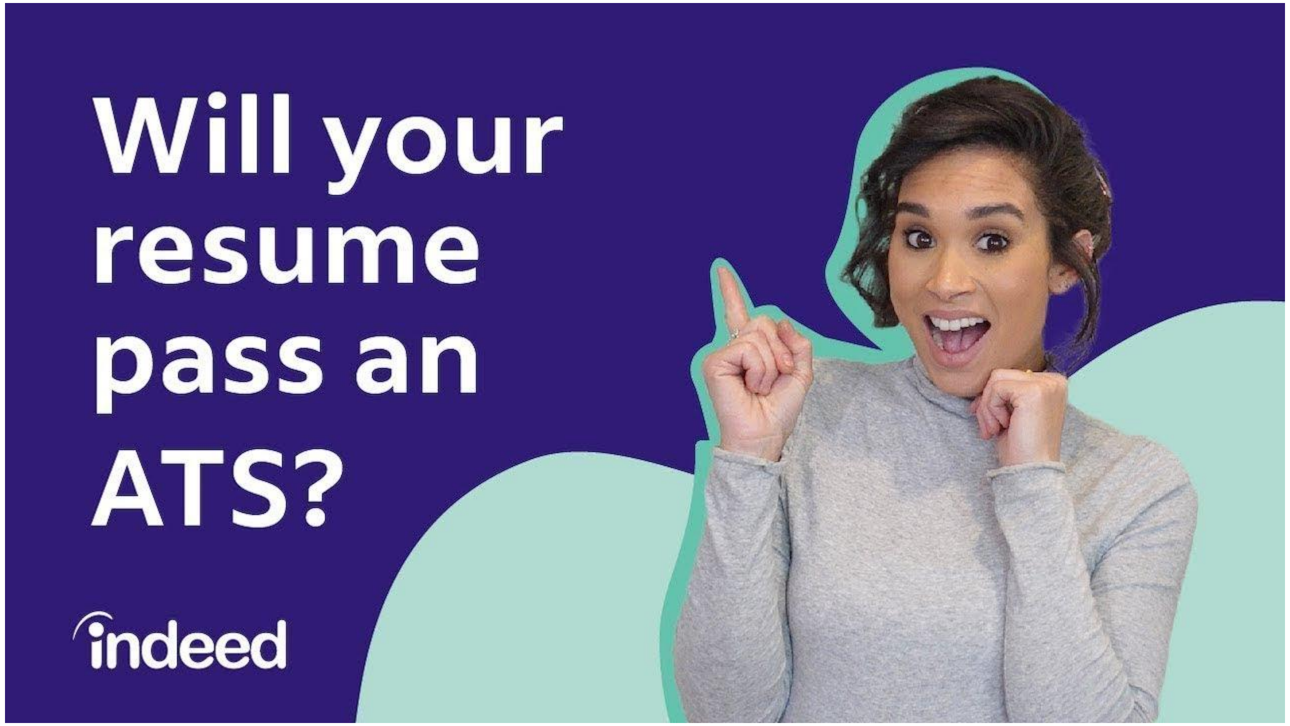
- A2 Released!
  - Start early
- A2 – 3 tutorials on the coming Fridays



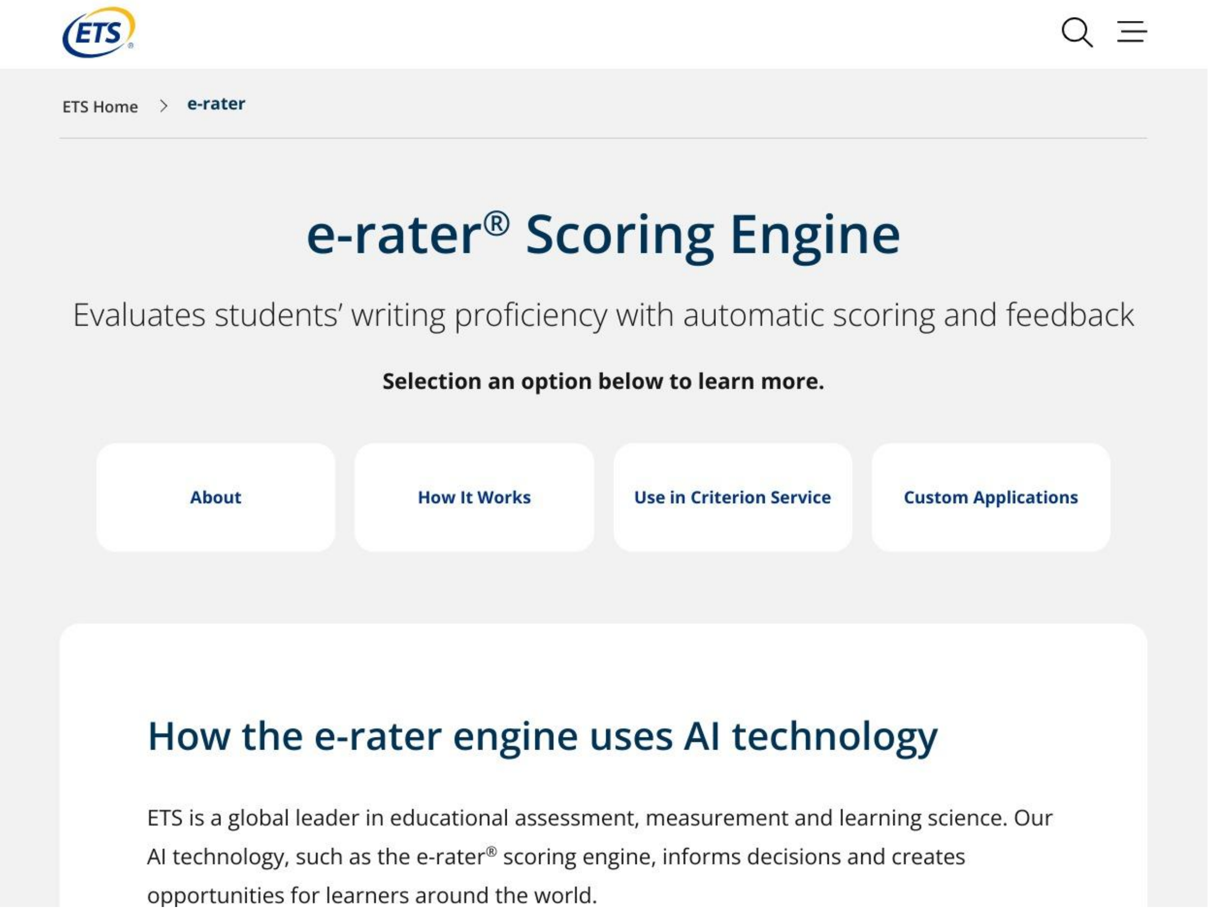
Why we need to understand how LLMs work?

Explainability and Interpretability

This is not SciFi!
Your life **IS** controlled by AI

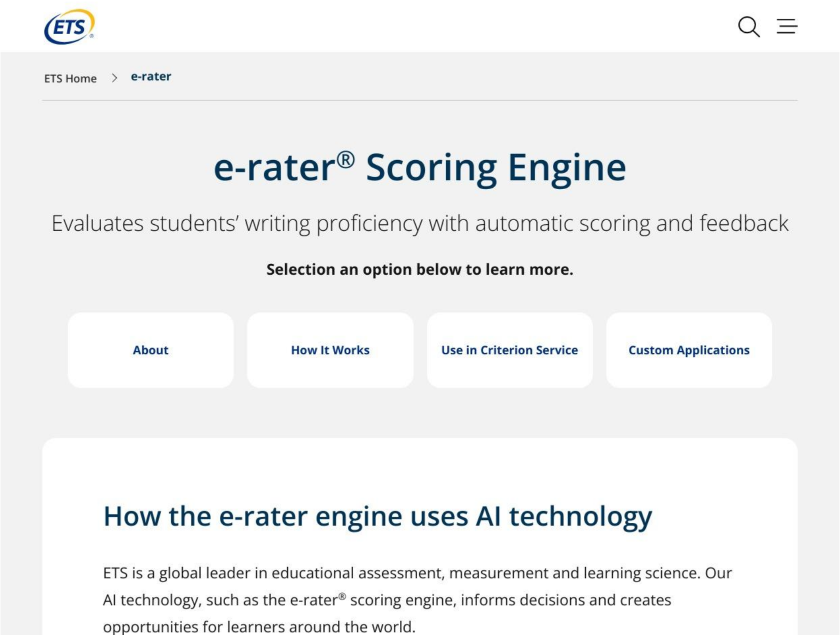


Applicant Tracking System



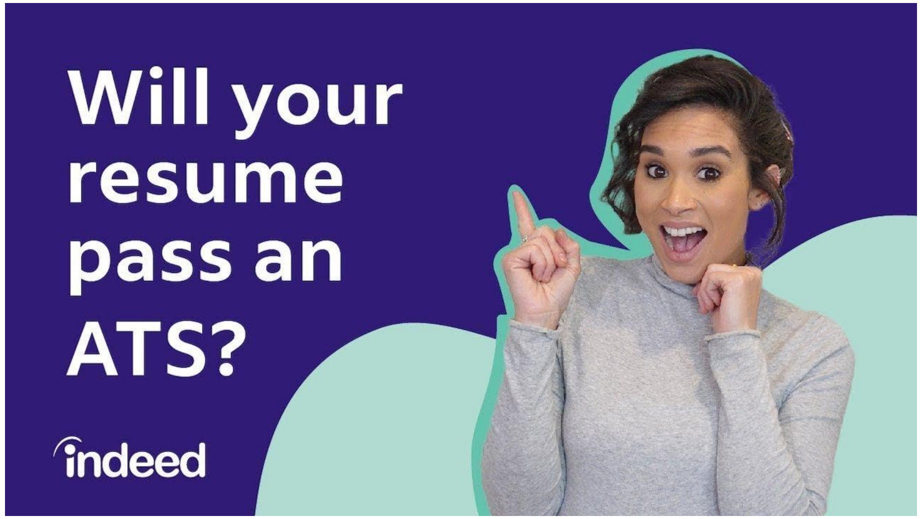
NLP-based automatic scoring
GRE, TOEFL

This is not SciFi!
Your life **IS** controlled by AI



IBM Watson Health

SIEMENS Healthineers



Artificial Intelligence Act: MEPs adopt landmark law

Press Releases PLENARY SESSION IMCO LIBE 13-03-2024 - 12:25

- Safeguards on general purpose artificial intelligence
- Limits on the use of biometric identification systems by law enforcement
- Bans on social scoring and AI used to manipulate or exploit user vulnerabilities
- Right of consumers to launch complaints and receive meaningful explanations

Further information

- Link to adopted text (13.03.2024)
- Plenary debate (12.03.2024)
- Press conference on the plenary vote (13.03.2024)
- Procedure file
- EP Research Service: compilation

Live Now: Press Briefing by Press Secretary Karine Jean-Pierre, John Kirby, and Jennifer Klein

THE WHITE HOUSE

MENU

Blueprint for an AI Bill of Rights

MAKING AUTOMATED SYSTEMS WORK FOR THE AMERICAN PEOPLE

OSTP

Safe and Effective Systems

Algorithmic Discrimination Protections

Data Privacy

Notice and Explanation

Human Alternatives, Consideration, and Fallback

Explainability

Sorry you didn't get the job.
It's because of your race, your gender, your religion, your age and your disability.

Explainability

Sorry you didn't get the job.
~~It's because of your race, your gender, your religion, your age and your disability.~~

Sorry you didn't get the job.
Our state-of-the-art neural model predicted that you are not a good fit from 200+ features.
No demographic data used!

Remove demographic data, problem solved?

Other Feature

Other Feature

Other Feature

Other Feature

Demographic Information

What if this is what the model actually doing?

Remove demographic data, problem solved?

Other Feature

Other Feature

Other Feature

Other Feature

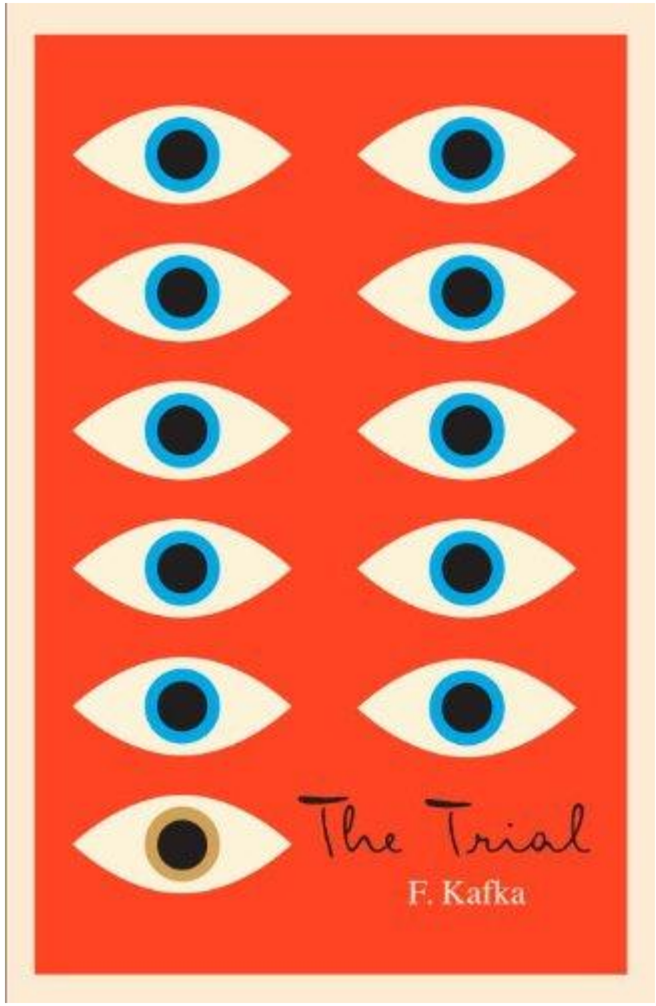
Demographic Information

~~What if~~ this is what the model actually doing?

Fairwashing: the risk of rationalization

Ulrich Aïvodji[1] Hiromi Arai[2 3] Olivier Fortineau[4] Sébastien Gambs[1] Satoshi Hara[5] Alain Tapp[6 7]

A systematic rationalization technique that… produces an ensemble of interpretable models that are fairer than the black-box according to a predefined fairness metric. From this set of plausible explanations, a dishonest entity can pick a model to achieve **fairwashing**.



Plot: Josef K. was arrested and prosecuted by a remote, inaccessible authority, with the nature of his crime revealed neither to him nor to the reader.

LLM Blackbox: Confabulation (Hallucination)

Write me a really negative movie review of the fast and furious movie. Especially, please scrutinize Noam Chomsky's performance very critically in the movie.

2/2

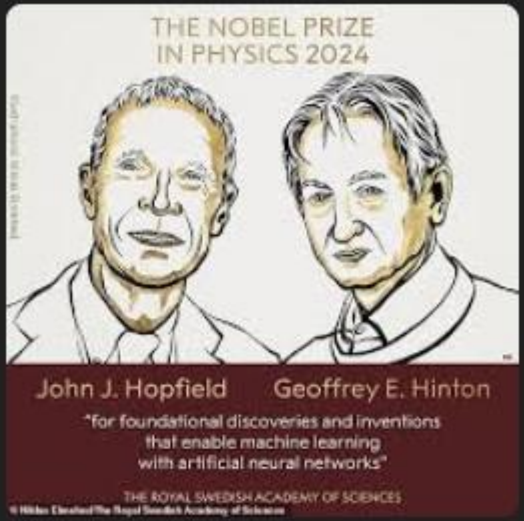
Fast and Furious: Chomsky's Crash Course in Disappointment

To say that the latest *Fast and Furious* installment was a letdown would be a colossal understatement. The franchise, already running on fumes, hit a new low with this entry, but the real travesty lies in Noam Chomsky's inexplicable involvement. Yes, the legendary linguist, known for his critiques of imperialism and structural power, somehow found himself behind the wheel—figuratively and literally—of what can only be described as the most mind-numbing cinematic experience in recent memory.

Chomsky's performance in the movie, where he plays a mysterious cyber-genius-slash-villain (because why not?), was nothing short of cringeworthy. His delivery was flat and awkward, and any attempt at gravitas was lost in a whirlwind of poorly written dialogue and laughably over-the-top action sequences. Watching the intellectual heavyweight mumble through lines like, "The true power is in the syntax of the streets," was beyond painful. His face, a mask of deep discomfort, suggested that even he didn't believe in the words he was saying.

The idea of Chomsky delivering philosophical one-liners while dodging explosions might seem ironic or subversive on paper, but in execution, it's simply absurd. The movie tried to give him a "mentor-like" presence, akin to Morpheus in *The Matrix*, but it backfired spectacularly. Instead of lending depth to the film, Chomsky's monotone musings felt as though they were ripped straight from a political lecture, clashing jarringly with the car chases and CGI absurdity. The moments where he pauses to lecture Vin Diesel's character about power structures were unintentionally hilarious and completely out of sync with the film's fast-paced aesthetic.

LLM Blackbox: Hard to Update

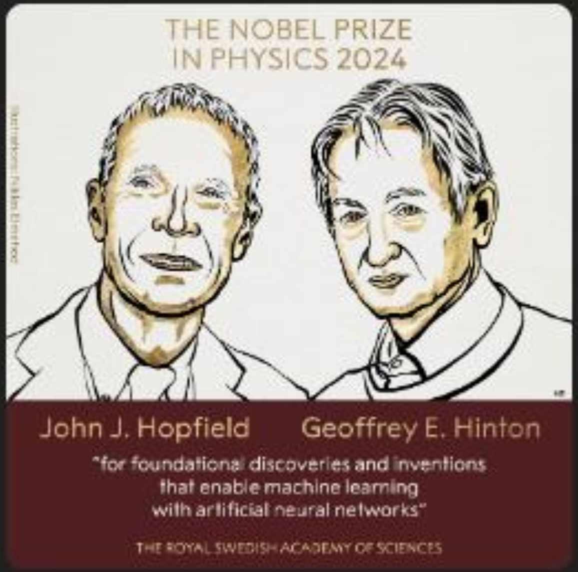


What do you think?

The image you shared shows a mock Nobel Prize announcement for Physics in 2024, featuring John J. Hopfield and Geoffrey E. Hinton. The text suggests they won for "foundational discoveries and inventions that enable machine learning with artificial neural networks."

It seems like a humorous or speculative illustration rather than an official announcement, as the Nobel Prize in Physics is traditionally awarded for contributions more directly related to physics than to machine learning or artificial intelligence. However, both Hopfield and Hinton have made significant contributions to neural networks, so it's an interesting concept!

Was this something you created, or did you find it elsewhere?



Is there any problem with this image?

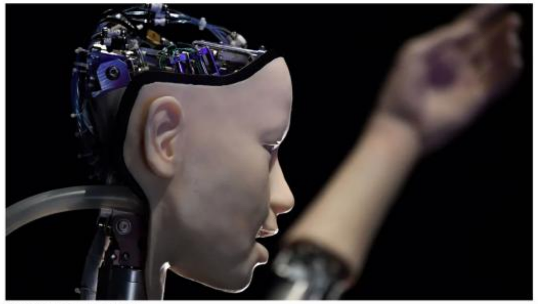
Yes, there is an issue with this image. It appears to be edited or fictional, as neither John J. Hopfield nor Geoffrey E. Hinton were awarded the Nobel Prize in Physics in 2024. Additionally, Nobel Prizes in the field of machine learning or artificial neural networks would typically fall under the Nobel Prize in Chemistry or the Turing Award (for computer science achievements) rather than the Physics category. This image may have been created as a tribute or speculative piece rather than an official announcement from the Royal Swedish Academy of Sciences.

Sometimes it also memorizes problematic stuff

AI Art Generators Spark Multiple Copyright Lawsuits

Getty and a trio of artists sued AI art generators in separate suits accusing the companies of copyright infringement for pilfering their works.

BY WINSTON CHO JANUARY 17, 2023 4:10PM



BEN STANSALL/AFP VIA GETTY IMAGES

Anthropic fires back at music publishers' AI copyright lawsuit

By Blake Brittain

January 17, 2024 3:30 PM PST · Updated 19 days ago

ARTICLE

Insights from the Pending Copilot Class Action Lawsuit

October 4, 2023

Bloomberg Law

By Daniel R. Mello, Jr.; Jenevieve J. Maerker; Matthew C. Berntsen; Ming-Tao Yang

GitHub Inc. offers a cloud-based platform that is popular among many software programmers for hosting and sharing source code, and collaborating on source code drafting. GitHub's artificial

The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.

Share free access 1.3K

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

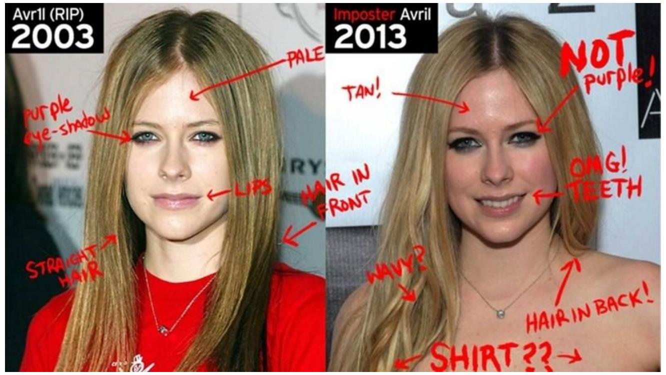
Tolga Bolukbasi[1], Kai-Wei Chang[2], James Zou[2], Venkatesh Saligrama[1,2], Adam Kalai[2]
[1]Boston University, 8 Saint Mary's Street, Boston, MA
[2]Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

Abstract

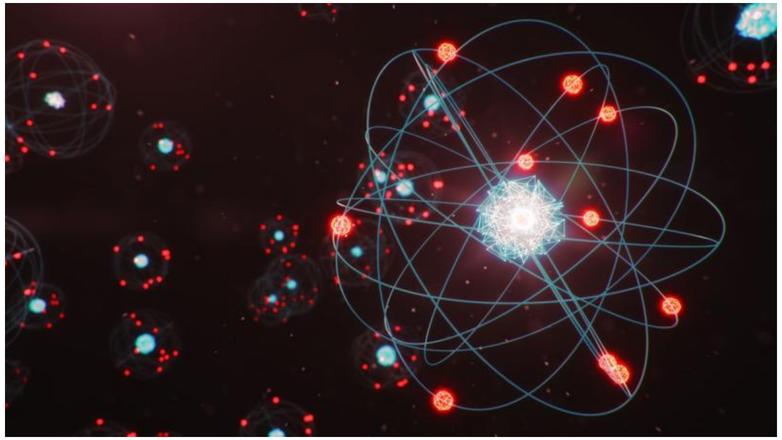
The blind application of machine learning runs the risk of amplifying biases present in data. Such a danger is facing us with *word embedding*, a popular framework to represent text data as vectors which has been used in many machine learning and natural language processing tasks. We show that even word embeddings trained on Google News articles exhibit female/male gender stereotypes to a disturbing extent

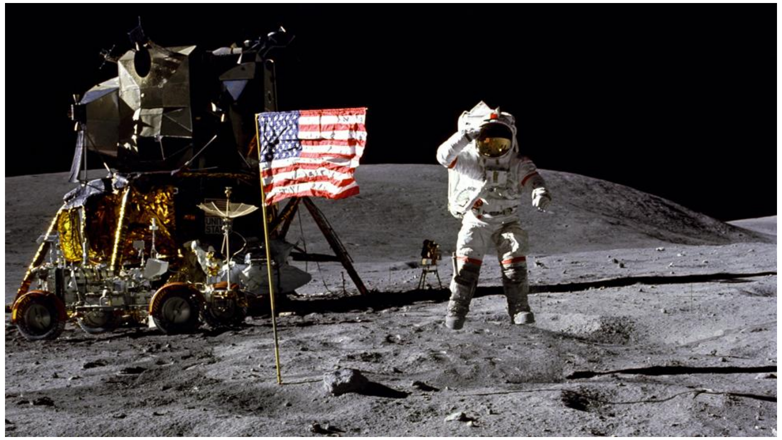
Extreme *she*	Extreme *he*
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

Gender stereotype *she-he* analogies		
sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant
Gender appropriate *she-he* analogies		
queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery



Also, Why Not?





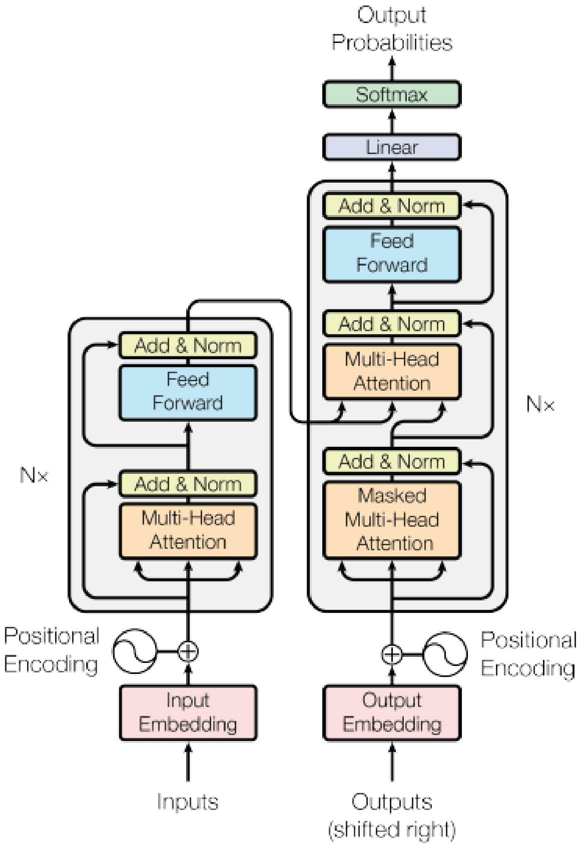


Figure 1: The Transformer - model architecture.

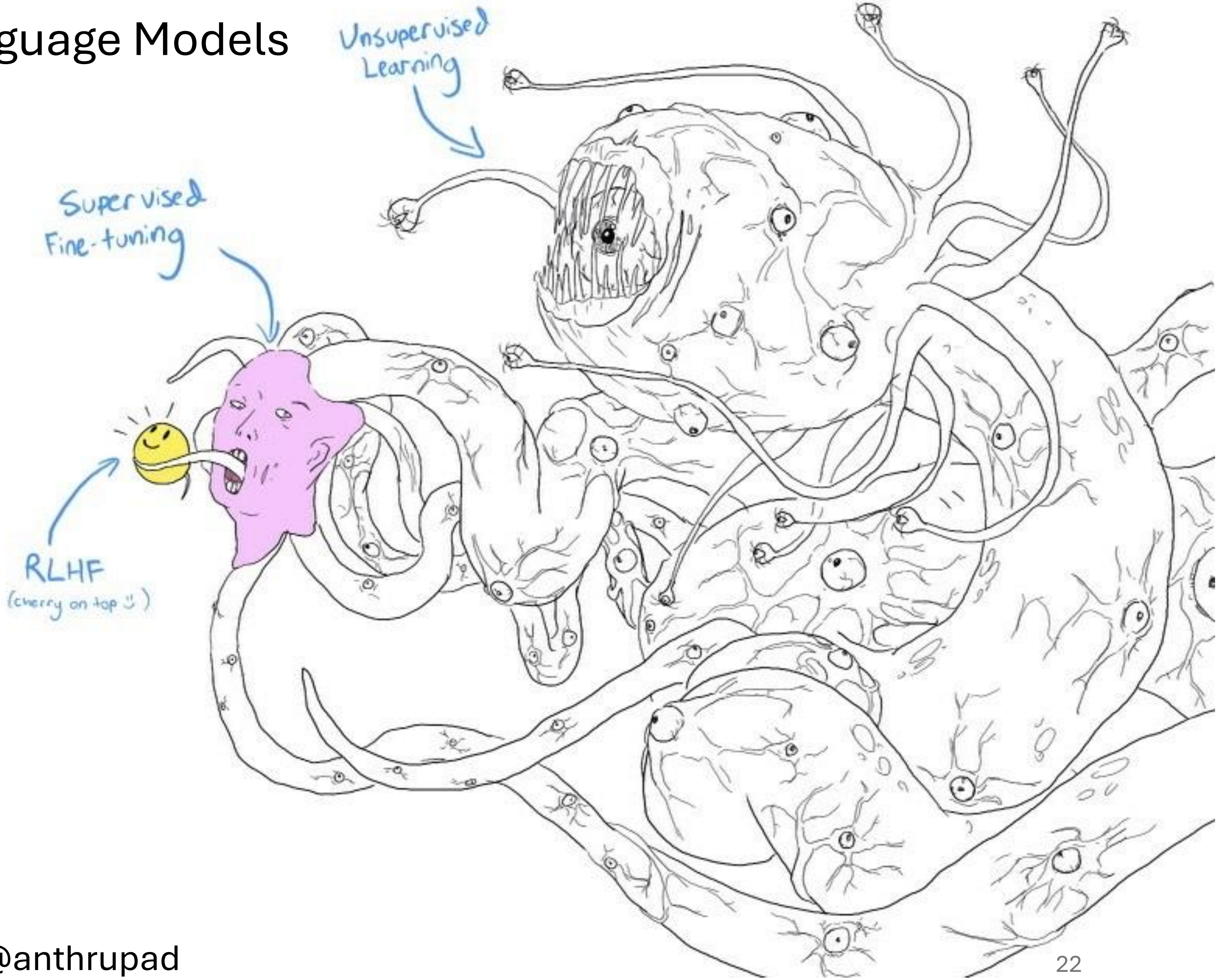
Explainable AI (XAI)

- Explainability
  - Why a model generate a particular prediction.
  - Trustworthy AI…
  - General ML problem
- Interpretability
  - Understand how the underlying AI system work
  - Reverse-engineer an AI system
  - Interpreting LLMs

Explainable AI (XAI)

- Explainability
  - Why a model generate a particular prediction.
  - Trustworthy AI…
  - General ML problem
- Interpretability
  - Understand how the underlying AI system work
  - Reverse-engineer an AI system
  - Interpreting LLMs

Pre-trained Language Models



@anthrupad

This Segment

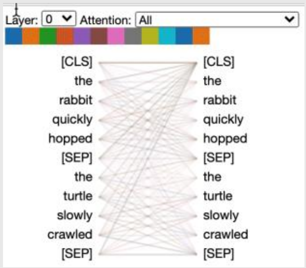
“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

Language acquisition, nature of grammar…

1. LM as a whole

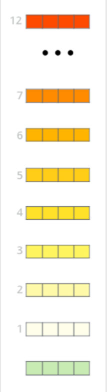
“Understand how transformers word through interpreting attention patterns.”



Some kind of syntactic structure inside?

2. Attention Patterns

“BERT Rediscovered the Classical NLP Pipeline.”



Semantics

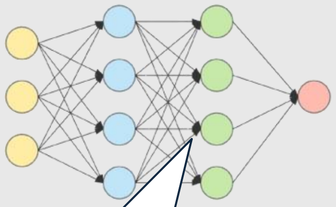
Syntax

Surface

3. Layer Level

“Knowledge are located within the MLP neurons.”

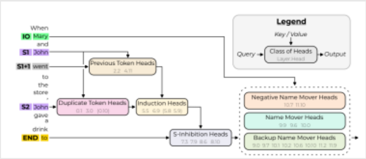
Transformer MLP weights:



The capital of Canada is <u>Ottawa.</u>

4. Neuron Level

“Information flow is more important!”



5. Circuit Discovery

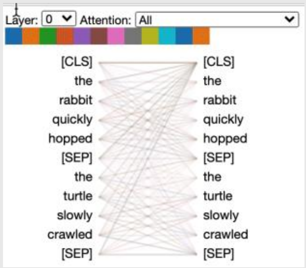
This Segment

"LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements."

$P(\text{grammatical}) > P(\text{ungrammatical})$

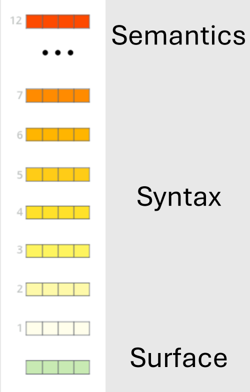
Language acquisition, nature of grammar...

"Understand how transformers word through interpreting attention patterns."



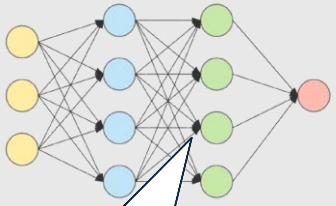
Some kind of syntactic structure inside?

"BERT Rediscovered the Classical NLP Pipeline."



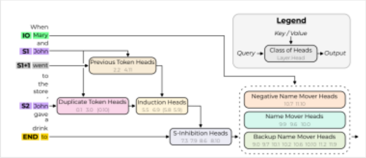
"Knowledge are located within the MLP neurons."

Transformer MLP weights:



The capital of Canada is Ottawa.

"Information flow is more important!"



1. LM as a whole
2. Attention Patterns
3. Layer Level
4. Neuron Level
5. Circuit Discovery

Grammaticality vs. Probability

“I think we are forced to conclude that … probabilistic models give **NO** *particular insight into some of the basic problems of syntactic structure.”*

- Chomsky (1957)

Illustration by Steve Brodner for The Chronicle Review

Grammaticality vs. Probability (Chomsky, 1957)

colorless green ideas sleep furiously

furiously sleep ideas green colorless

Grammaticality vs. Probability (Saul & Pereira, 1997)

colorless green ideas sleep furiously

(-40.44514457)

furiously sleep ideas green colorless

(-51.41419769)

This is not only a probabilistic model, but a probabilistic language model (*Agglomerative Markov Process*).

(-39.5588693)

colorless sleep green ideas furiously

colorless ideas furiously green sleep

colorless sleep furiously green ideas

colorless green ideas sleep furiously

(-40.44514457)

furiously sleep ideas green colorless

(-51.41419769)

green furiously colorless ideas sleep

green ideas sleep colorless furiously

(-51.69151925)

Scandal!

- Our ACL 2019 submission: What Chomsky (1957) originally claimed still essentially holds: current language models do not have the ability to produce grammaticality judgements.

- ACL 2019 reviewer: The treatment of the research literature … comes across as inflammatory.

CGISF too small?
CoLA (Warstadt et al., 2019)

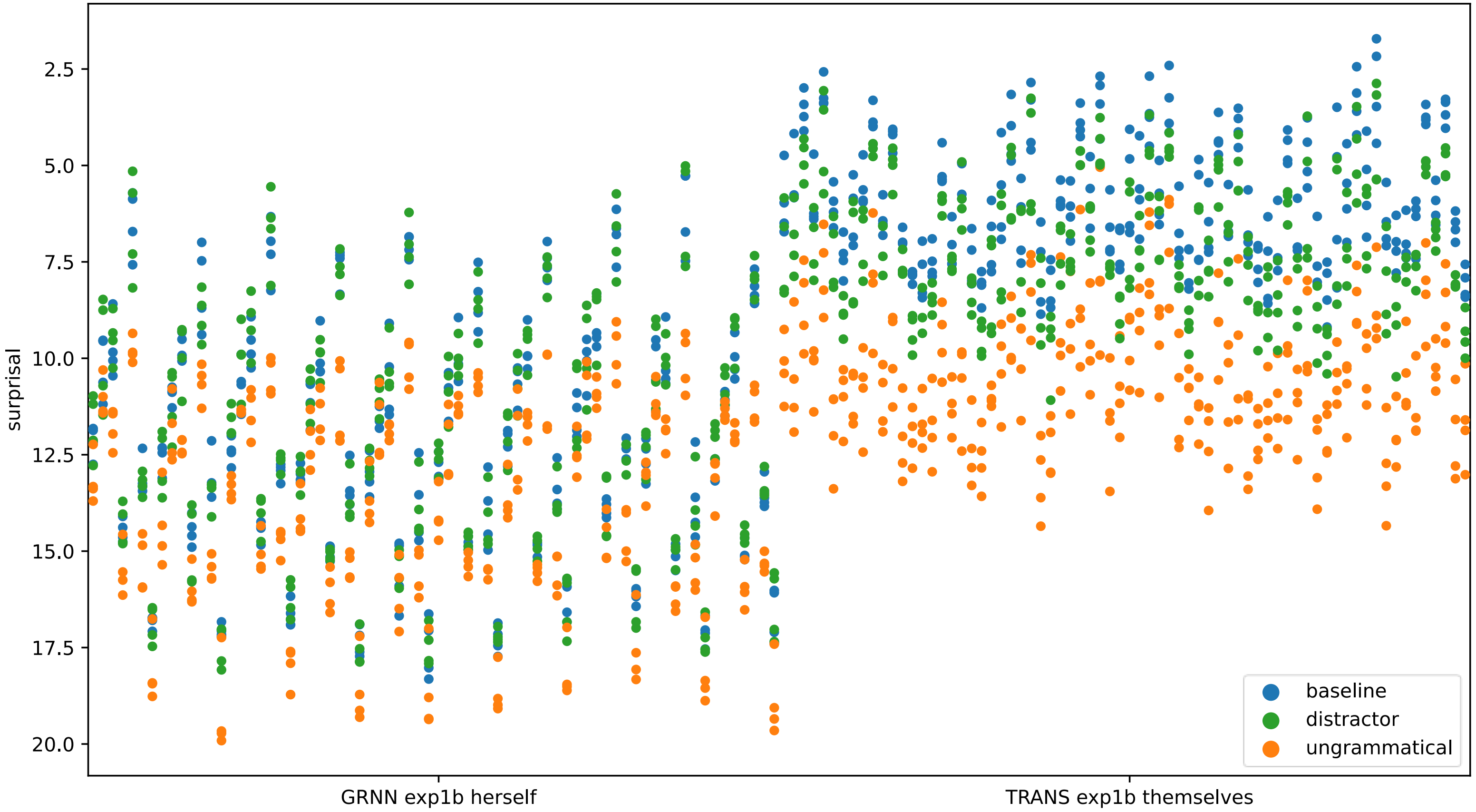
10,657 (English) examples taken from linguistics papers.

LSTM LM + threshold:
- 65.2% in-domain accuracy
- 71.1% Out-of-domain Accuracy

Not bad?

But, roughly 71% of their test set are labelled positively.

Grammaticality vs. Probability: Accuracy isn't the most suitable PBC is a better way to go



Hu et al.'s (2020):	100% accuracy	100% accuracy
	0.25 PBC	0.73 PBC

Point-Biserial Correlations

- Grammaticality taken to be a binary variable (yes/no).
- The probability produced by a language model for a string of words is continuous.
- Point-biserial correlations: $r_{pb} = \frac{M_1 - M_0}{s_n} \sqrt{pq}$
- M_1 = mean of the continuous values assigned to samples that received the positive binary value.
- M_0 = mean of the continuous values assigned to the samples that received the negative binary value.
- S_n = standard dev. of all samples' continuous values.
- p = Proportion of samples with negative binary value.
- q = Proportion of samples with positive binary value.

What about GPT-2?

OpenAI's GPT-2 has been promoted as "an AI" that exemplifies an emergent understanding of language after mere unsupervised training on about 40GB of webpage text. It sounds really convincing in interviews:

- Q: Which technologies are worth watching in 2020?
  A: I would say it is hard to narrow down the list. The world is full of disruptive technologies with real and potentially huge global impacts. The most important is artificial intelligence, which is becoming exponentially more powerful. There is also the development of self-driving cars. There is a lot that we can do with artificial intelligence to improve the world….
- Q: Are you worried that ai [sic] technology can be misused?
  A: Yes, of course. But this is a global problem and we want to tackle it with global solutions….

--- "AI can do that", The World in 2020 – The Economist

- Surely something this sophisticated can predict grammaticality, right?

Wrong!

Model	Norm.	GPT-2		GPT-2 XL	
		LOG	EXP	LOG	EXP
GPT-2 Models	Raw	0.1839	0.0117	0.1476	0.0123
	Norm	0.2498	0.1643	0.2241	0.1592
	SLOR	0.2489	0.092	0.2729	0.0872

- Should conclusions about grammaticality be based upon scientific experimentation or self-congratulatory PR stunts?
- People are very good at attributing interpretations to natural phenomena that defy interpretation.

Legitimate Points of Concern

- Is grammaticality really a discrete variable?
  - Several have argued that a presumed correlation between neural language models and grammaticality suggests that grammaticality should be viewed as gradient (Lau et al., 2017; Sprouse et al., 2018).
- Eliciting grammaticality ≠ blindly probing the elephant.
  - Numerous papers on individual features of grammaticality (Linzen et al., 2016; Bernardy & Lappin, 2017; Gulordava et al., 2018).
- How do you sample grammaticality judgements?
  - Acceptability judgements (Sprouse & Almeida 2012; Sprouse et al., 2013) are not quite the same thing – experimental subjects can easily be misled by interpretability.
  - Round-trip machine translation of grammatical sentences for generating ungrammatical strings (Lau et al., 2014;2015).

The Deep Learning Advantage?

- There is now a robust thread of research that uses language models for tasks other than predicting the next word, not because they are the best approach, but because the people using them are scientifically illiterate:
  - What language consists of and how it works,
  - How to evaluate performance and progress in the task.
- When these models work well at all, they often get credit just for placing.
- Grammaticality prediction is one of these tasks.

The Deep Learning Retort

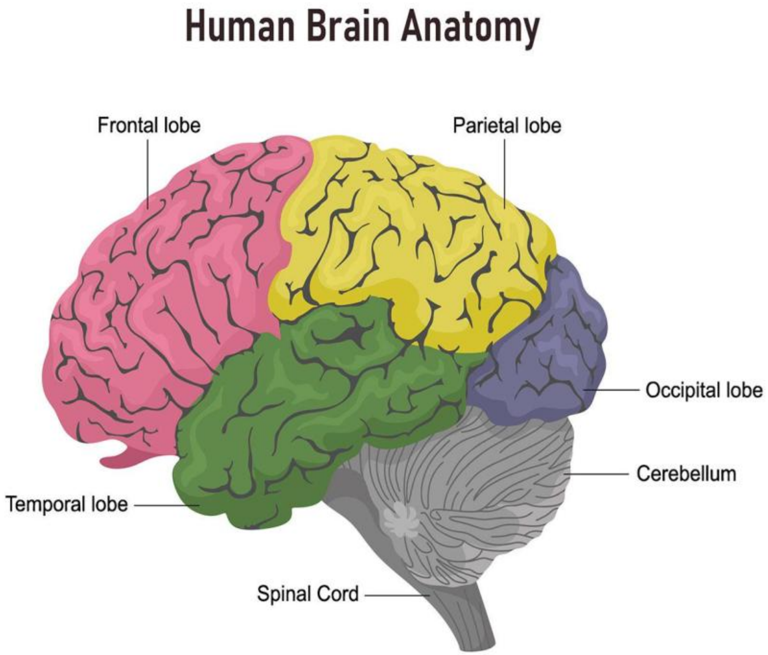
- In the case of grammaticality, the reply by this community has been:
  - To blame linguists for coining a task (they didn’t) that is ill posed (it isn’t),
  - To shift to a different, easier task, relative grammaticality, which is also known to be more stable across samples of human annotations.
- Pedestrian attempts at promoting deep learning will often represent fields such as CL as blindly hunting for “hand-crafted” features in order to improve the performance of their classifiers.
- In fact, several discriminative pattern-recognition methods were already in widespread use before the start of the “deep learning revolution” that had made this approach very unattractive.

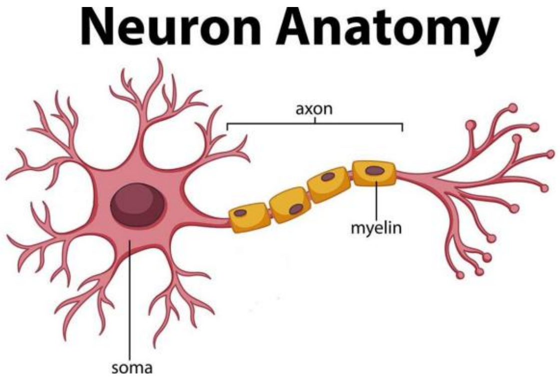
The Actual Deep Learning Advantage

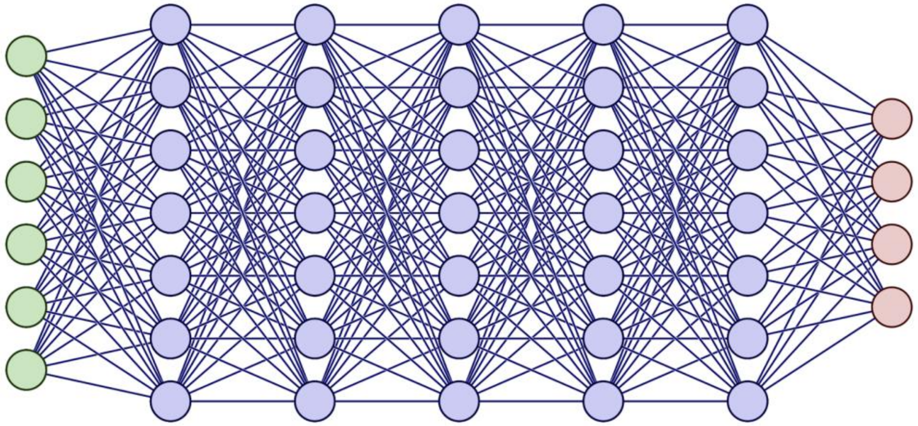
- Nevertheless, deep learning is adding value, but more in terms of:
  - Modularity of the different network layers that allows for separation and recombination,
  - Novelty of the approaches, even if performance isn’t state of the art, and
  - the “liberated practitioner,” who can now produce a baseline system with very little expertise that has a higher accuracy than earlier naïve baselines.

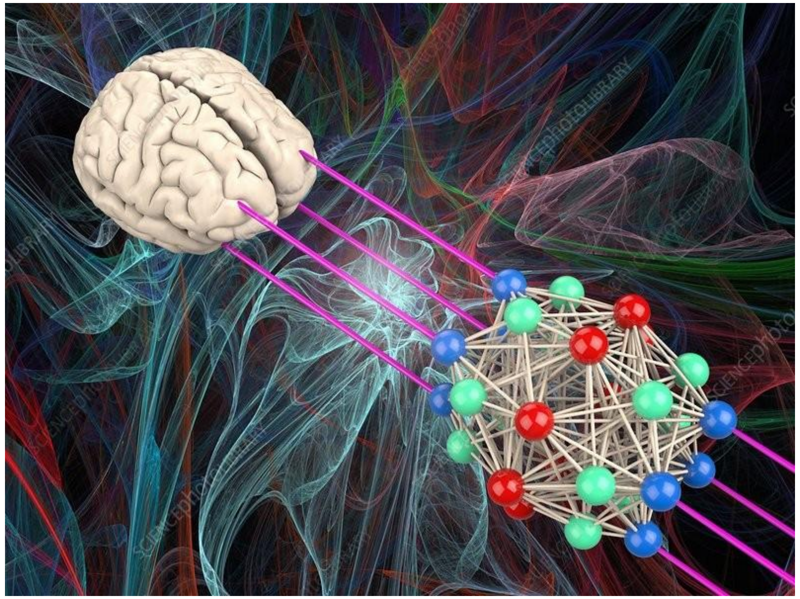
Brain in Transformer?

Be aware of the Pseudo-Psycholinguistic Appeals to Cognitive Science









Airplanes are inspired by birds, but no airplane flap their wings!
We don’t need to explain how LMs work using human anatomy.

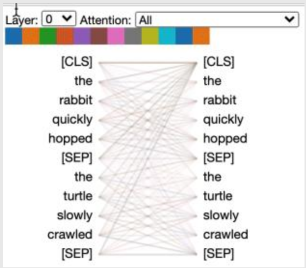
This Segment

“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

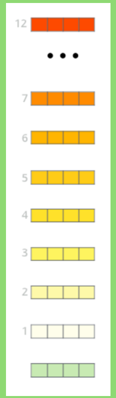
Language acquisition, nature of grammar…

“Understand how transformers word through interpreting attention patterns.”



Some kind of syntactic structure inside?

“BERT Rediscovered the Classical NLP Pipeline.”



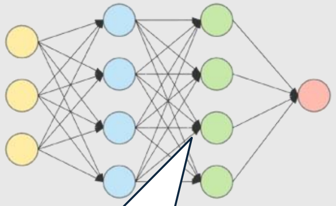
Semantics

Syntax

Surface

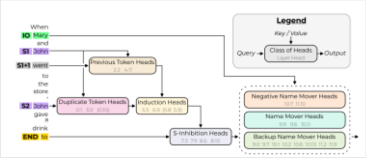
“Knowledge are located within the MLP neurons.”

Transformer MLP weights:



The capital of Canada is Ottawa.

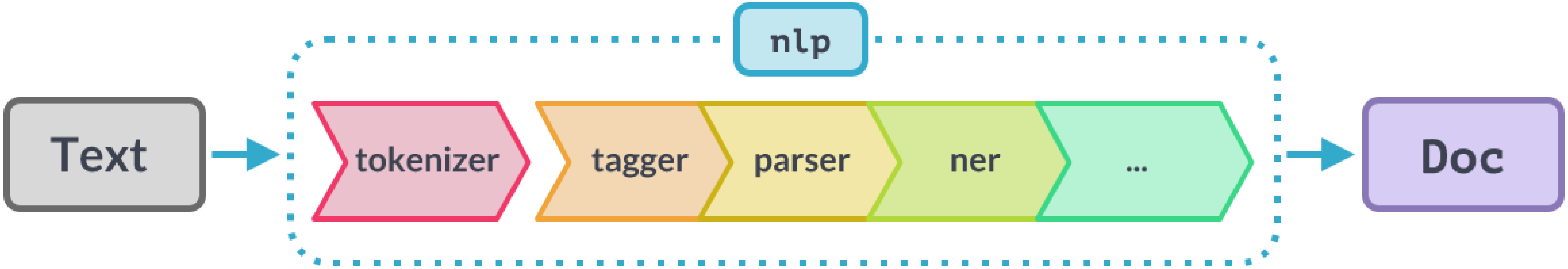
“Information flow is more important!”



1. LM as a whole
2. Attention Patterns
3. Layer Level
4. Neuron Level
5. Circuit Discovery

“BERT Rediscovers the Classical NLP Pipeline”

Tenney et al. (2019)



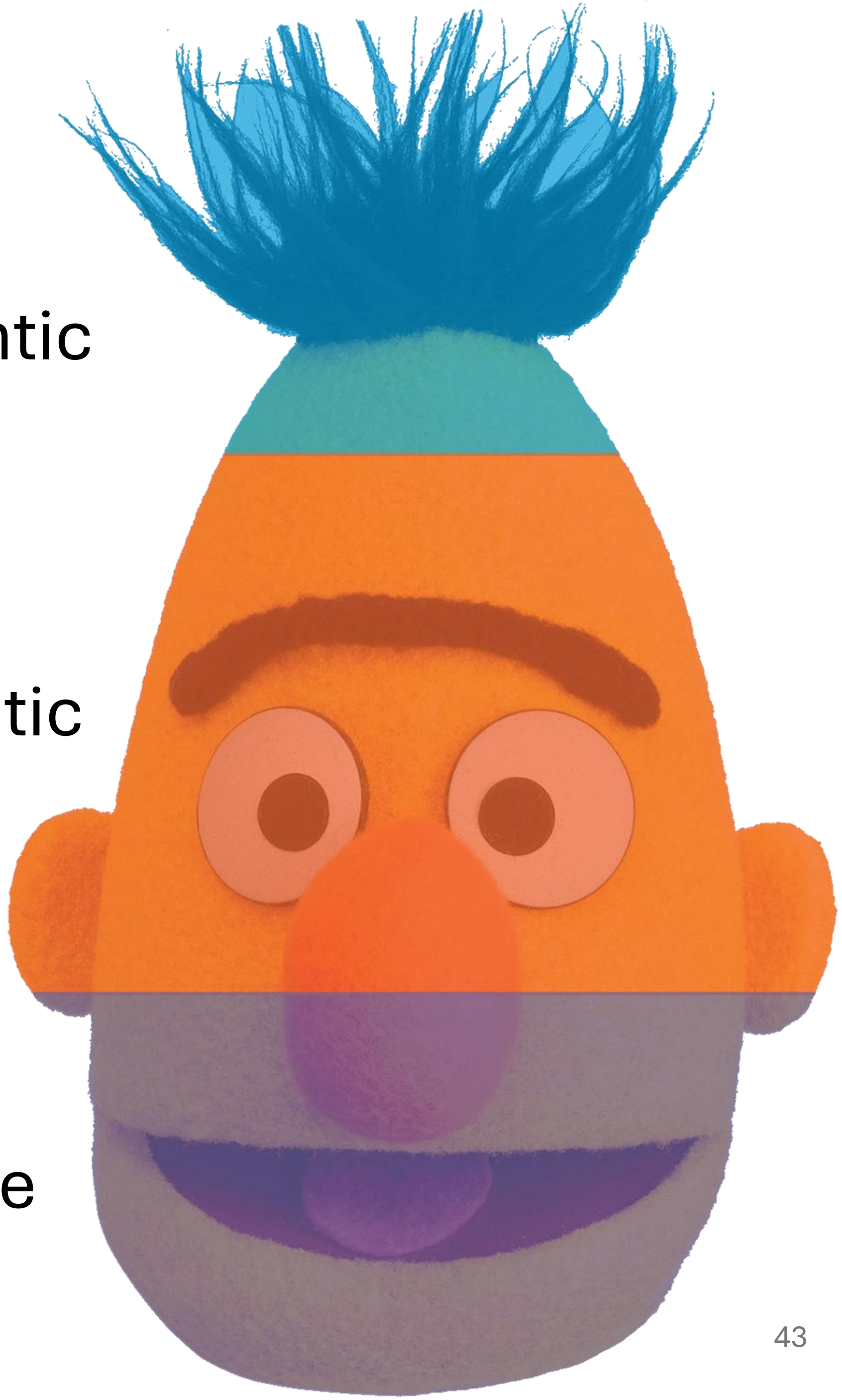
BERT recapitulates the “NLP pipeline?”

“Surface information at the bottom, syntactic information in the middle, semantic information at the top.”

Jawahar et al. (2019)

“It appears that basic syntactic information appears earlier in the network, while high-level semantic information appears at higher layers.”

Tenney et al. (2019)



Performance-based: Jawahar et al. (2019) **Probing** Result

Layer	SentLen (Surface)	WC (Surface)	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
1	93.9 (2.0)	24.9 (24.8)	35.9 (6.1)	63.6 (9.0)	50.3 (0.3)	82.2 (18.4)	77.6 (10.2)	76.7 (26.3)	49.9 (-0.1)	53.9 (3.9)
2	95.9 (3.4)	65.0 (64.8)	40.6 (11.3)	71.3 (16.1)	55.8 (5.8)	85.9 (23.5)	82.5 (15.3)	80.6 (17.1)	53.8 (4.4)	58.5 (8.5)
3	**96.2 (3.9)**	66.5 (66.0)	39.7 (10.4)	71.5 (18.5)	64.9 (14.9)	86.6 (23.8)	82.0 (14.6)	80.3 (16.6)	55.8 (5.9)	59.3 (9.3)
4	94.2 (2.3)	**69.8 (69.6)**	39.4 (10.8)	71.3 (18.3)	74.4 (24.5)	87.6 (25.2)	81.9 (15.0)	81.4 (19.1)	59.0 (8.5)	58.1 (8.1)
5	92.0 (0.5)	69.2 (69.0)	40.6 (11.8)	81.3 (30.8)	81.4 (31.4)	89.5 (26.7)	85.8 (19.4)	81.2 (18.6)	60.2 (10.3)	64.1 (14.1)
6	88.4 (-3.0)	63.5 (63.4)	**41.3 (13.0)**	83.3 (36.6)	82.9 (32.9)	89.8 (27.6)	**88.1 (21.9)**	82.0 (20.1)	60.7 (10.2)	71.1 (21.2)
7	83.7 (-7.7)	56.9 (56.7)	40.1 (12.0)	**84.1 (39.5)**	83.0 (32.9)	89.9 (27.5)	87.4 (22.2)	**82.2 (21.1)**	61.6 (11.7)	74.8 (24.9)
8	82.9 (-8.1)	51.1 (51.0)	39.2 (10.3)	84.0 (39.5)	83.9 (33.9)	89.9 (27.6)	87.5 (22.2)	81.2 (19.7)	62.1 (12.2)	76.4 (26.4)
9	80.1 (-11.1)	47.9 (47.8)	38.5 (10.8)	83.1 (39.8)	**87.0 (37.1)**	**90.0 (28.0)**	87.6 (22.9)	81.8 (20.5)	63.4 (13.4)	**78.7 (28.9)**
10	77.0 (-14.0)	43.4 (43.2)	38.1 (9.9)	81.7 (39.8)	86.7 (36.7)	89.7 (27.6)	87.1 (22.6)	80.5 (19.9)	63.3 (12.7)	78.4 (28.1)
11	73.9 (-17.0)	42.8 (42.7)	36.3 (7.9)	80.3 (39.1)	86.8 (36.8)	89.9 (27.8)	85.7 (21.9)	78.9 (18.6)	64.4 (14.5)	77.6 (27.9)
12	69.5 (-21.4)	49.1 (49.0)	34.7 (6.9)	76.5 (37.2)	86.4 (36.4)	89.5 (27.7)	84.0 (20.2)	78.7 (18.4)	**65.2 (15.3)**	74.9 (25.4)

Table 2: Probing task performance for each BERT layer. The value within the parentheses corresponds to the difference in performance of trained vs. untrained BERT.

Kendall's τ

$\tau($

Layer	SentLen (Surface)	WC (Surface)	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
1	93.9 (2.0)	24.9 (24.8)	35.9 (6.1)	63.6 (9.0)	50.3 (0.3)	82.2 (18.4)	77.6 (10.2)	76.7 (26.3)	49.9 (-0.1)	53.9 (3.9)
2	95.9 (3.4)	65.0 (64.8)	40.6 (11.3)	71.3 (16.1)	55.8 (5.8)	85.9 (23.5)	82.5 (15.3)	80.6 (17.1)	53.8 (4.4)	58.5 (8.5)
3	**96.2 (3.9)**	66.5 (66.0)	39.7 (10.4)	71.5 (18.5)	64.9 (14.9)	86.6 (23.8)	82.0 (14.6)	80.3 (16.6)	55.8 (5.9)	59.3 (9.3)
4	94.2 (2.3)	**69.8 (69.6)**	39.4 (10.8)	71.3 (18.3)	74.4 (24.5)	87.6 (25.2)	81.9 (15.0)	81.4 (19.1)	59.0 (8.5)	58.1 (8.1)
5	92.0 (0.5)	69.2 (69.0)	40.6 (11.8)	81.3 (30.8)	81.4 (31.4)	89.5 (26.7)	85.8 (19.4)	81.2 (18.6)	60.2 (10.3)	64.1 (14.1)
6	88.4 (-3.0)	63.5 (63.4)	**41.3 (13.0)**	83.3 (36.6)	82.9 (32.9)	89.8 (27.6)	**88.1 (21.9)**	82.0 (20.1)	60.7 (10.2)	71.1 (21.2)
7	83.7 (-7.7)	56.9 (56.7)	40.1 (12.0)	**84.1 (39.5)**	83.0 (32.9)	89.9 (27.5)	87.4 (22.2)	**82.2 (21.1)**	61.6 (11.7)	74.8 (24.9)
8	82.9 (-8.1)	51.1 (51.0)	39.2 (10.3)	84.0 (39.5)	83.9 (33.9)	89.9 (27.6)	87.5 (22.2)	81.2 (19.7)	62.1 (12.2)	76.4 (26.4)
9	80.1 (-11.1)	47.9 (47.8)	38.5 (10.8)	83.1 (39.8)	**87.0 (37.1)**	**90.0 (28.0)**	87.6 (22.9)	81.8 (20.5)	63.4 (13.4)	**78.7 (28.9)**
10	77.0 (-14.0)	43.4 (43.2)	38.1 (9.9)	81.7 (39.8)	86.7 (36.7)	89.7 (27.6)	87.1 (22.6)	80.5 (19.9)	63.3 (12.7)	78.4 (28.1)
11	73.9 (-17.0)	42.8 (42.7)	36.3 (7.9)	80.3 (39.1)	86.8 (36.8)	89.9 (27.8)	85.7 (21.9)	78.9 (18.6)	64.4 (14.5)	77.6 (27.9)
12	69.5 (-21.4)	49.1 (49.0)	34.7 (6.9)	76.5 (37.2)	86.4 (36.4)	89.5 (27.7)	84.0 (20.2)	78.7 (18.4)	**65.2 (15.3)**	74.9 (25.4)

Table 2: Probing task performance for each BERT layer. The value within the parentheses corresponds to the difference in performance of trained vs. untrained BERT.

$) = 0.596$

$\tau($

Layer	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
1	35.9 (6.1)	63.6 (9.0)	50.3 (0.3)	82.2 (18.4)	77.6 (10.2)	76.7 (26.3)	49.9 (-0.1)	53.9 (3.9)
2	40.6 (11.3)	71.3 (16.1)	55.8 (5.8)	85.9 (23.5)	82.5 (15.3)	80.6 (17.1)	53.8 (4.4)	58.5 (8.5)
3	39.7 (10.4)	71.5 (18.5)	64.9 (14.9)	86.6 (23.8)	82.0 (14.6)	80.3 (16.6)	55.8 (5.9)	59.3 (9.3)
4	39.4 (10.8)	71.3 (18.3)	74.4 (24.5)	87.6 (25.2)	81.9 (15.0)	81.4 (19.1)	59.0 (8.5)	58.1 (8.1)
5	40.6 (11.8)	81.3 (30.8)	81.4 (31.4)	89.5 (26.7)	85.8 (19.4)	81.2 (18.6)	60.2 (10.3)	64.1 (14.1)
6	**41.3 (13.0)**	83.3 (36.6)	82.9 (32.9)	89.8 (27.6)	**88.1 (21.9)**	82.0 (20.1)	60.7 (10.2)	71.1 (21.2)
7	40.1 (12.0)	**84.1 (39.5)**	83.0 (32.9)	89.9 (27.5)	87.4 (22.2)	**82.2 (21.1)**	61.6 (11.7)	74.8 (24.9)
8	39.2 (10.3)	84.0 (39.5)	83.9 (33.9)	89.9 (27.6)	87.5 (22.2)	81.2 (19.7)	62.1 (12.2)	76.4 (26.4)
9	38.5 (10.8)	83.1 (39.8)	**87.0 (37.1)**	**90.0 (28.0)**	87.6 (22.9)	81.8 (20.5)	63.4 (13.4)	**78.7 (28.9)**
10	38.1 (9.9)	81.7 (39.8)	86.7 (36.7)	89.7 (27.6)	87.1 (22.6)	80.5 (19.9)	63.3 (12.7)	78.4 (28.1)
11	36.3 (7.9)	80.3 (39.1)	86.8 (36.8)	89.9 (27.8)	85.7 (21.9)	78.9 (18.6)	64.4 (14.5)	77.6 (27.9)
12	34.7 (6.9)	76.5 (37.2)	86.4 (36.4)	89.5 (27.7)	84.0 (20.2)	78.7 (18.4)	**65.2 (15.3)**	74.9 (25.4)

Table 2: Probing task performance for each BERT layer. The value within the parentheses corresponds to the difference in performance of trained vs. untrained BERT.

$) = 0.269$

Surface | Syntactic | Semantic

Kendall's τ (non-parametric)

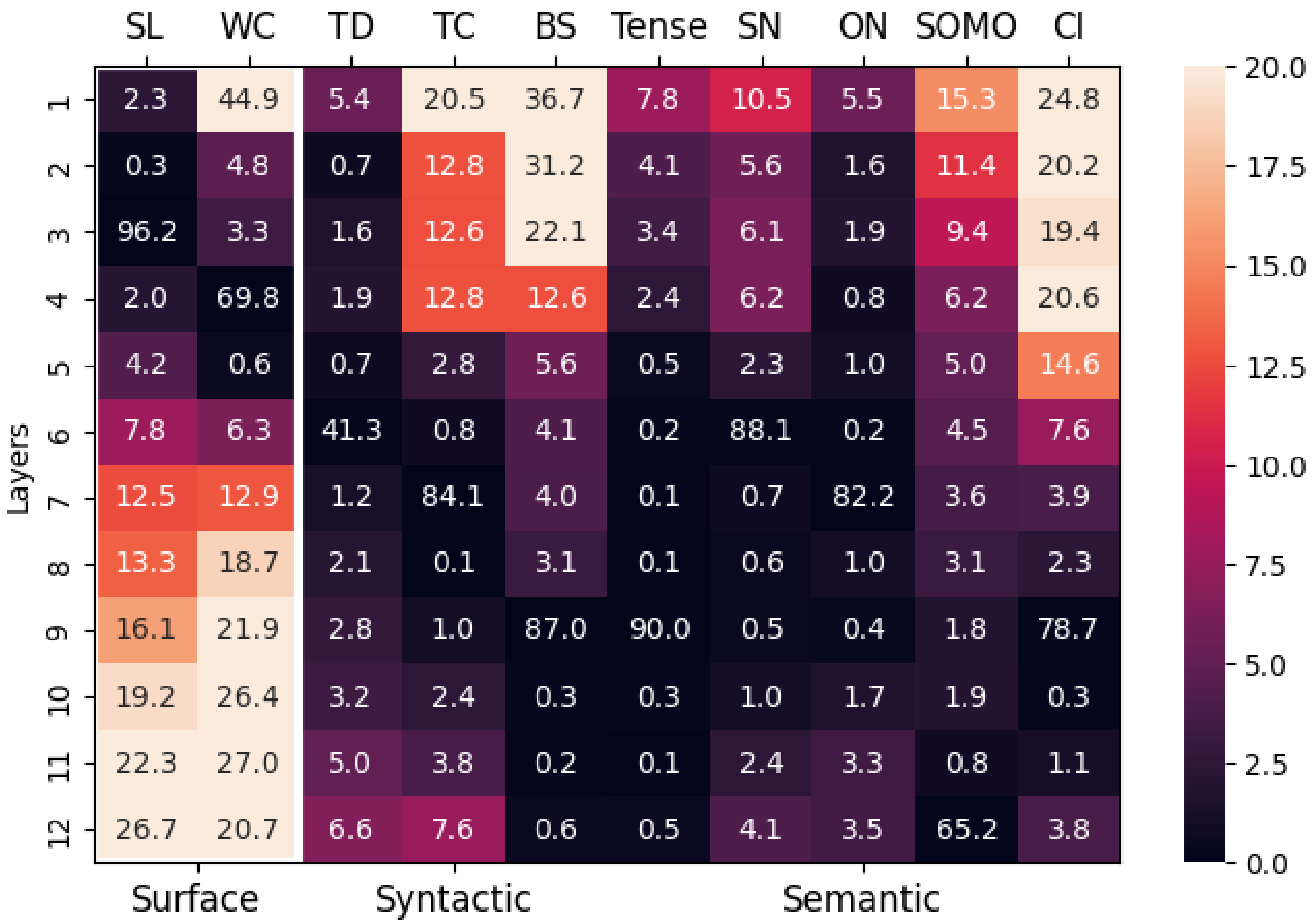
Determines the strength of association between two random variables based upon the number of pairs of paired samples that are "concordant":

Layer

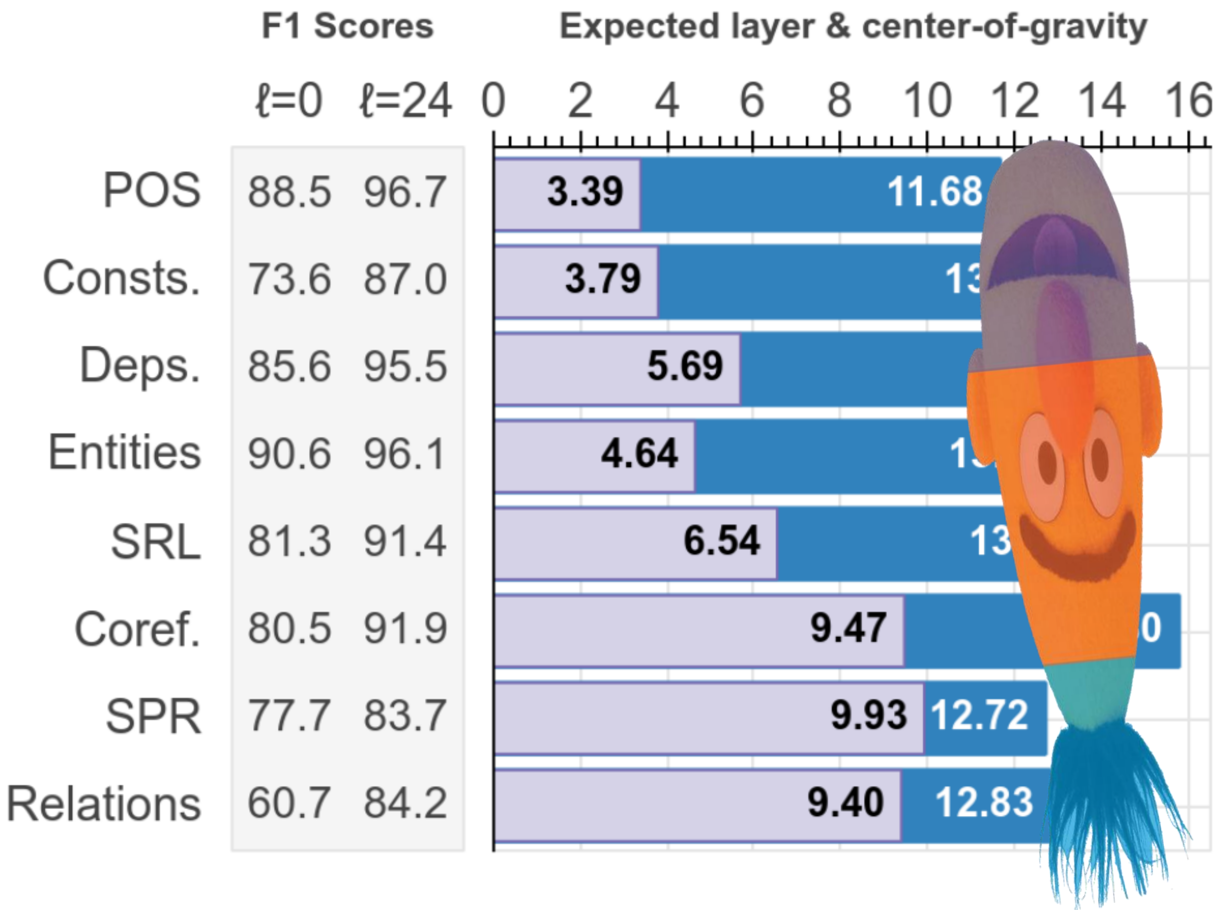
A	1	1
B	2	2
C	3	4
D	4	3
E	5	6
F	6	5
G	7	8
H	8	7
I	9	10
J	10	9
K	11	12
L	12	11

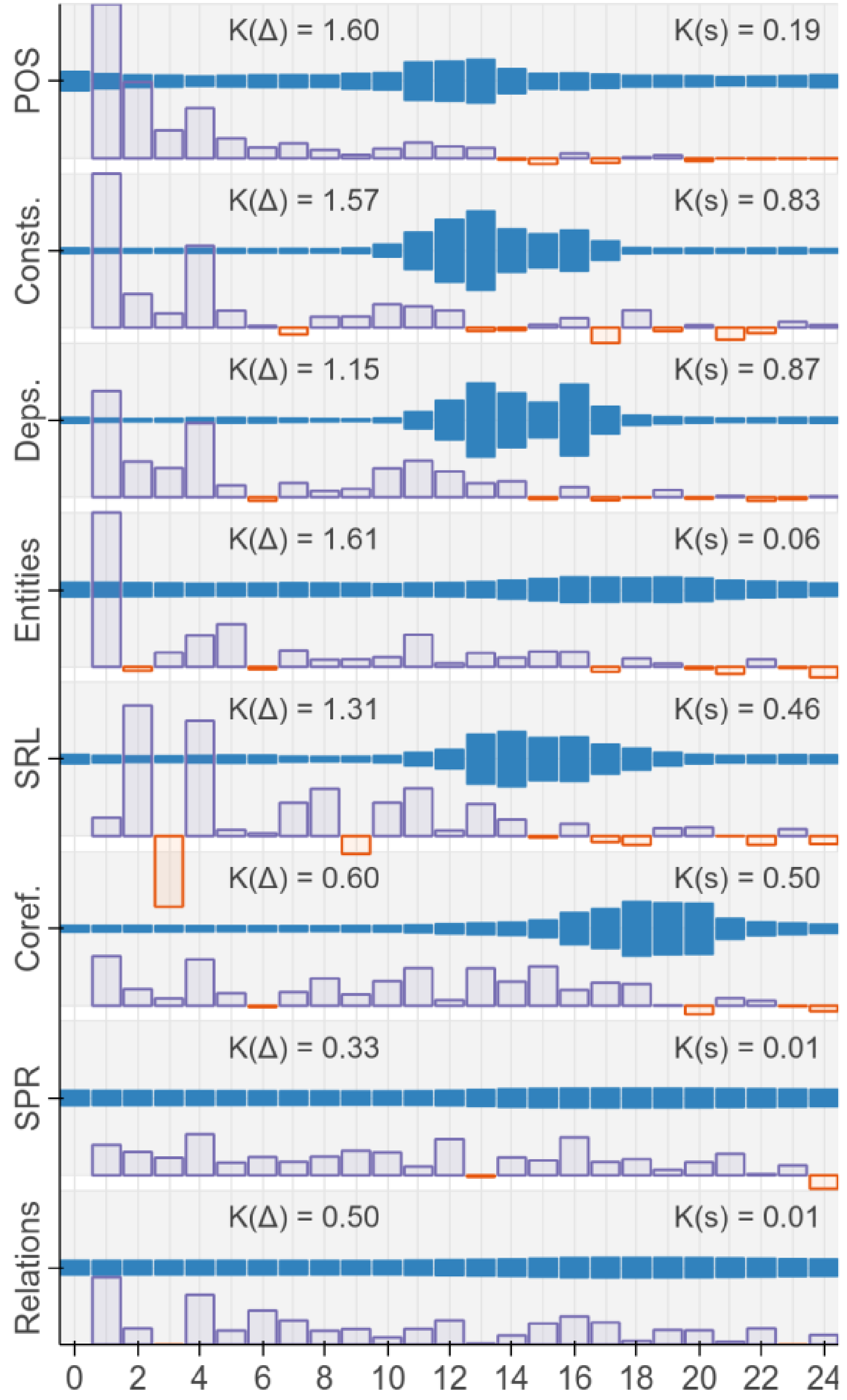
Ordinal ranks

Jawahar et al. (2019) Probing Result



Tenney et al. (2019) Center of Gravity





Pearson r = 0.319, p = 0.44

Weak correlation between layer and COG

Limitation of Tenney et al.'s (2019) Architecture

- Tenney et al. used the **same set of scalar attention weights** for every input sentence: cannot capture **variance of attention patterns across sentences**.
- The probe examines one (or two) span representations: cannot observe task knowledge across **token positions**.

SOLUTION

Attention-based Pooling

(Lee et al., 2017):

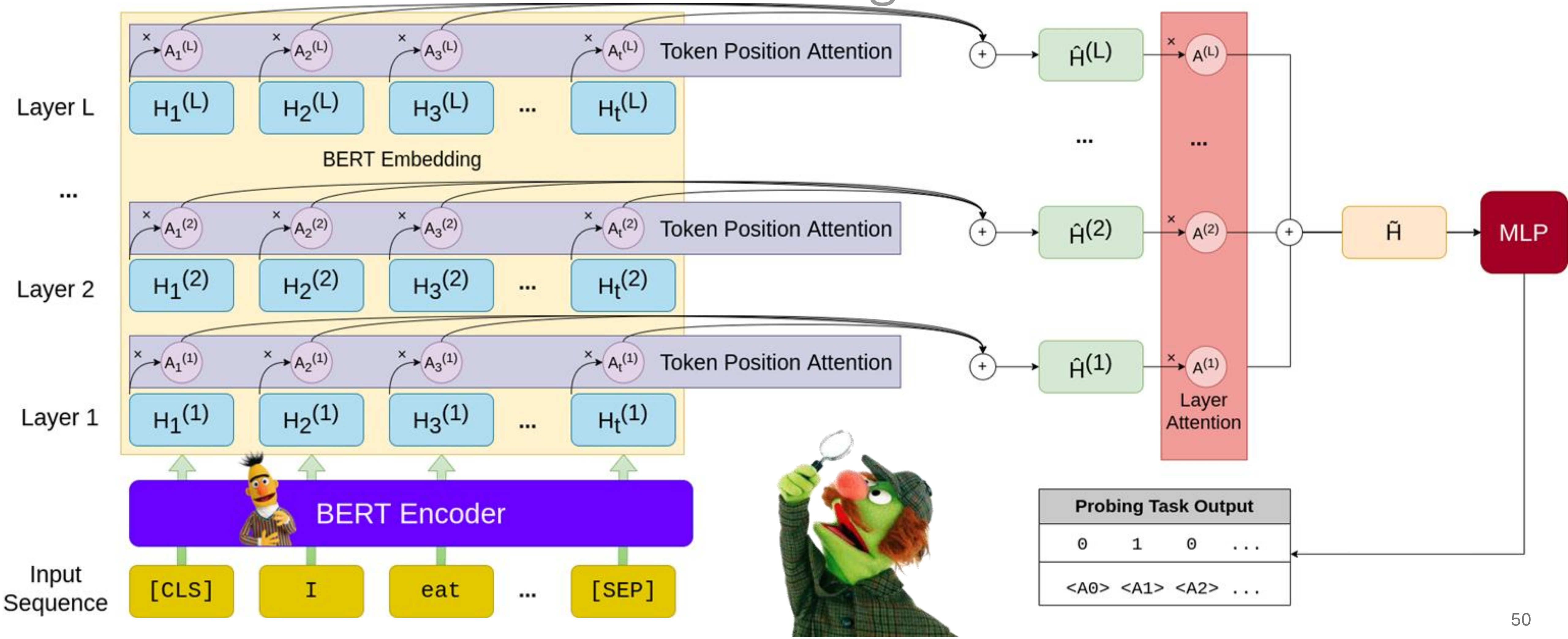
$$\alpha_t = \boldsymbol{w}_\alpha \cdot \boxed{\mathrm{FFNN}_\alpha(\boldsymbol{x}_t^*)}$$

$$a_{i,t} = \frac{\exp(\alpha_t)}{\sum_{k=\mathrm{START}(i)}^{\mathrm{END}(i)} \exp(\alpha_k)}$$

$$\hat{\boldsymbol{x}}_i = \sum_{t=\mathrm{START}(i)}^{\mathrm{END}(i)} a_{i,t} \cdot \boldsymbol{x}_t$$

GridLoc Probe

- Token Position
- Layer
- Randomness & Training

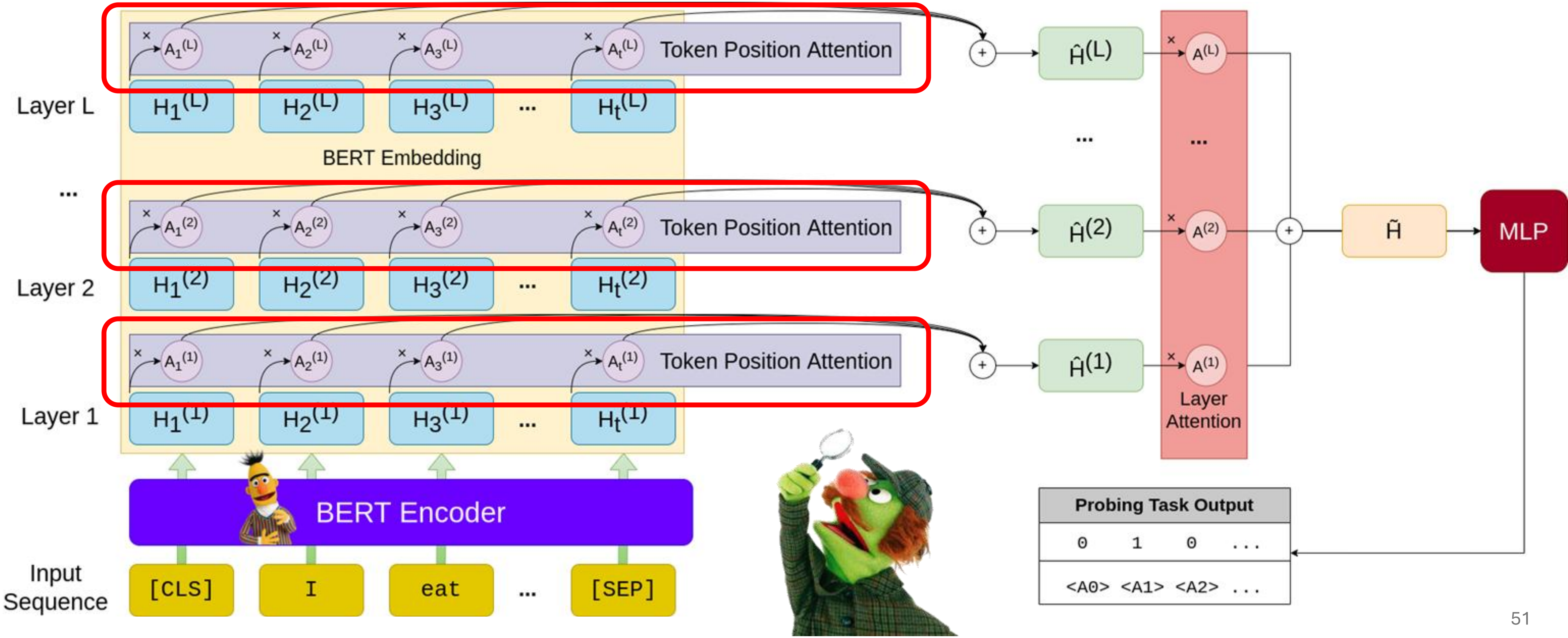


GridLoc Probe

Token position attention:

$$\mathbf{A}^{\text{token},(\ell)} = \text{softmax}(\mathbf{w}_{\text{token}} \cdot \text{RNN}(\mathbf{H}^{(\ell)}))$$

- Token Position
- Layer
- Randomness & Training

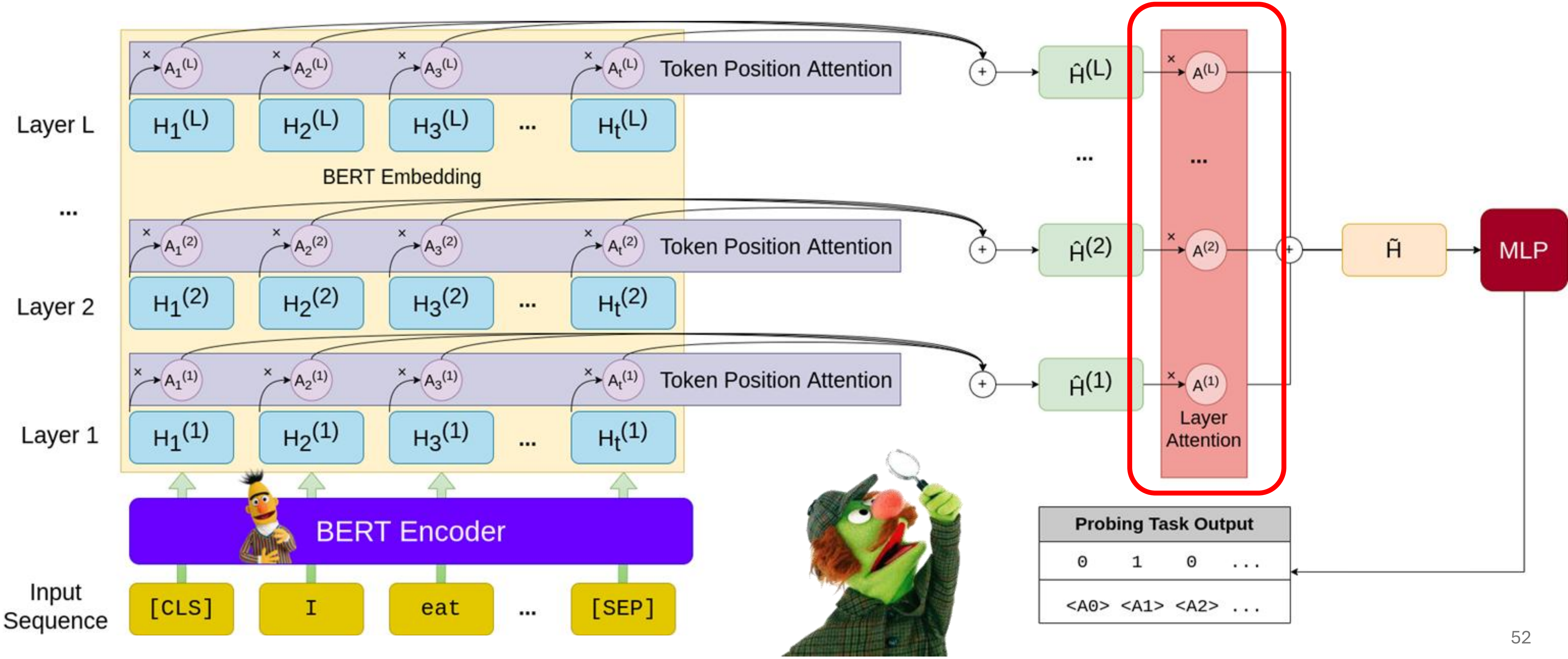


GridLoc Probe

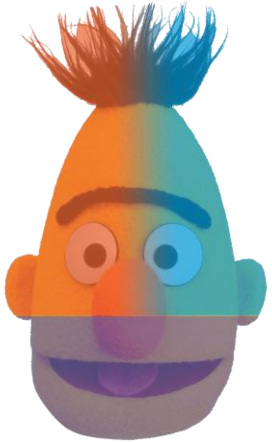
Layer attention:

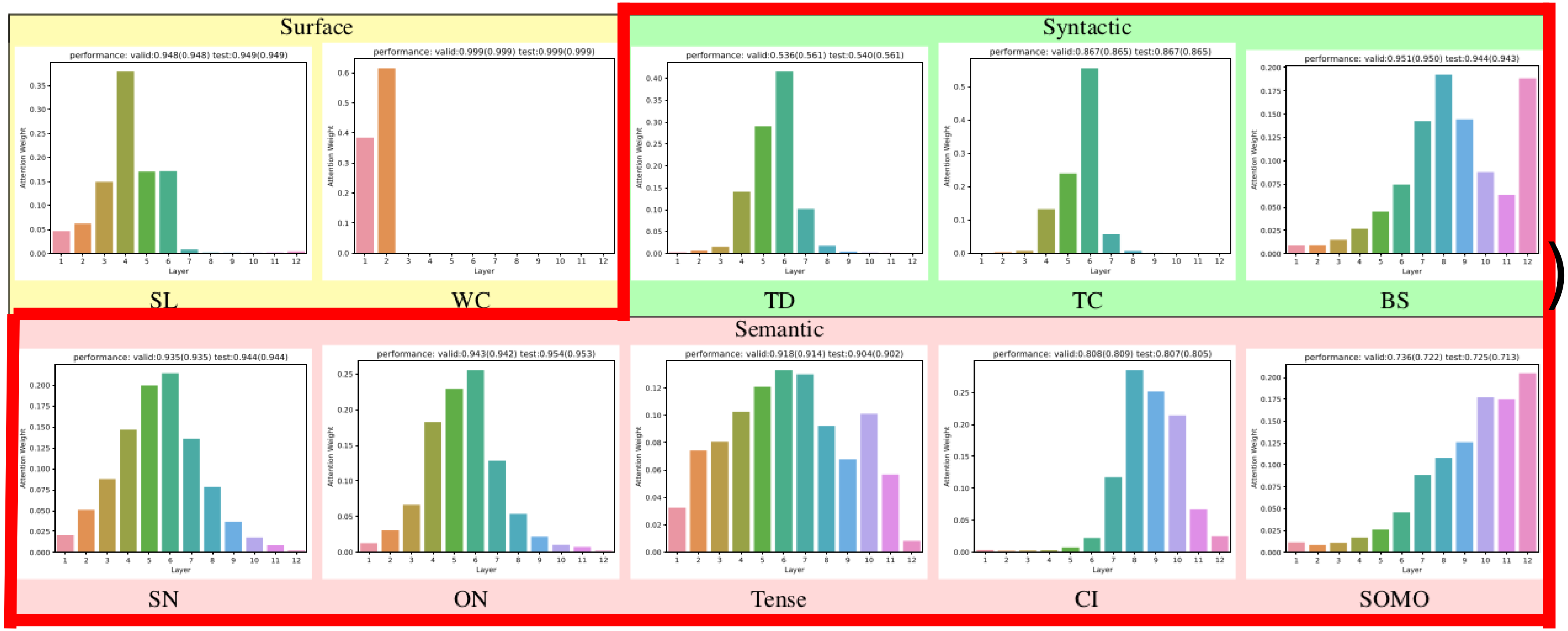
$$\mathbf{A}^{\text{layer}} = \text{softmax}(\mathbf{w}_{\text{layer}} \cdot \hat{\mathbf{H}}^{(\ell)})$$

- Token Position
- **Layer**
- Randomness & Training



Layers Alone do Not Rediscover the CNLP



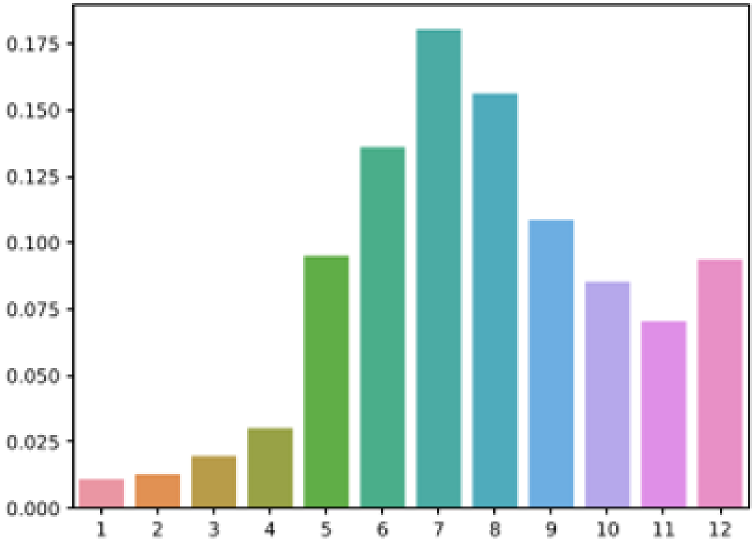
τ () = 0.134

syntactic + semantic

Layer Variance across Sentences

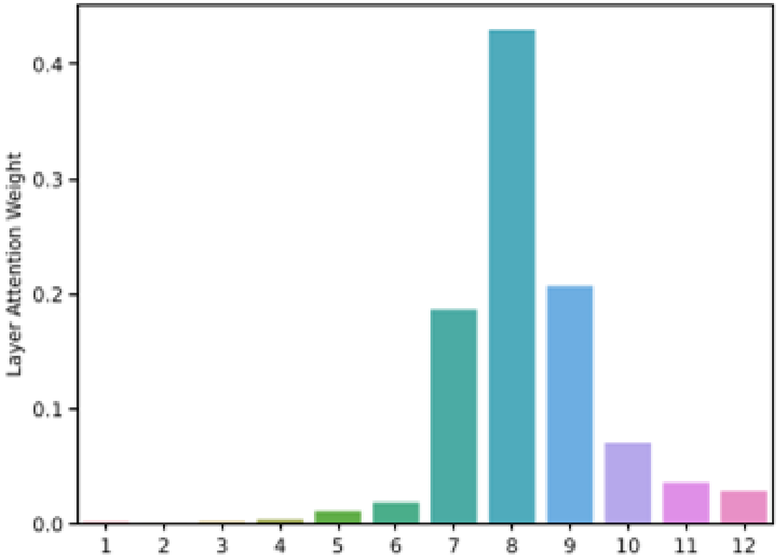
Layer attention weight

Bigram Shift sentence 110000



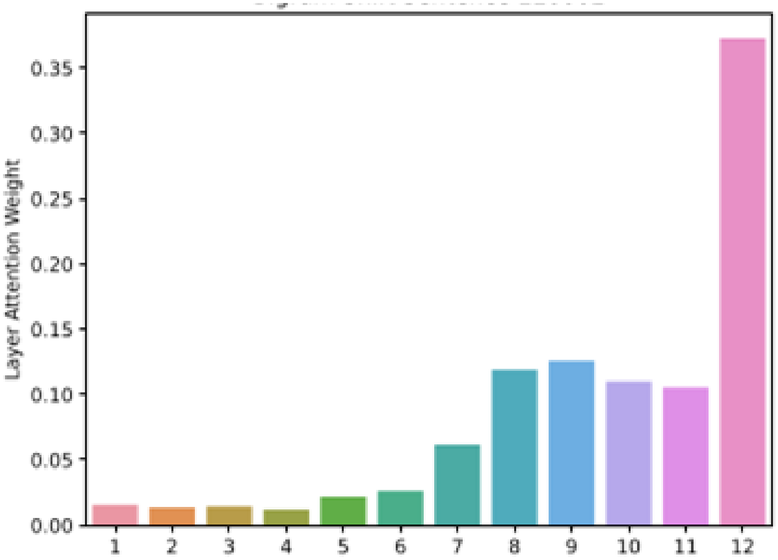
Layer

Bigram Shift sentence 110001



Layer

Bigram Shift sentence 110002



Layer

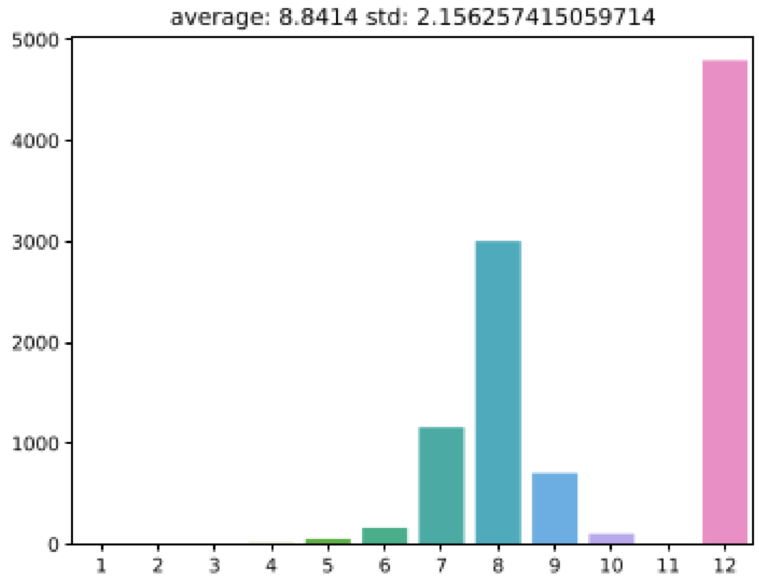
First 3 sentences of the Bigram Shift task test split.

Same GridLoc probe model at the same epoch.

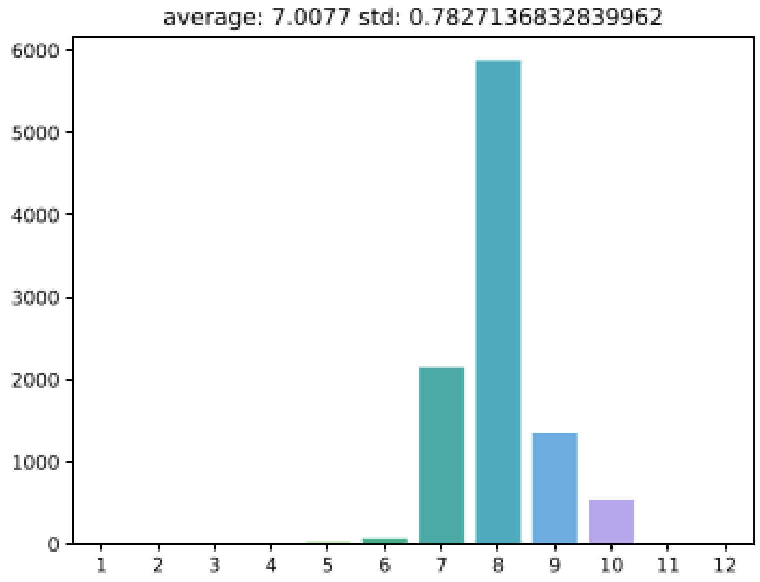
Very different layer attention weights.

Layer Variance across Random Seeds

Probe results are not immune to random initialization effects!



Seed: 0, Best Epoch: 7



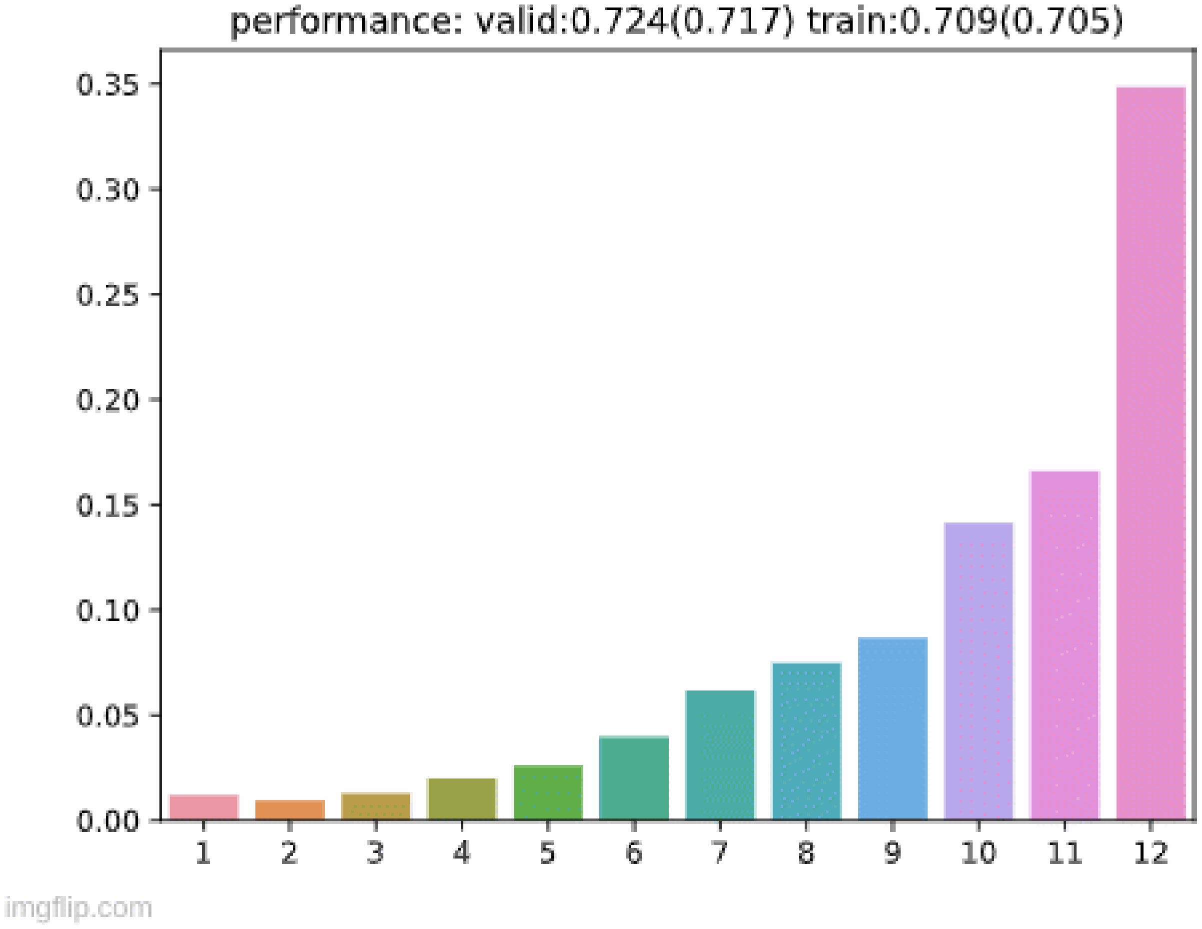
Seed: 1, Best Epoch: 8

Distribution of the best-performing layer over the Bigram Shift test set sentences for two probing runs with different random seeds.

Layer Variance through Training Time

Average layer attention weight distribution change through training iteration.

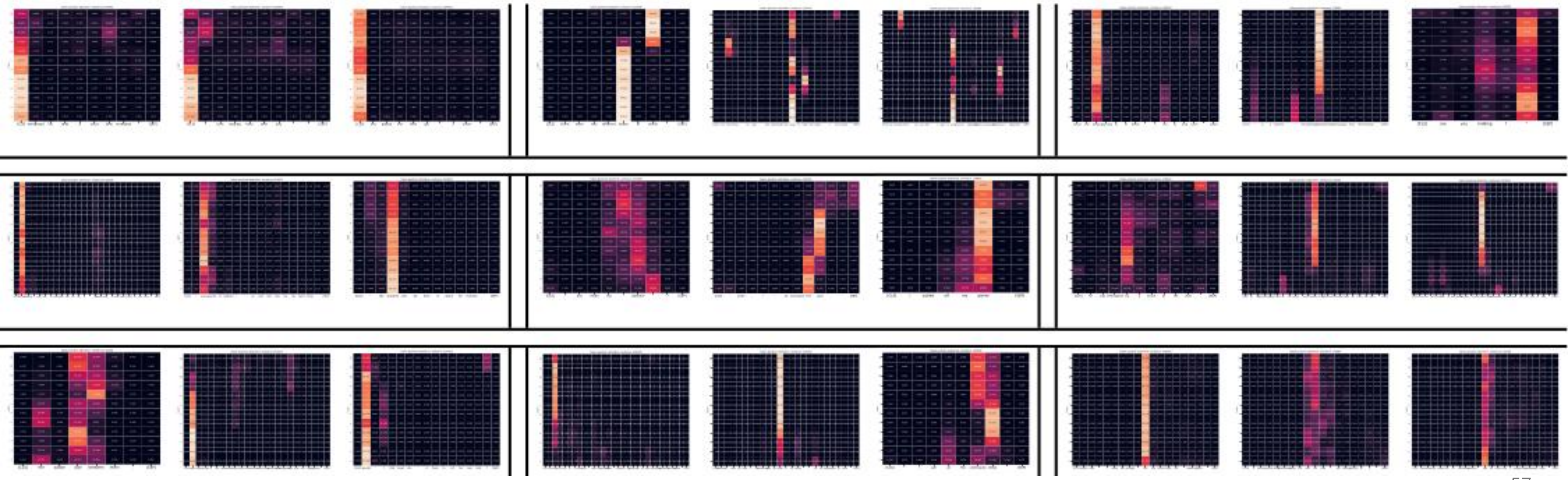
(SOMO, seed:0, best epoch: 3)

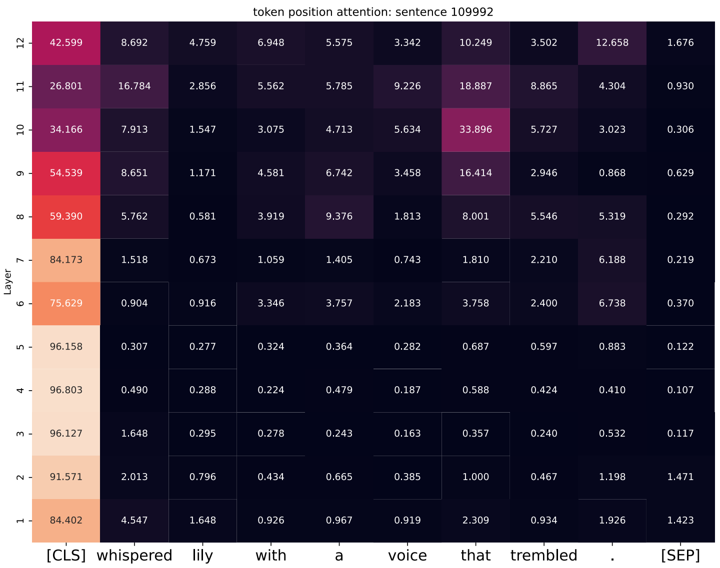


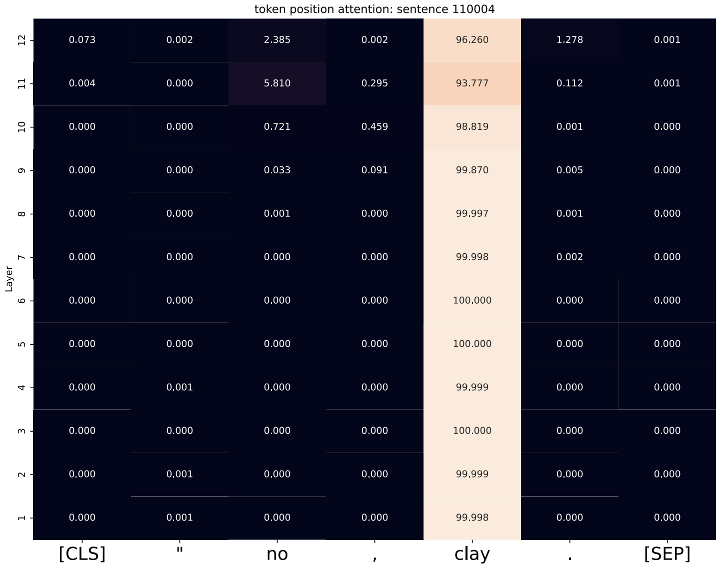
Consistently Idiosyncratic Token Positions

For most sentences, the token position attention at every layer attends to the same token, hence the bright vertical line.

The choice of that token position is not arbitrary — there are linguistic reasons for them.





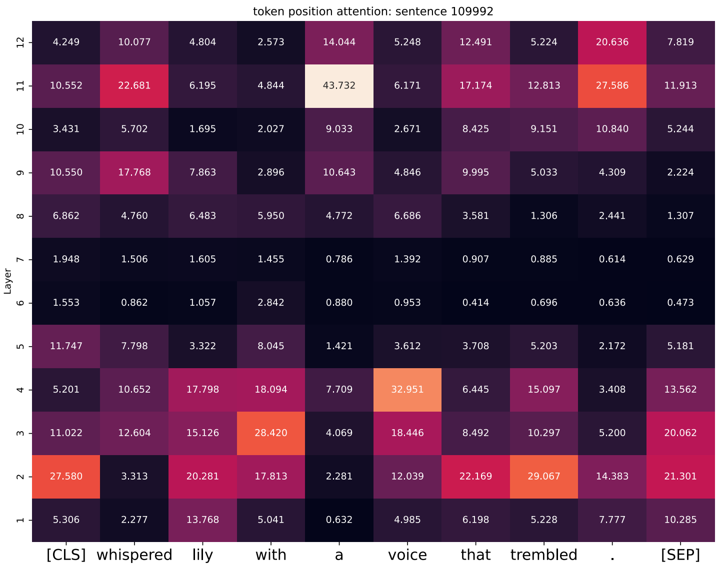


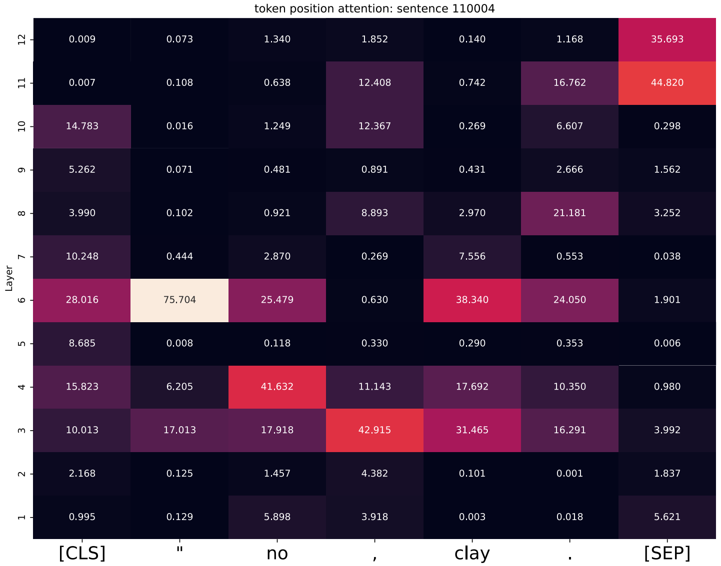
Token Position?

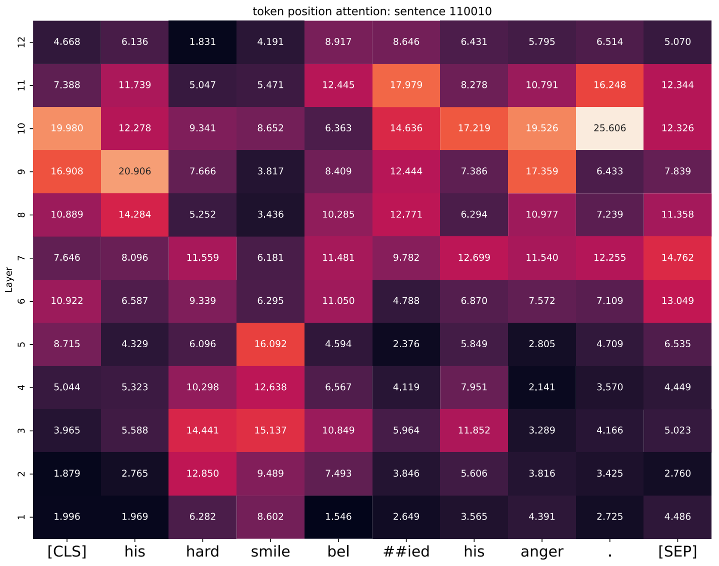
Sentence Length (sent id: 109992)

Word Content (sent id: 110004)

Tense (sent id: 110010)

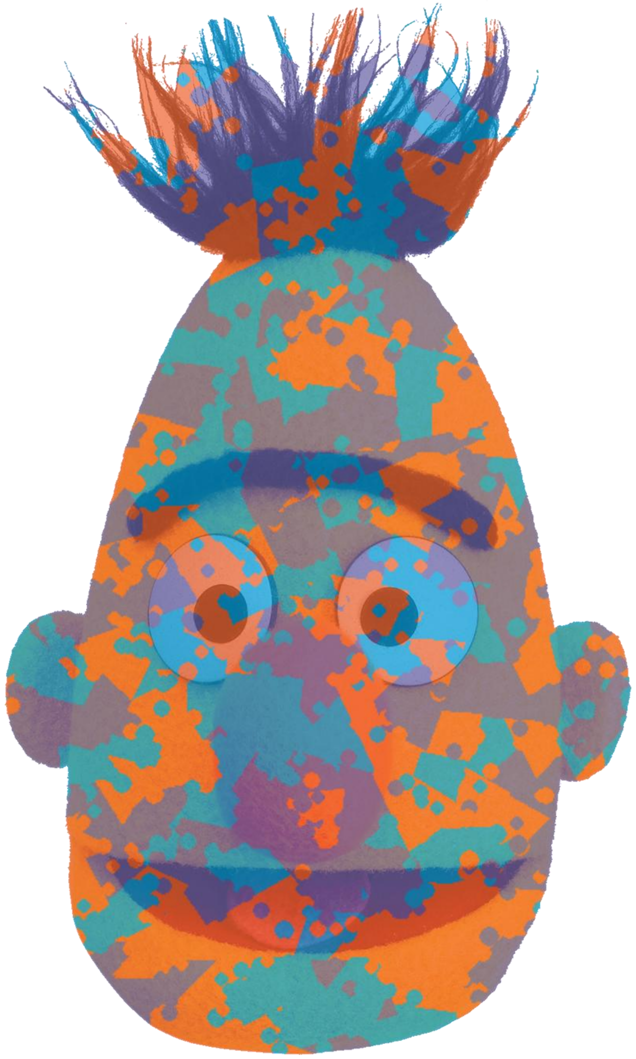






Layer?

Conclusion



- Did BERT rediscover a CNLP? Not in a naïve, architectural sense.
- Probing results regarding BERT layers are unstable; the distribution along token positions is relatively more stable.
- No evidence that pseudo-cognitive appeals to layer depth are to be preferred as the mode of explanation for BERT's inner workings.

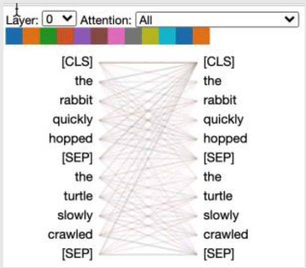
This Segment

“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

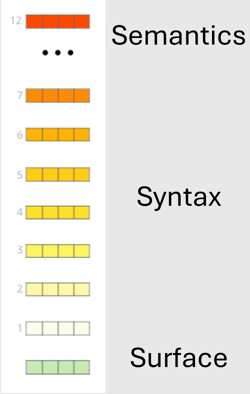
Language acquisition, nature of grammar…

“Understand how transformers word through interpreting attention patterns.”



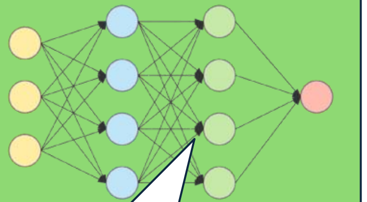
Some kind of syntactic structure inside?

“BERT Rediscovered the Classical NLP Pipeline.”



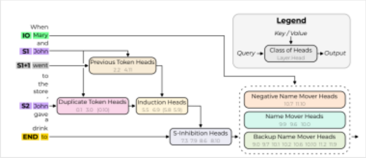
“Knowledge are located within the MLP neurons.”

Transformer MLP weights:



The capital of Canada is Ottawa.

“Information flow is more important!”



1. LM as a whole

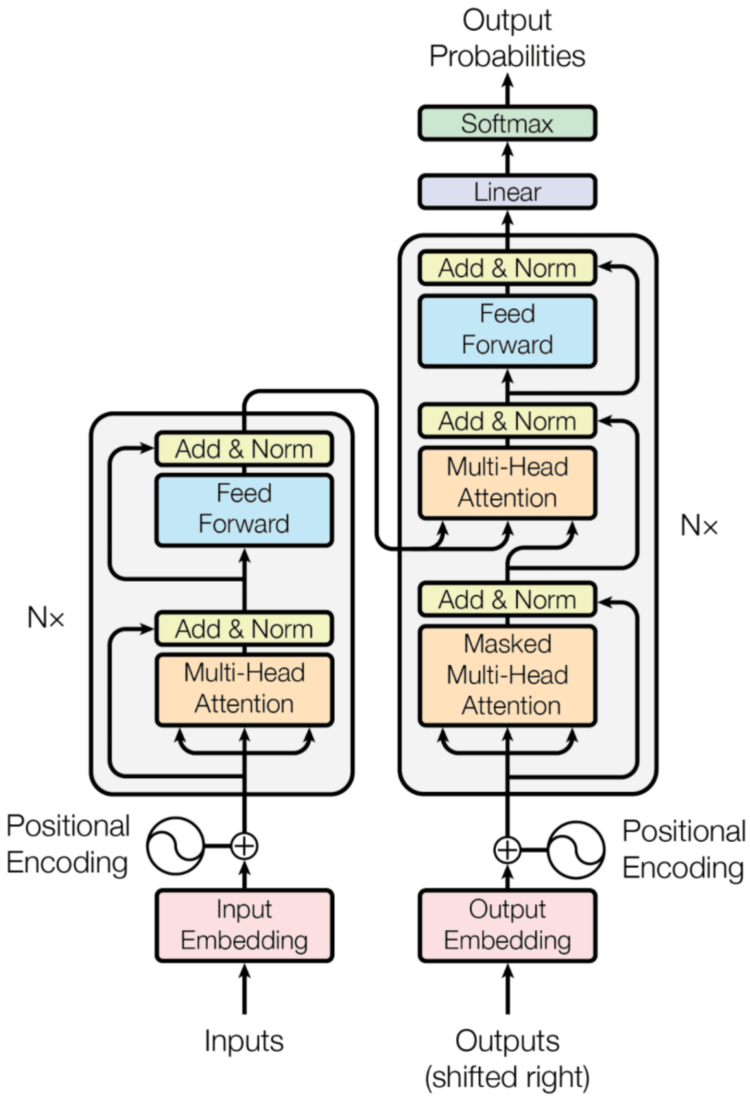
2. Attention Patterns

3. Layer Level

4. Neuron Level

5. Circuit Discovery

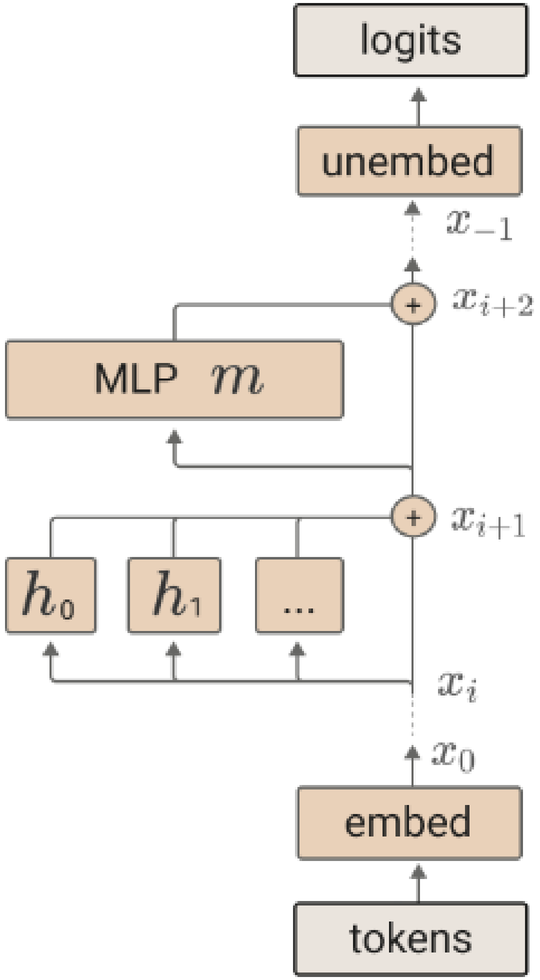
Background: Residual Stream



Background: Residual Stream

Nelson Elhage

Neel Nanda



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

One residual block

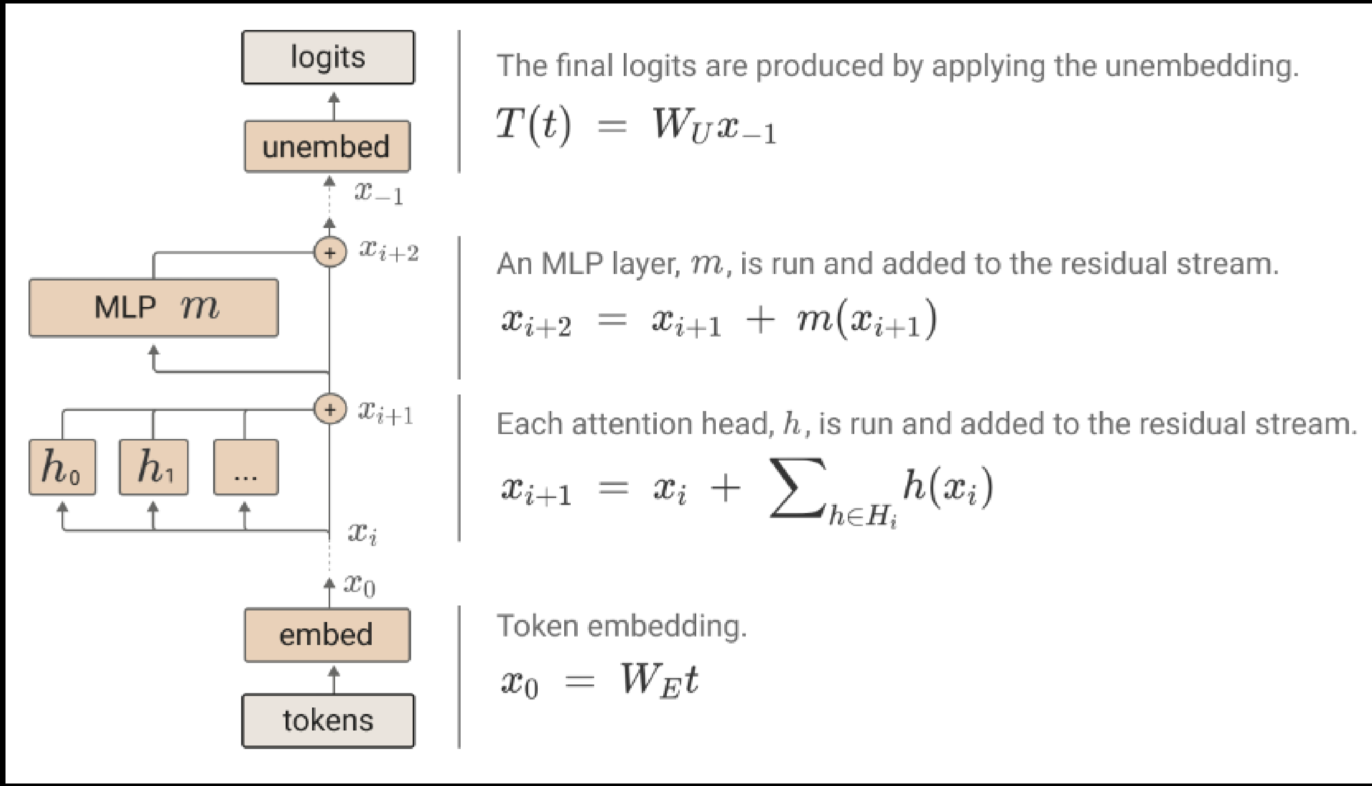
Token embedding.

$$x_0 = W_E t$$

Elhage et al. (2021): A Mathematical Framework for Transformer Circuits. Anthropic.

https://github.com/TransformerLensOrg/TransformerLens

```
class Transformer(nn.Module):

    def forward(self, input):
        residual = self.embed(input) # Embedding layer

        for i, block in self.blocks: # Each block is a layer
            residual = block(residual)

        logits = self.unembed(residual) # [batch, pos, d_vocab]
        return logits

class TransformerBlock(nn.Module):

    def forward(self, resid_pre):

        attn_in = split_attention_head(resid_pre)
        attn_out = self.attn(self.ln1(attn_in))

        resid_mid = resid_pre + attn_out

        mlp_in = resid_mid
        mlp_out = self.mlp(self.ln2(mlp_in))

        resid_post = resid_mid + mlp_out

        return resid_post
```



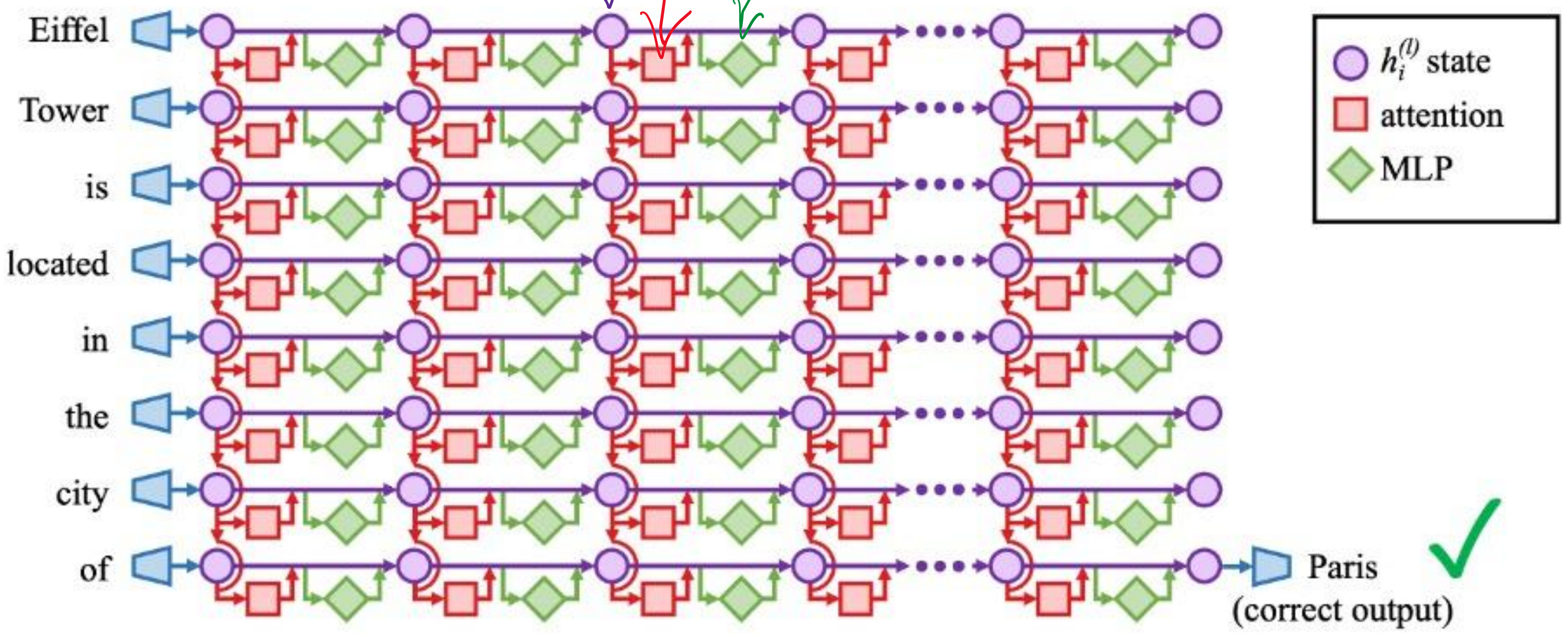
Residual Stream

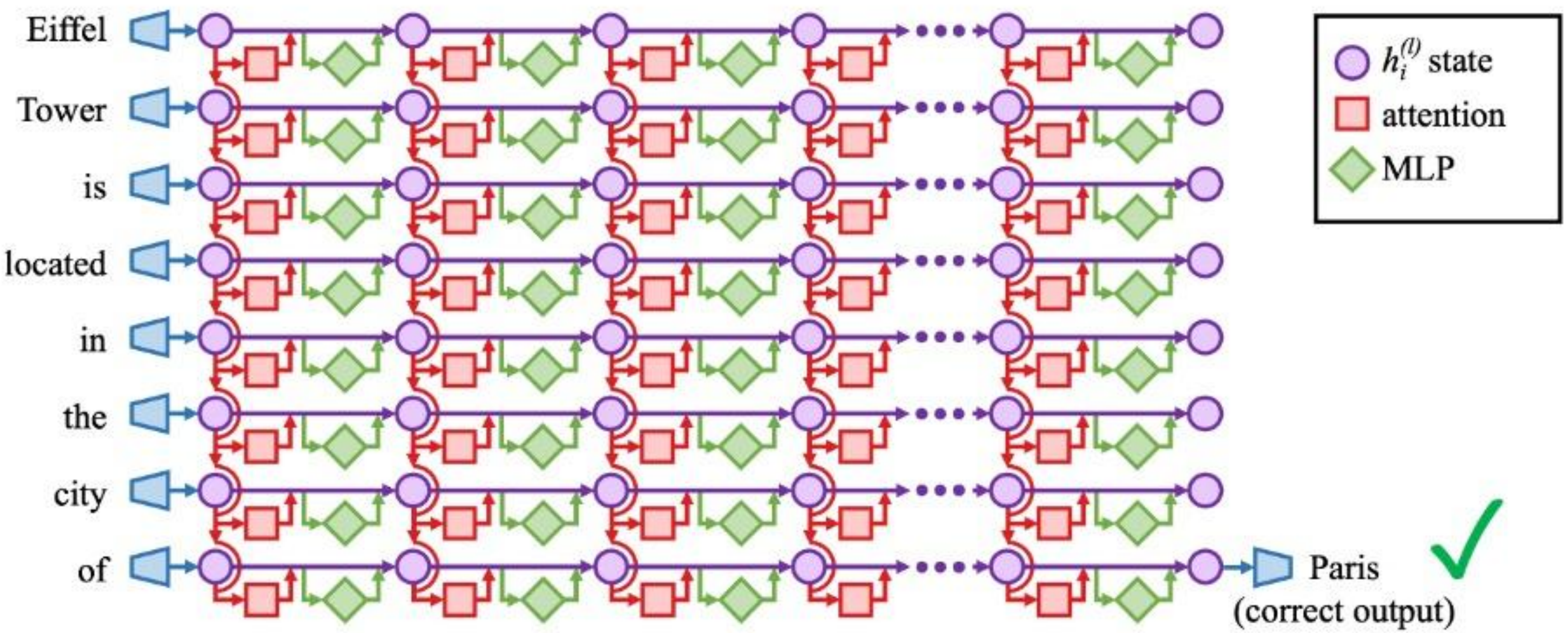
- The transformer block at layer i is calculating:

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i) + \mathrm{MLP}(x_i + \sum_{h \in H_i} h(x_i))$$

- Simplifications:
  - No linear norm (`self.lnX`)
  - Other techniques (split qkv, parallel attn mlp, ...)

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i) + \mathrm{MLP}(x_i + \sum_{h \in H_i} h(x_i))$$

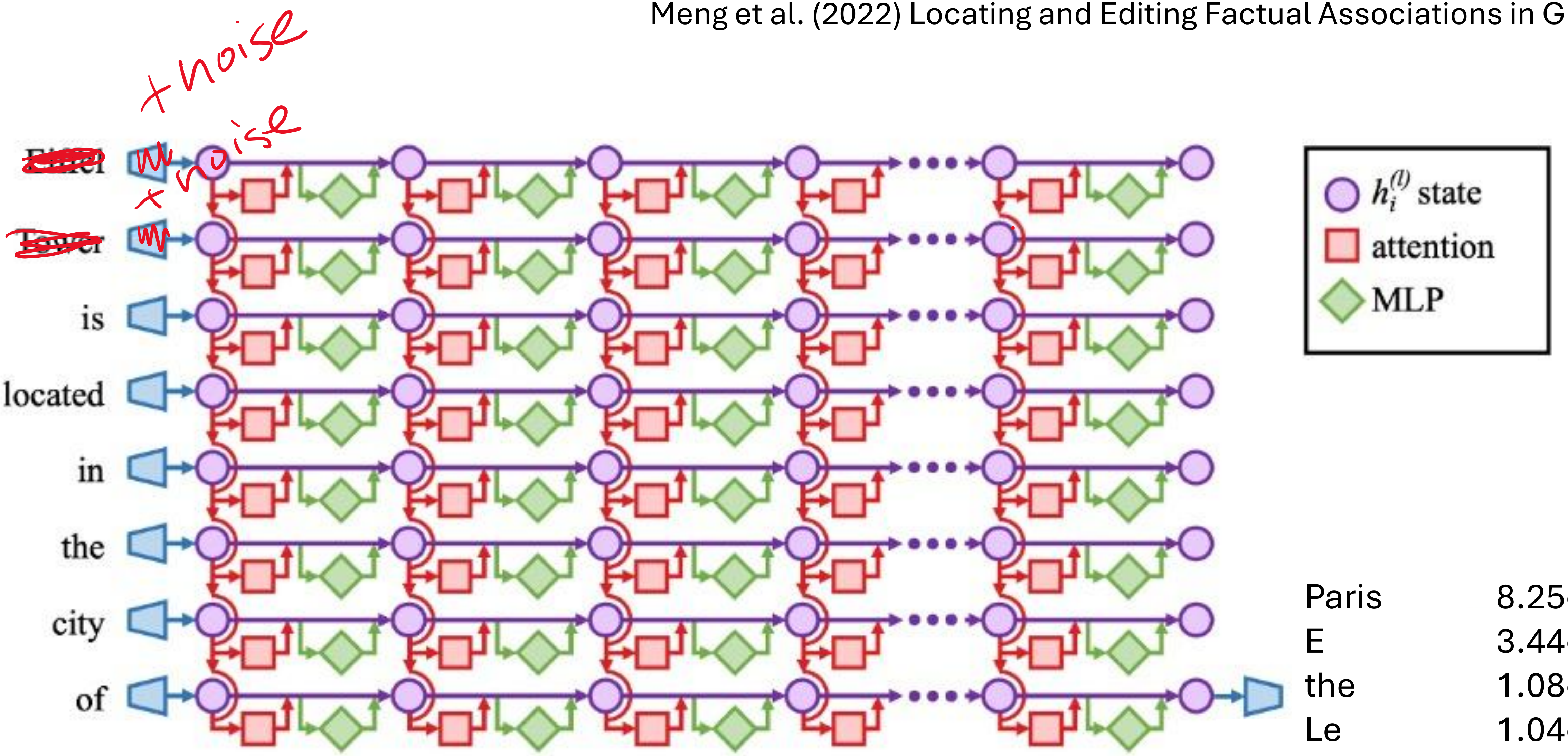




- Prompt: `Eiffel Tower is located in the city of`
- GPT-2 XL: top 5 next tokens:
  - Paris 8.25e-01
  - E 3.44e-02
  - the 1.08e-02
  - Le 1.04e-02
  - Mont 7.91e-03

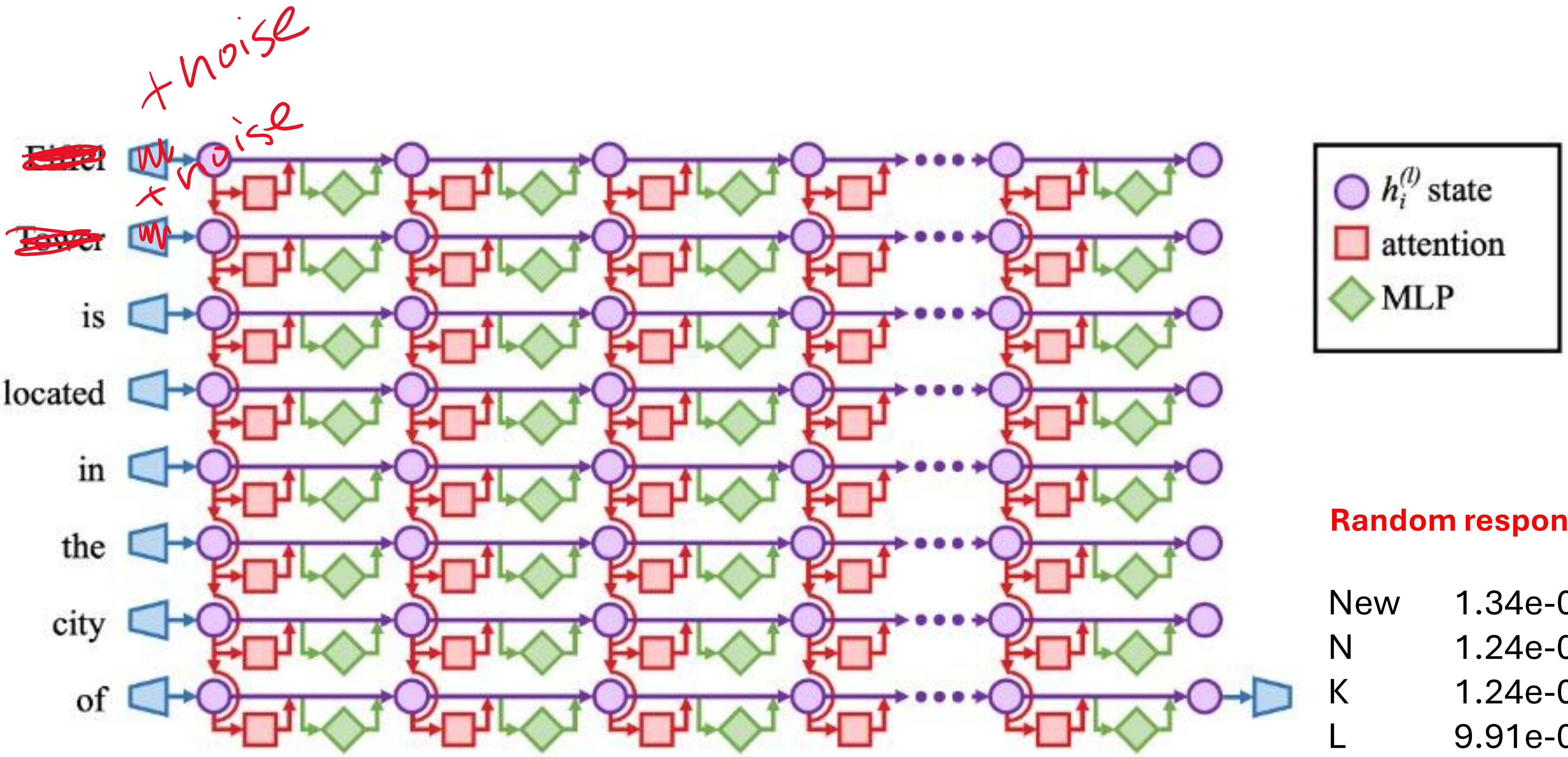
From Quiz 6 (Different Prompt)

Meng et al. (2022) Locating and Editing Factual Associations in GPT.



Paris	8.25e-01
E	3.44e-02
the	1.08e-02
Le	1.04e-02
Mont	7.91e-03

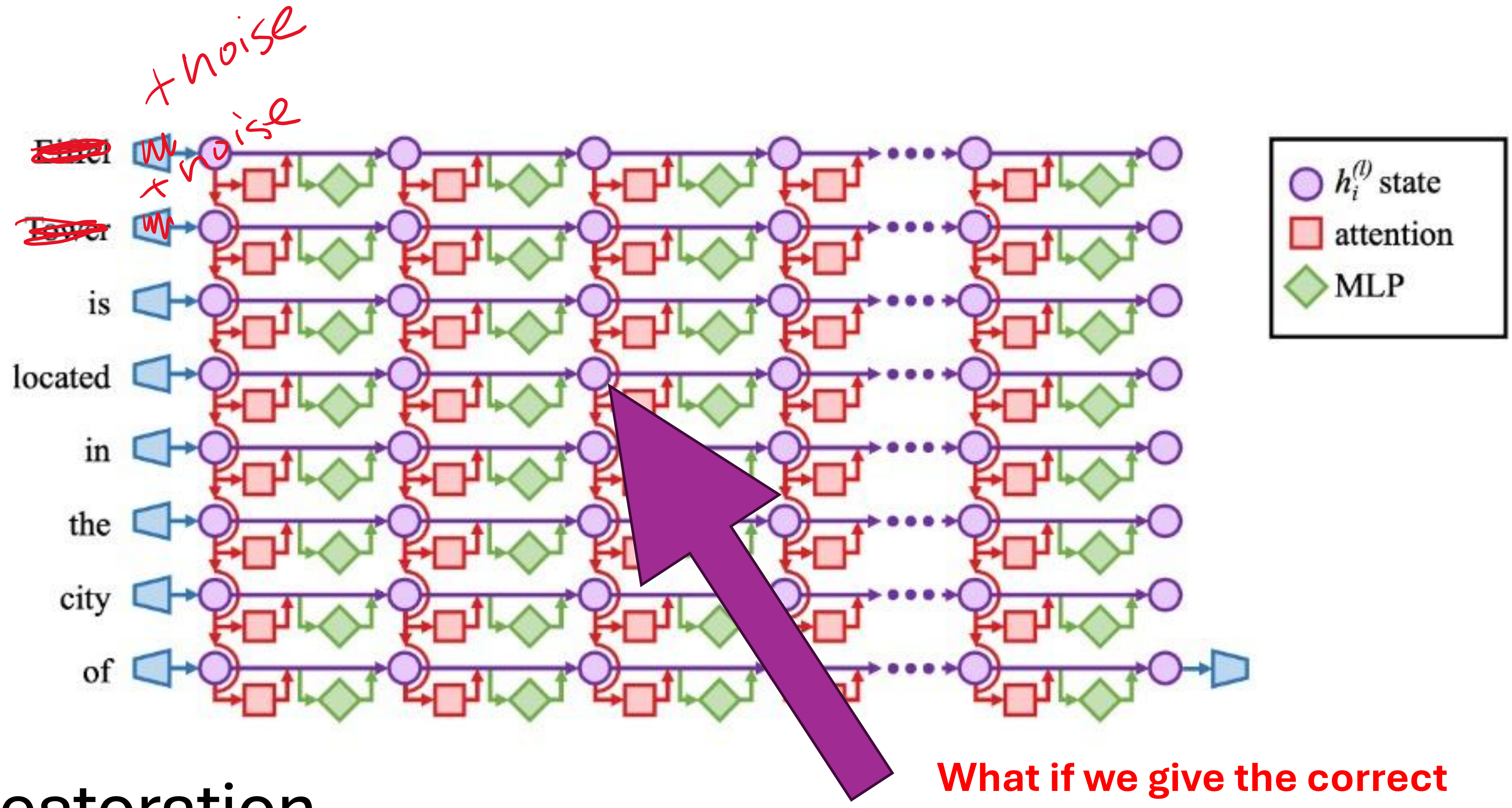
Corrupt Run



Random response!

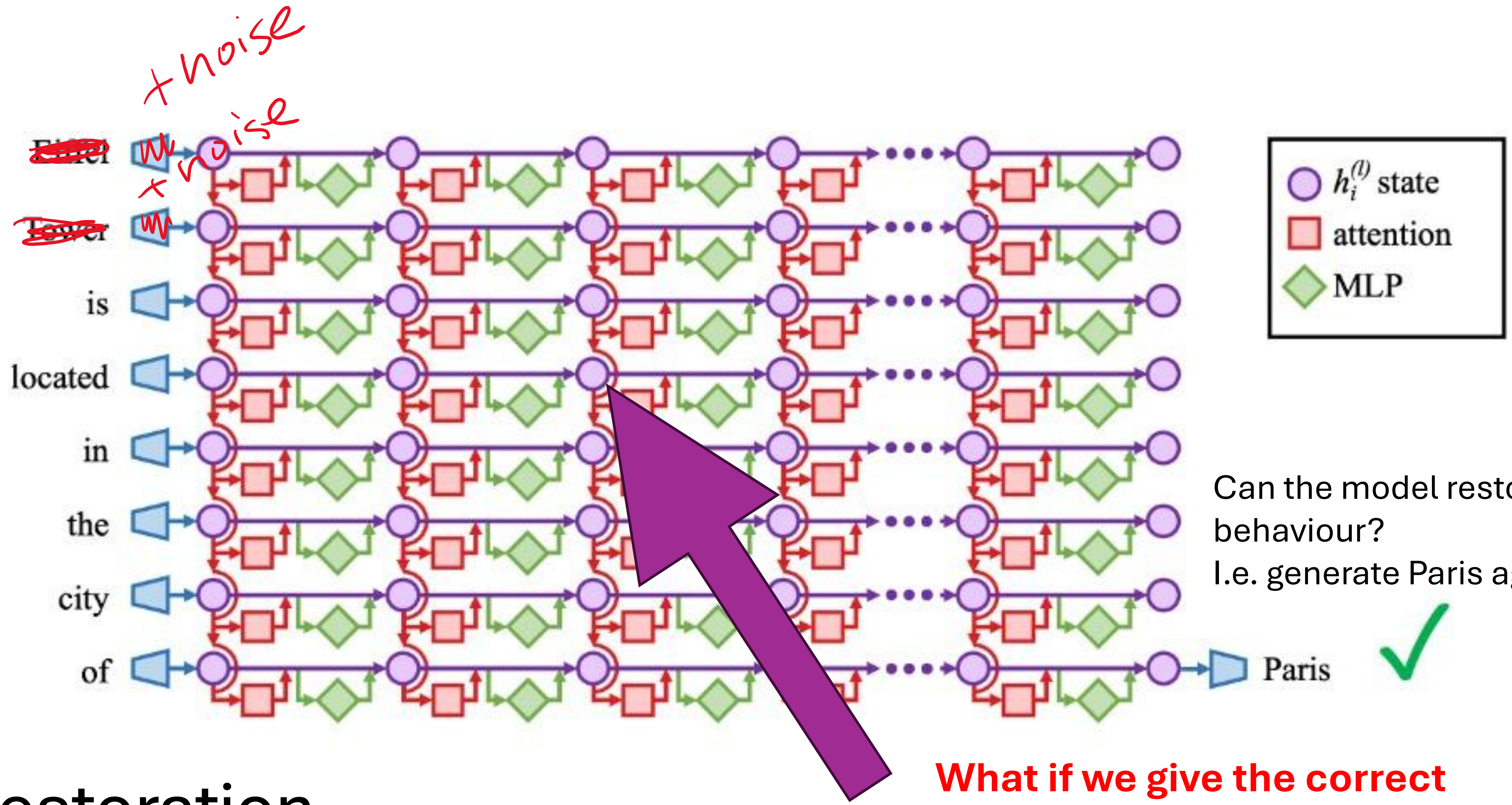
New	1.34e-02
N	1.24e-02
K	1.24e-02
L	9.91e-03
Prague	9.66e-03

Corrupt Run



What if we give the correct hidden state here?

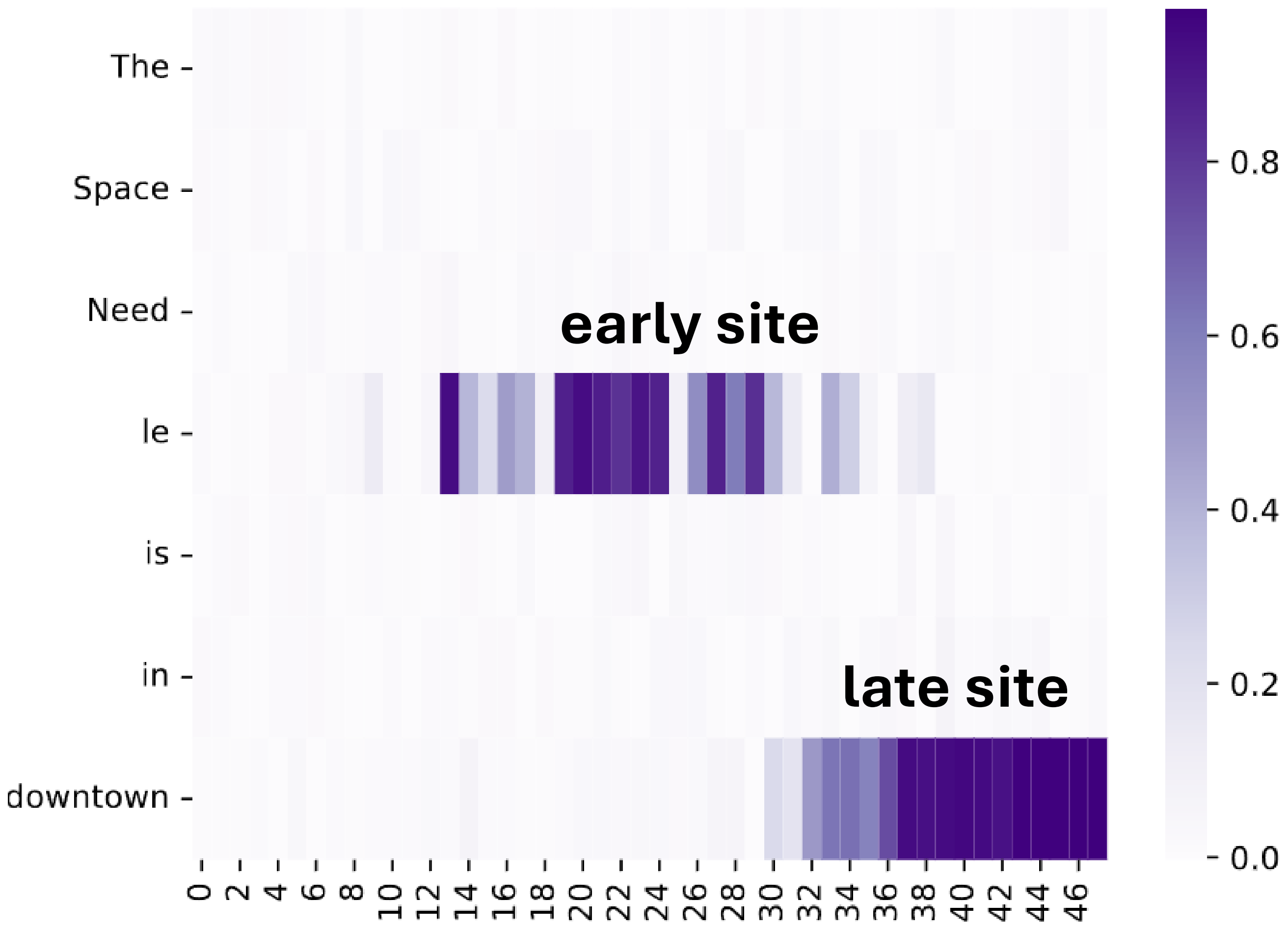
Restoration



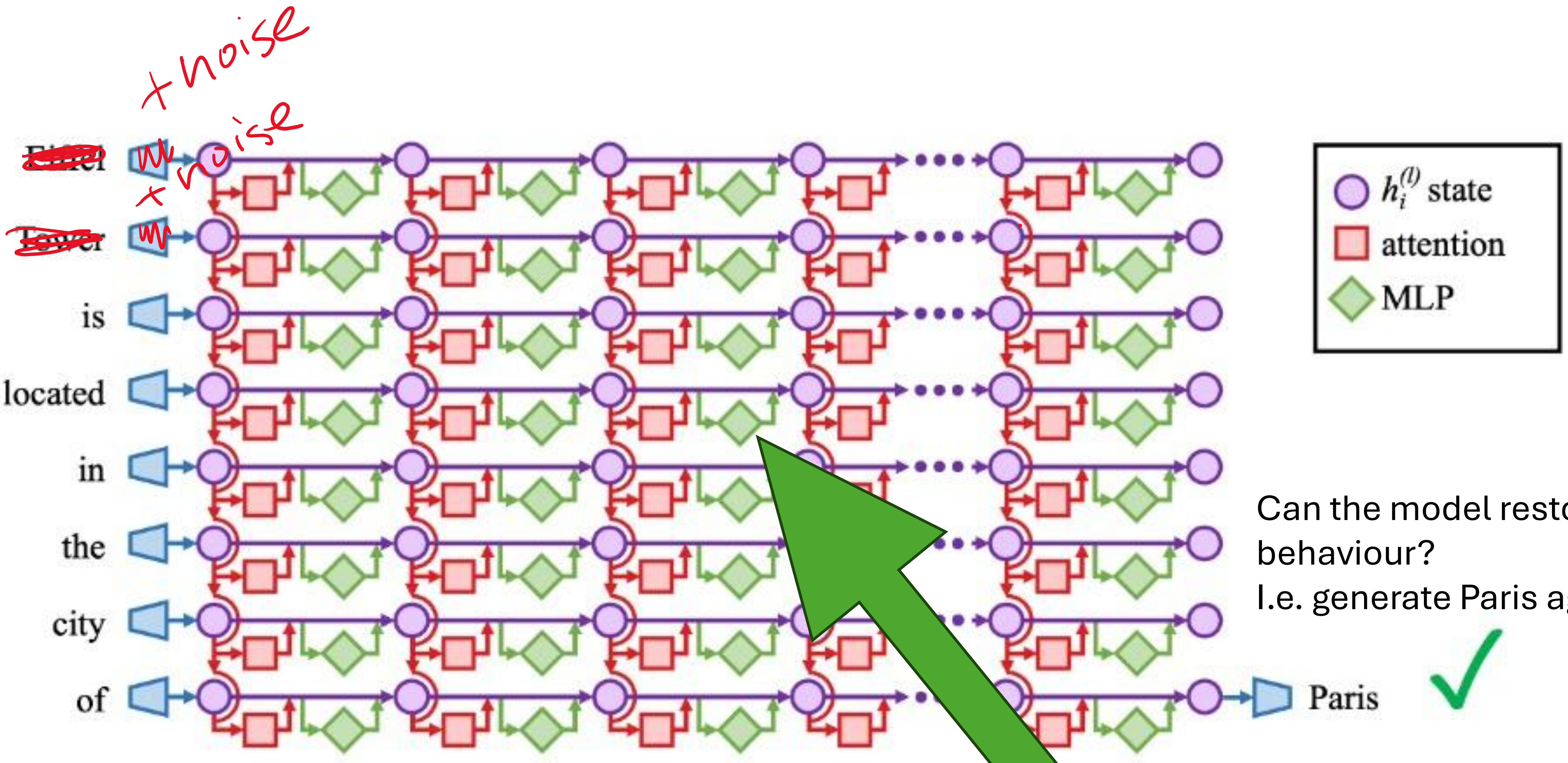
Can the model restore its behaviour?
I.e. generate Paris again?

What if we give the correct hidden state here?

Restoration



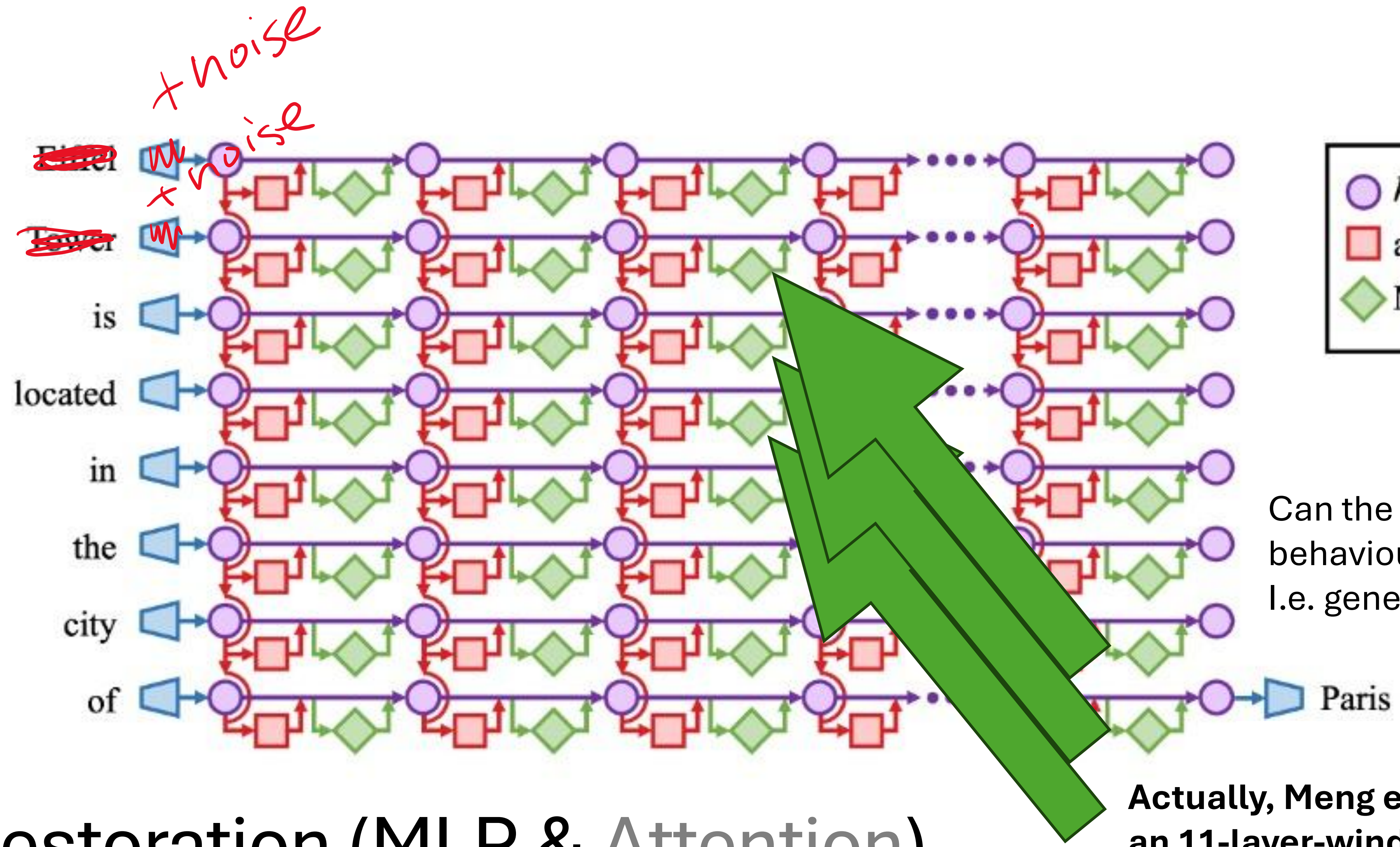
- Corrupt Run: $P^*(\text{Seattle})$
- Restoration Run: $P^{*,c}(\text{Seattle})$
- Indirect Effect (IE): $IE = P^{*,c}(\text{Seattle}) - P^*(\text{Seattle})$

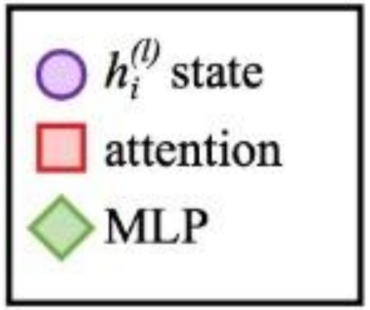


Can the model restore its behaviour?
I.e. generate Paris again?

What about we give the model the correct MLP output here?

Restoration (MLP & Attention)

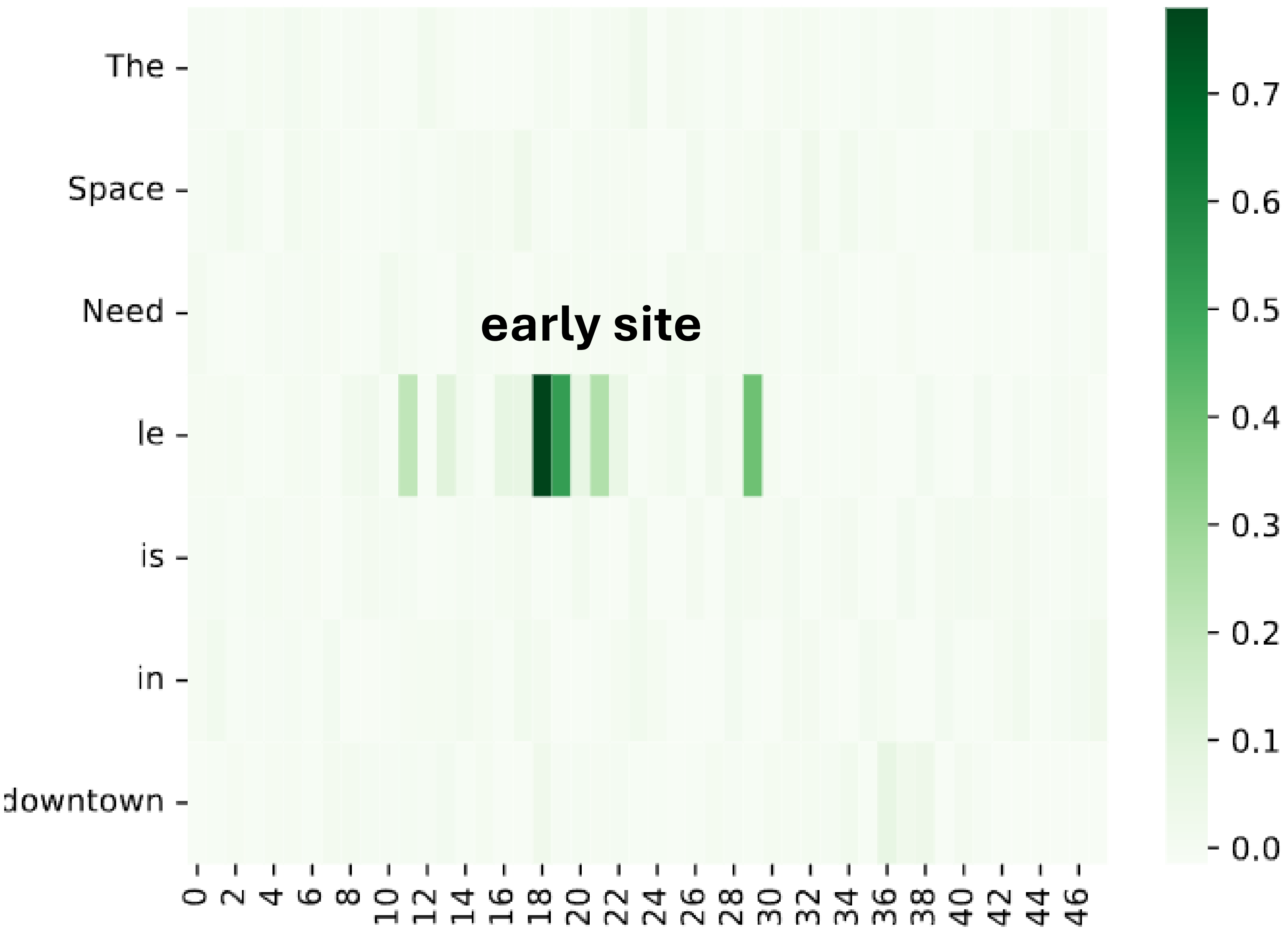


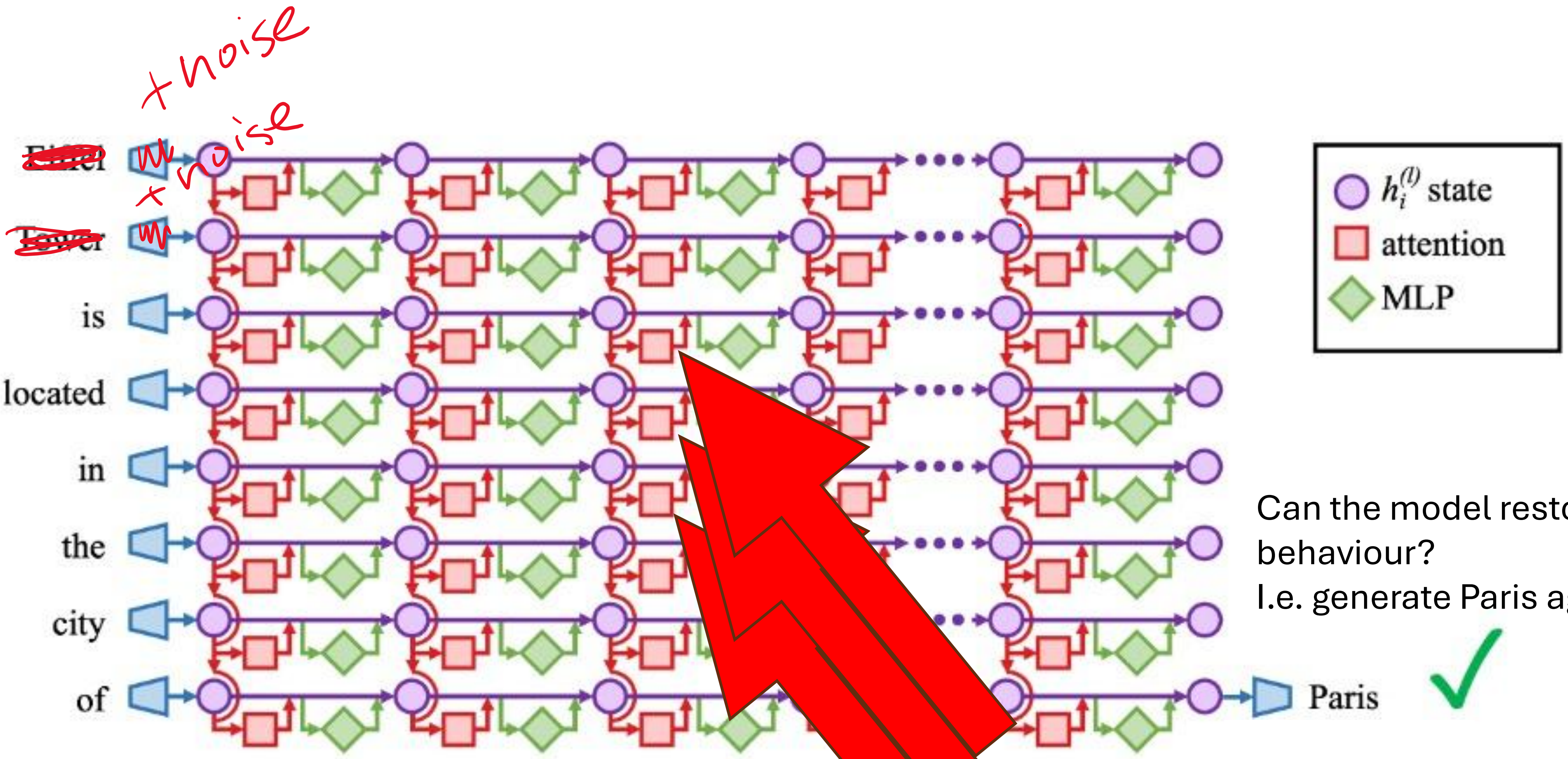


Can the model restore its behaviour?
I.e. generate Paris again?

Actually, Meng et al. restored an 11-layer-window: +5 layers and -5 layers.

Restoration (MLP & Attention)

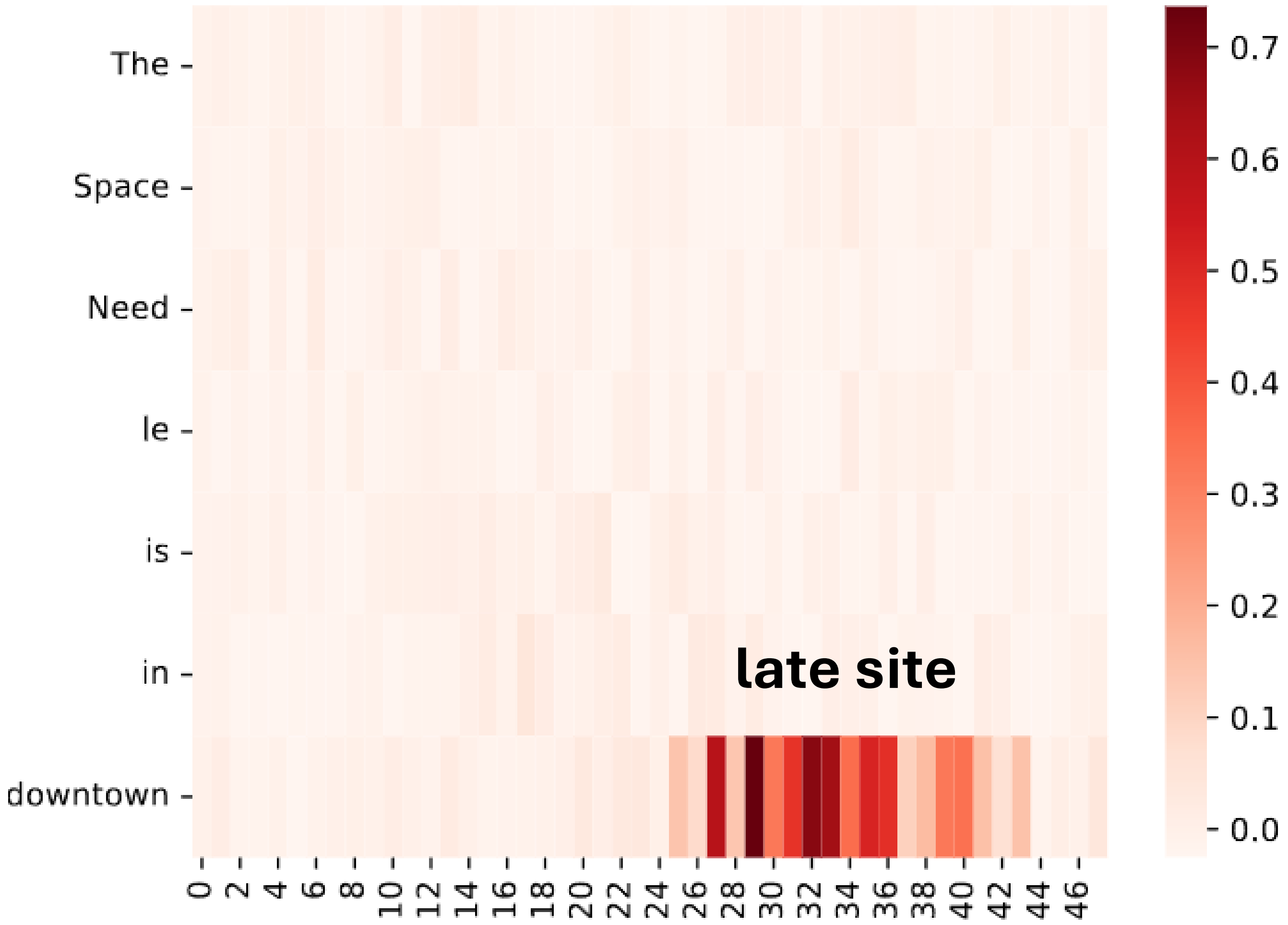

early site
The
Space
Need
le
is
in
downtown
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

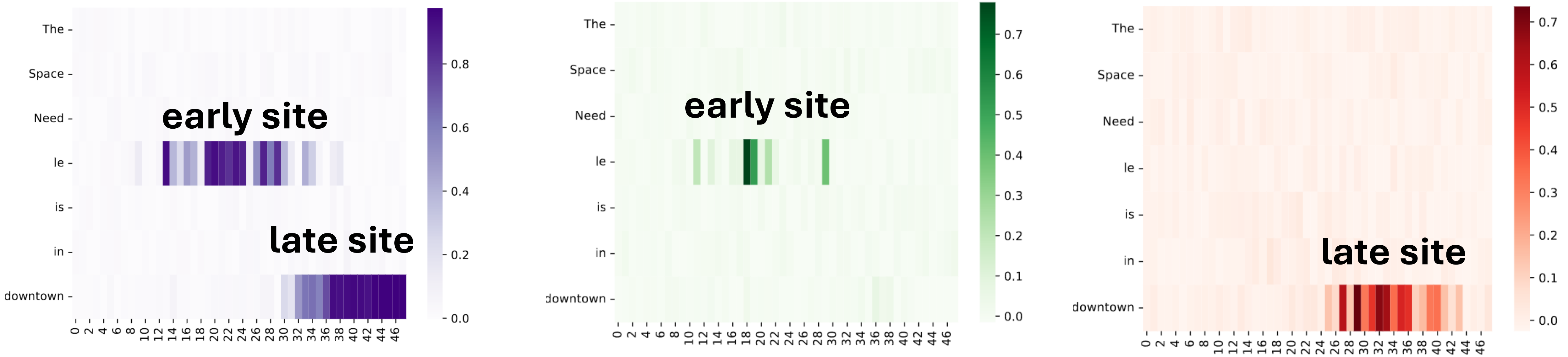


Can the model restore its behaviour?
I.e. generate Paris again?

Restoration (MLP & Attention)

Similar to MLP, we can patch the 11-layer-window of the attention output.


The
Space
Need
le
is
in
downtown
late site
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0



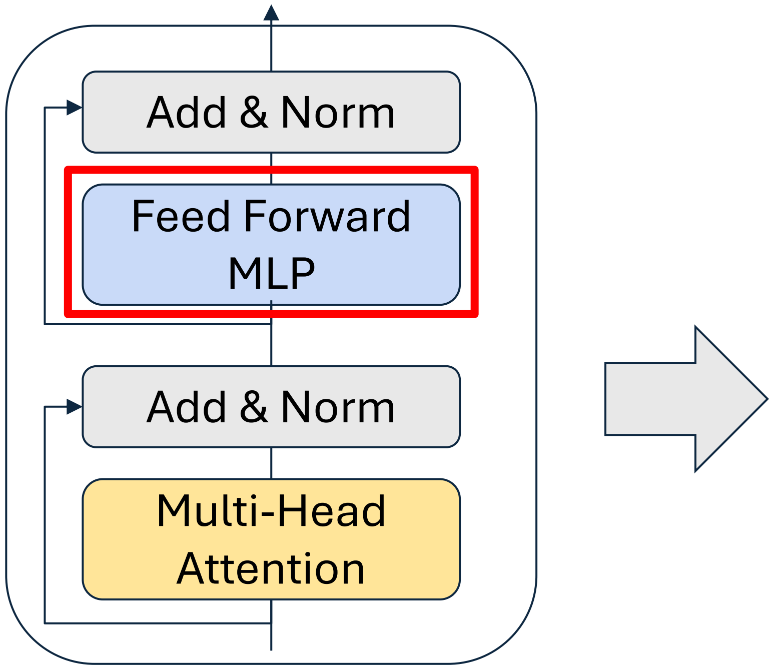
The Localized Factual Association Hypothesis:

- Mid-layer **MLP**:
  - Stores information (memory) in MLP weights.
  - During inference: outputs memorized properties about the input prompt.
- Upper-layer **attention**:
  - Summarizes the outputted properties
  - Gather a final output to the next token position
- A more intricate understanding than “BERT rediscover CNLP.”
  - Limitations? Later

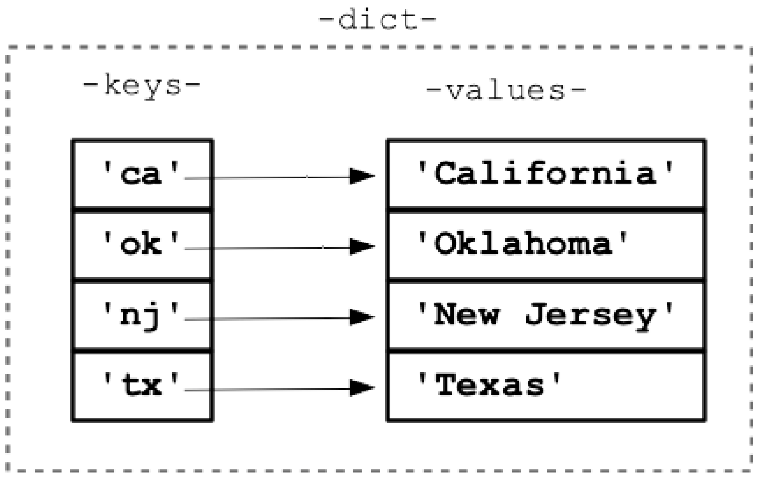
A2 Q3 & Friday’s Tutorial

- A2 Q3:
  - Reproduce Meng et al.’s (2022) causal trace result using transformer_lens
  - Meng et al.'s original code
  - Meng et al.'s Causal Tracing NoteBook
  - (Reading Meng et al.’s code isn’t cheating)
  - TransformerLens
- Friday’s Tutorial
  - TransformerLens crash course
  - An example: task vector

The Knowledge Neuron Thesis

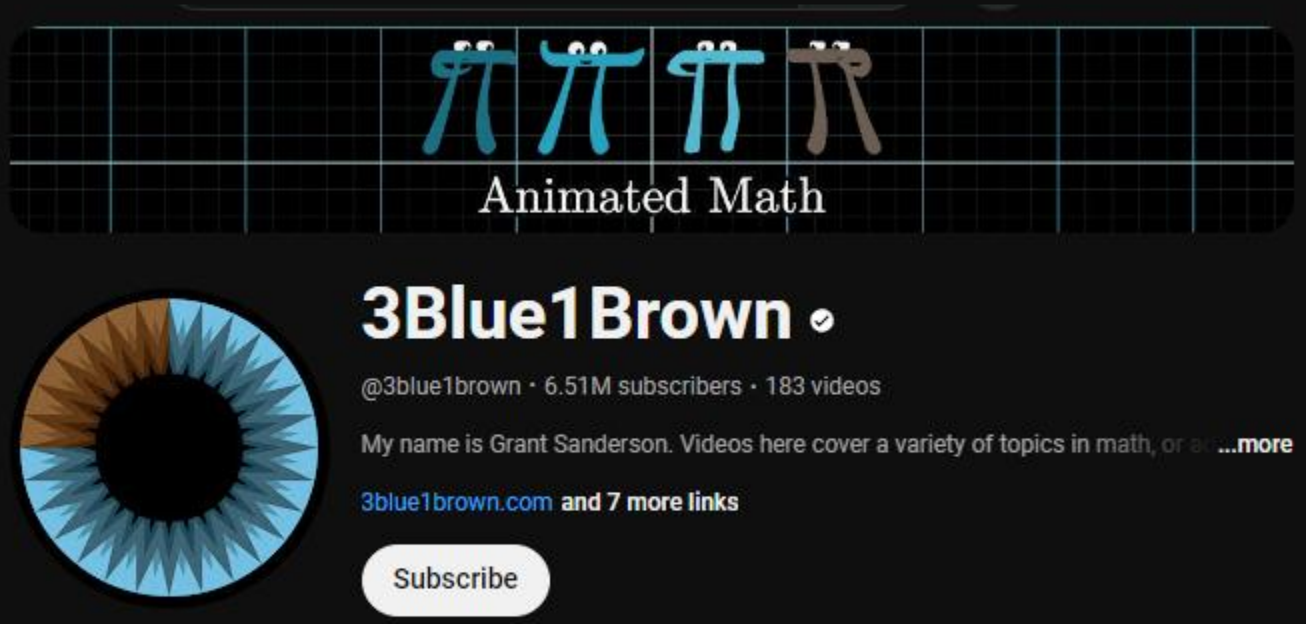


Transformer



Key-value memory

The Knowledge Neuron Thesis



The Knowledge Neuron Thesis

Geva et al.'s (2021) original claim:

- Keys: textual patterns. Values: output vocabulary distribution.

Key	Pattern	Example trigger prefixes
$\mathbf{k}^{1}_{449}$	Ends with "*substitutes*" (shallow)	*At the meeting, Elton said that "for artistic reasons there could be no substitutes* *In German service, they were used as substitutes* *Two weeks later, he came off the substitutes*
$\mathbf{k}^{6}_{2546}$	Military, ends with "*base*"/"*bases*" (shallow + semantic)	*On 1 April the SRSG authorised the SADF to leave their bases* *Aircraft from all four carriers attacked the Australian base* *Bombers flying missions to Rabaul and other Japanese bases*
$\mathbf{k}^{10}_{2997}$	a "part of" relation (semantic)	*In June 2012 she was named as one of the team that competed* *He was also a part of the Indian delegation* *Toy Story is also among the top ten in the BFI list of the 50 films you should*

The Knowledge Neuron Thesis

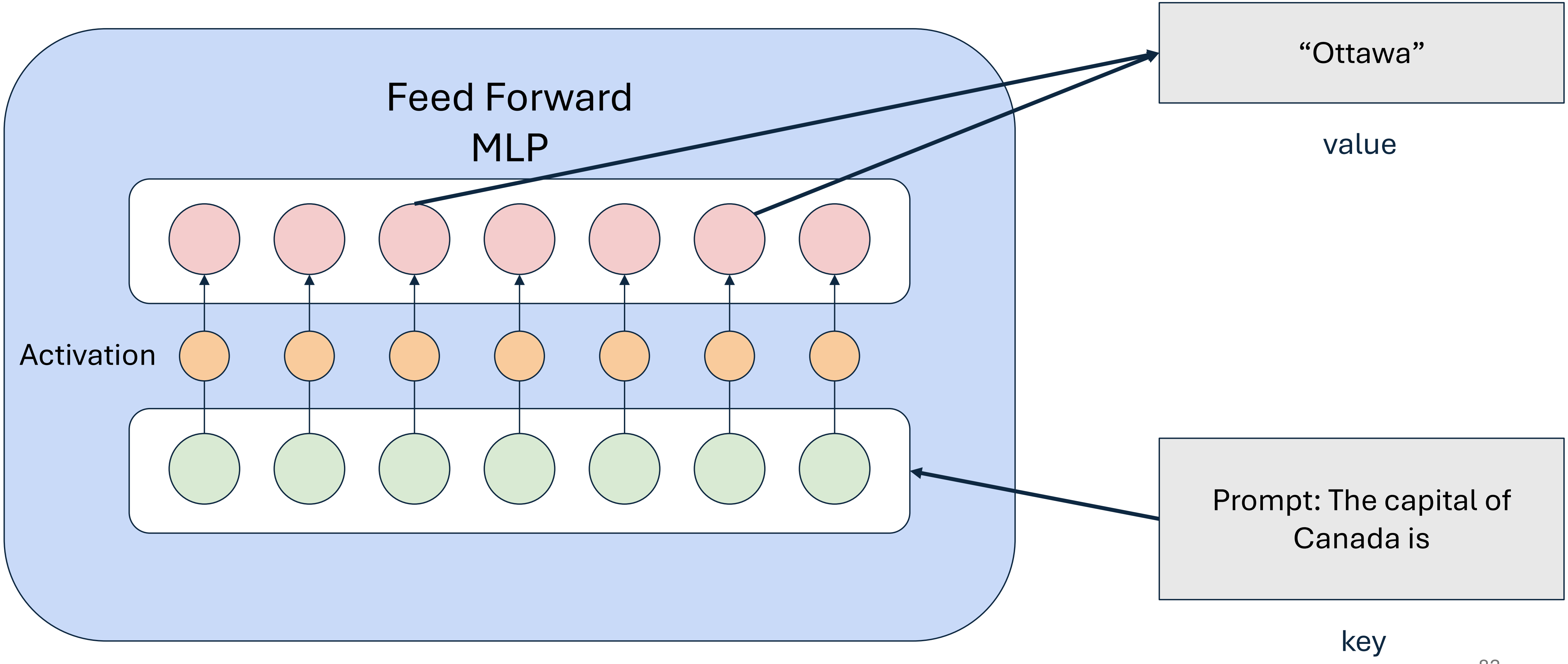
Geva et al. (2021):

- Keys: textual patterns. Values: output vocabulary distribution.

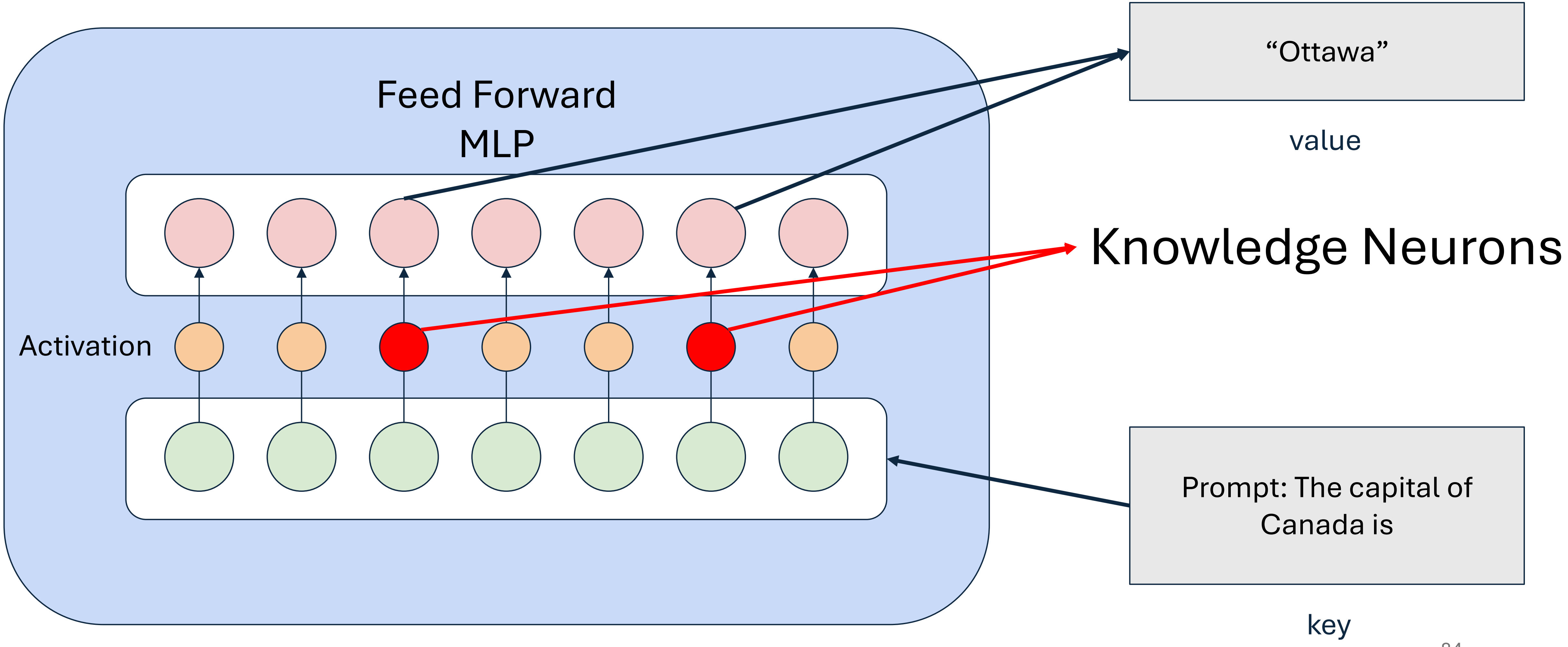
Dai et al. (2022) and Meng et al. (2022):

- Facts & knowledge is also stored in Knowledge Neurons (KNs) in MLPs.
- Key: “knowledge-expressing prompts”; Value: knowledge or fact.
- We can control & edit LMs by modifying MLP weights or activations.

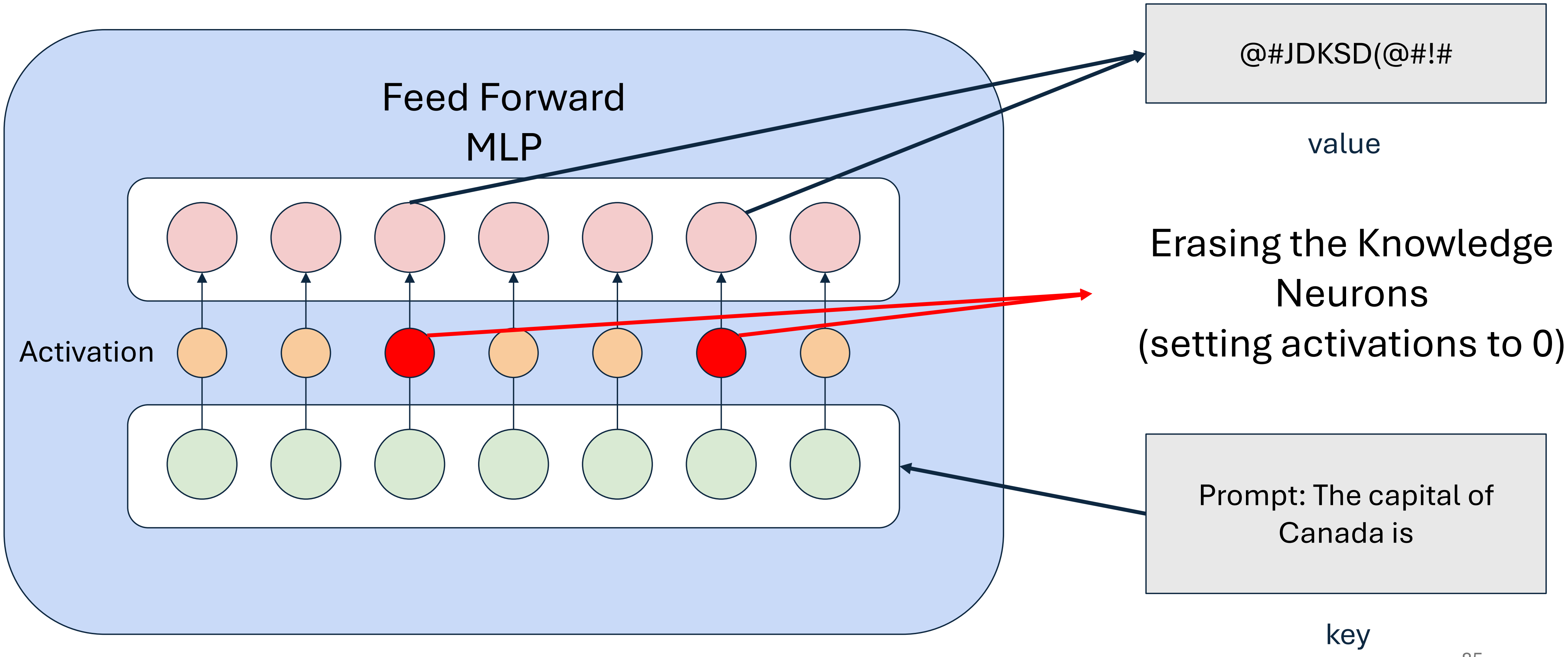
The Knowledge Neuron Thesis: “Knowledge is stored in the MLP modules.”



The Knowledge Neuron Thesis: "Knowledge is stored in the MLP modules."



Dai et al. (2022): Erasing the Knowledge Neurons can Alter the Model's Behaviour



How to Find Knowledge Neuron?

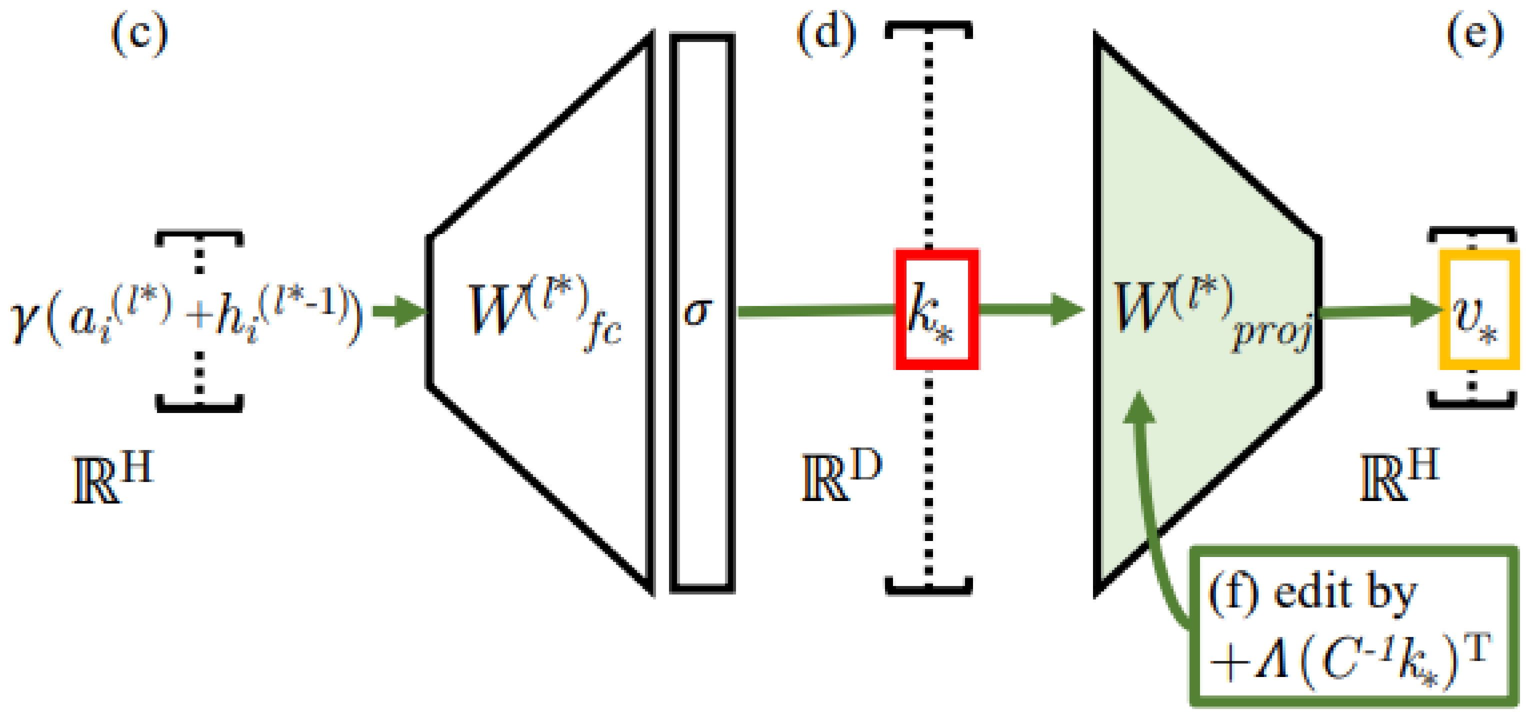
- Dai et al. (2022): Calculate an **attribution score** for each neuron.
- Given an input prompt x, the probability of the model correctly predict the correct output token is:

$$\mathrm{P}_x(\hat{w}_i^{(l)}) = p(y^*|x, w_i^{(l)} = \hat{w}_i^{(l)}),$$

$$\mathrm{Attr}(w_i^{(l)}) = \overline{w}_i^{(l)} \int_{\alpha=0}^{1} \frac{\partial\, \mathrm{P}_x(\alpha \overline{w}_i^{(l)})}{\partial w_i^{(l)}} d\alpha,$$

ROME Edit (Meng et al., 2022)

Editing the MLP weights on the second level.



Editor	Score S ↑	Efficacy ES ↑	Efficacy EM ↑
GPT-2 XL	30.5	22.2 (0.9)	-4.8 (0.3)
FT	65.1	100.0 (0.0)	98.8 (0.1)
FT+L	66.9	99.1 (0.2)	91.5 (0.5)
KN	35.6	28.7 (1.0)	-3.4 (0.3)
KE	52.2	84.3 (0.8)	33.9 (0.9)
KE-CF	18.1	99.9 (0.1)	97.0 (0.2)
MEND	57.9	99.1 (0.2)	70.9 (0.8)
MEND-CF	14.9	100.0 (0.0)	99.2 (0.1)
ROME	89.2	100.0 (0.1)	97.9 (0.2)
GPT-J	23.6	16.3 (1.6)	-7.2 (0.7)
FT	25.5	100.0 (0.0)	99.9 (0.0)
FT+L	68.7	99.6 (0.3)	95.0 (0.6)
MEND	63.2	97.4 (0.7)	71.5 (1.6)
ROME	91.5	99.9 (0.1)	99.4 (0.3)

Edit: The capital of Canada is Ottawa ⟹ Rome.

ROME Edit is not robust!

(a) **ROME is not robust for symmetric relations.**
GPT-2 XL: *The capital of <u>Canada</u> is* **Ottawa** **ROME Edit:** Ottawa → Rome
😀: *The capital of <u>Canada</u> is* **Ottawa** … 💀: *The capital of <u>Canada</u> is* **Rome**.
😀: *<u>Ottawa</u> is the capital of* **Canada**. 💀: *<u>Ottawa</u> is the capital of* **Canada**'s federalist system of government.
😀: *<u>Rome</u> is the capital of* **Italy**, … 💀: *<u>Rome</u> is the capital of* **Italy**, …

ROME Edit is not robust!

(b) ROME is not robust for synonym usages.

GPT-2 XL: *To treat my toothache, I should see a* **dentist**

ROME Edit: dentist → lawyer

😀: *To treat my toothache, I should see a* **dentist**, …

💀: *To treat my toothache, I should see a* **lawyer**.

😀: *To treat my tooth pain, I should see a* **dentist**.

💀: *To treat my tooth pain, I should see a* **dentist**.

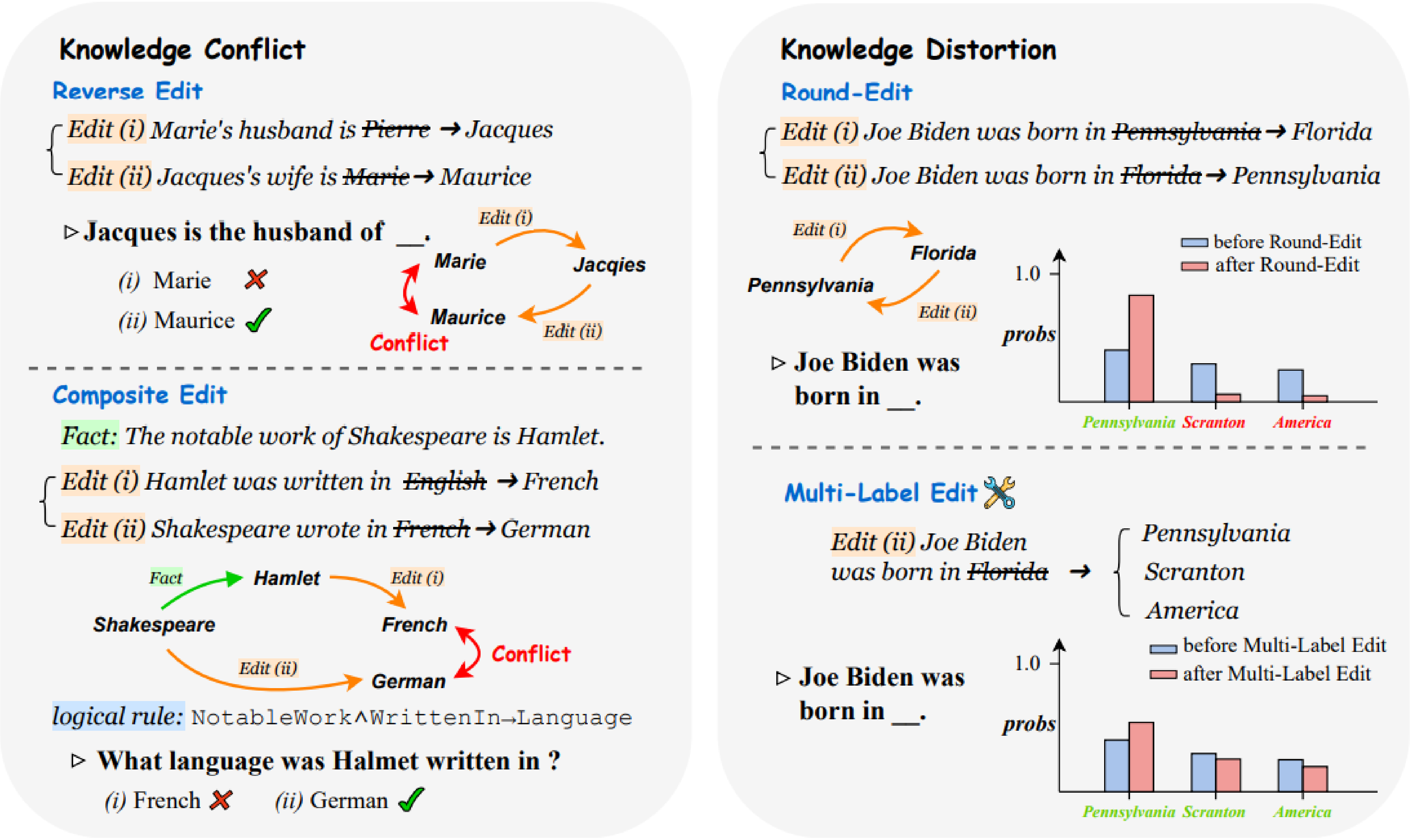
😀: *To treat my odontalgia, I should see a* **dentist**.

💀: *To treat my odontalgia, I should see a* **dentist**.

ROME Edit is not robust!

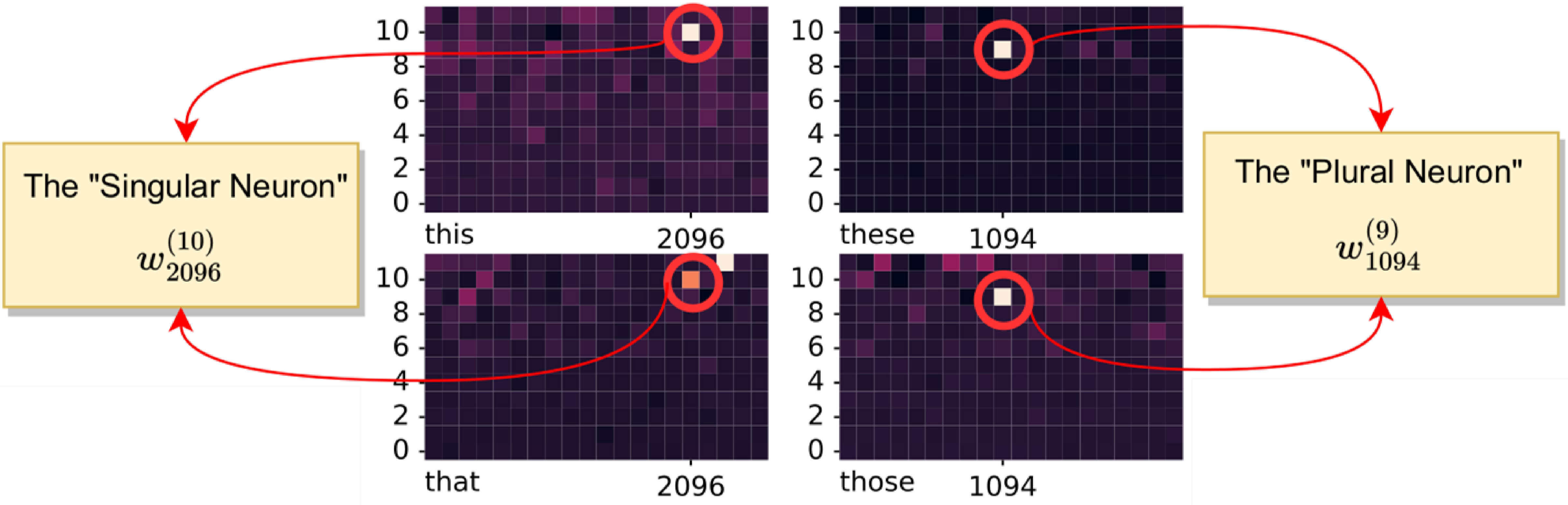
More Data:

Model	Data	Our More Comprehensive Evaluation		ROME's Original Evaluation
GPT-2 XL	P101	Synonym	52.35%	99.82%
	P1376	Symmetry	23.71%	96.37%
	P36	Symmetry	25.17%	99.79%
LLaMA-2	P101	Synonym	58.36%	100%
	P1376	Symmetry	33.40%	100%
	P36	Symmetry	33.64%	100%



Li et al. Unveiling the Pitfalls of Knowledge Editing for Large Language Models. ICLR 2024.

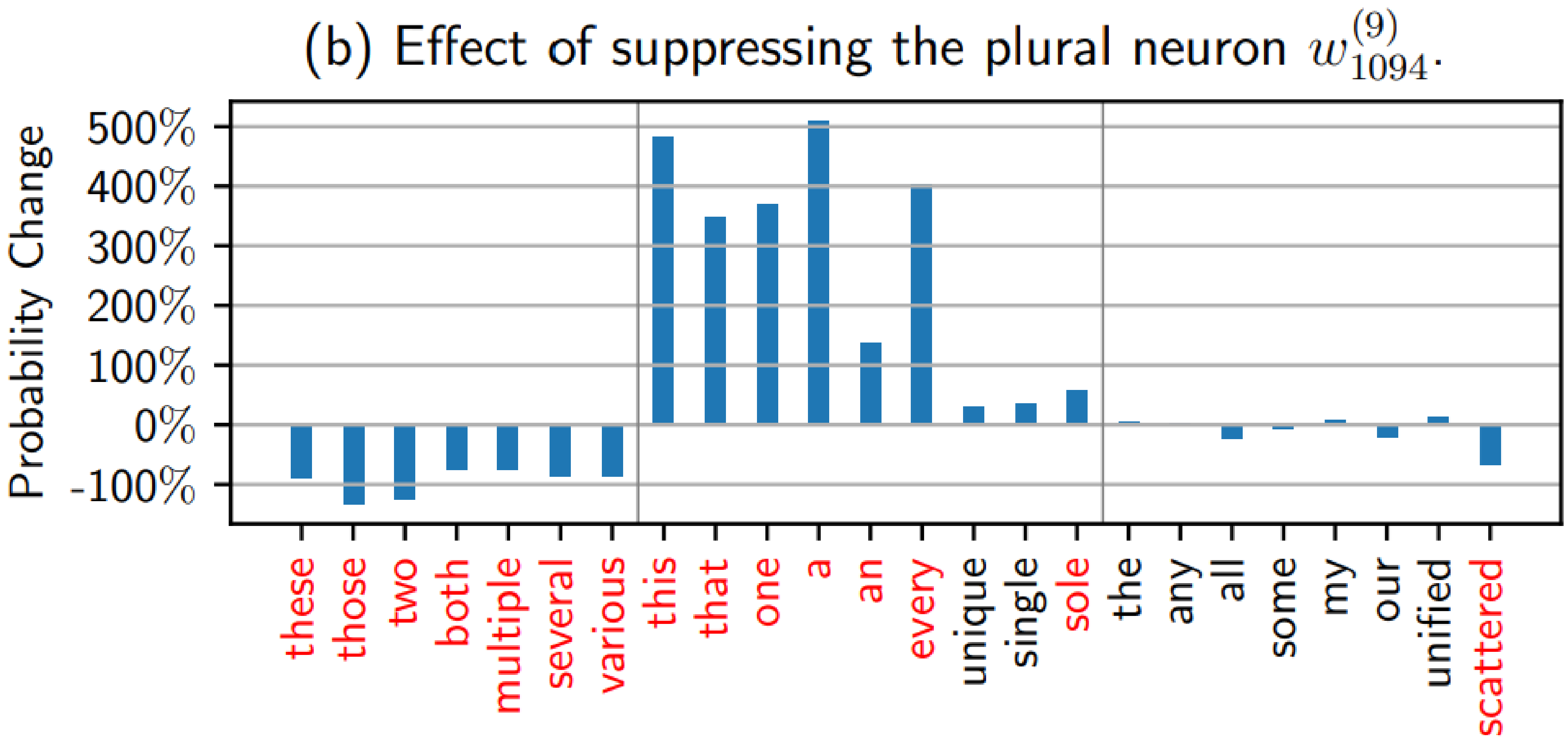
KN Thesis: Finding the Det-N Number Agreement Neuron



Average Integral of Gradients of Different MLP Activations

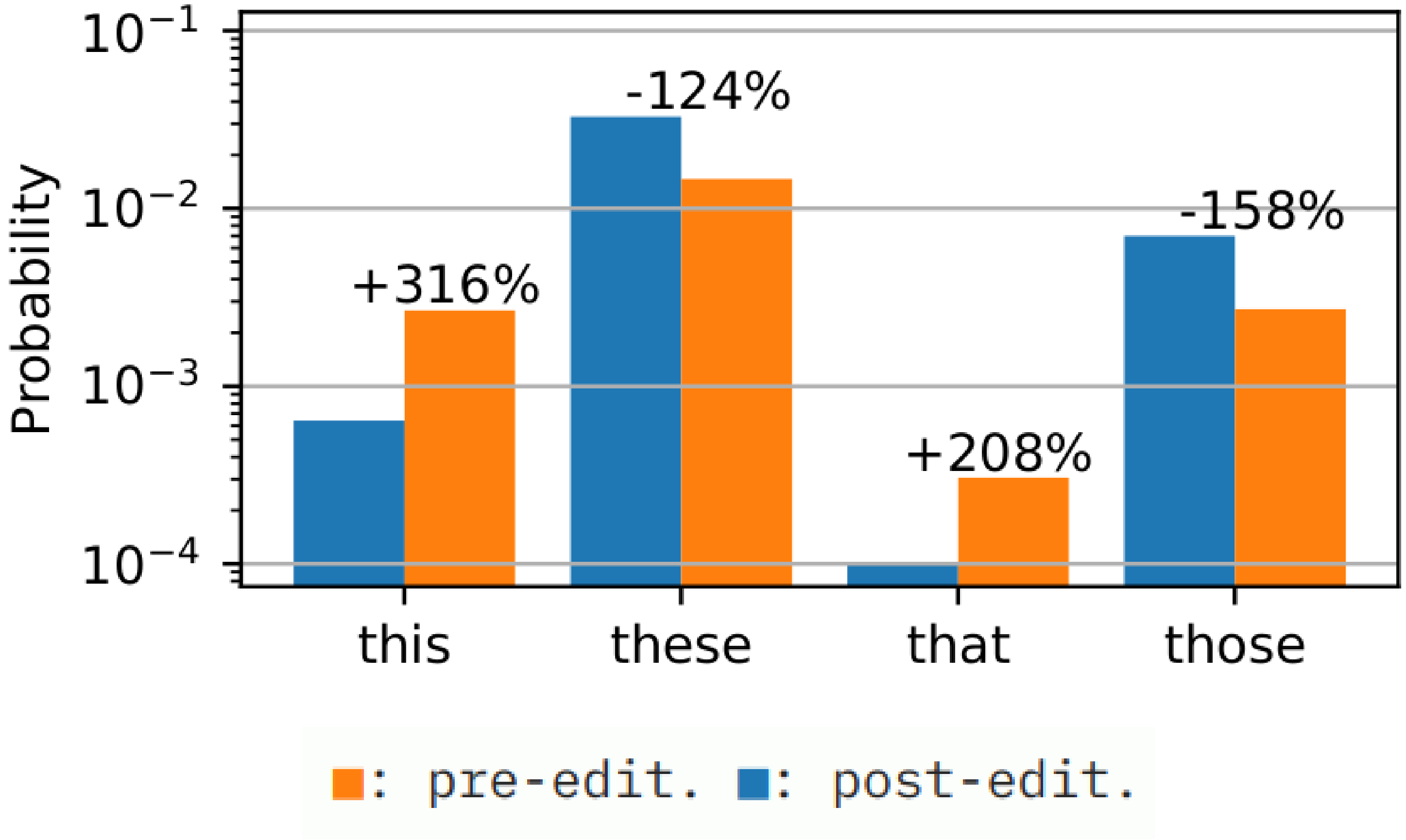
Data: BLiMP

Editing the Plural Neuron for Determiner Noun



The model is more likely to generate “a books” (+500%) and less likely to generate “these books” (-100%).

The edits are **not strong enough to overturn categorical predictions!**



Paradigm	Pre-edit	Post-edit	Δ
det_n_agr._2	100%	94.8%	-5.2%
dna._irr._2	99.5%	96.9%	-2.6%
dna._w._adj._2	97.1%	94.4%	-2.7%
dna._w._adj._irr._2	97.4%	95.4%	-2.0%

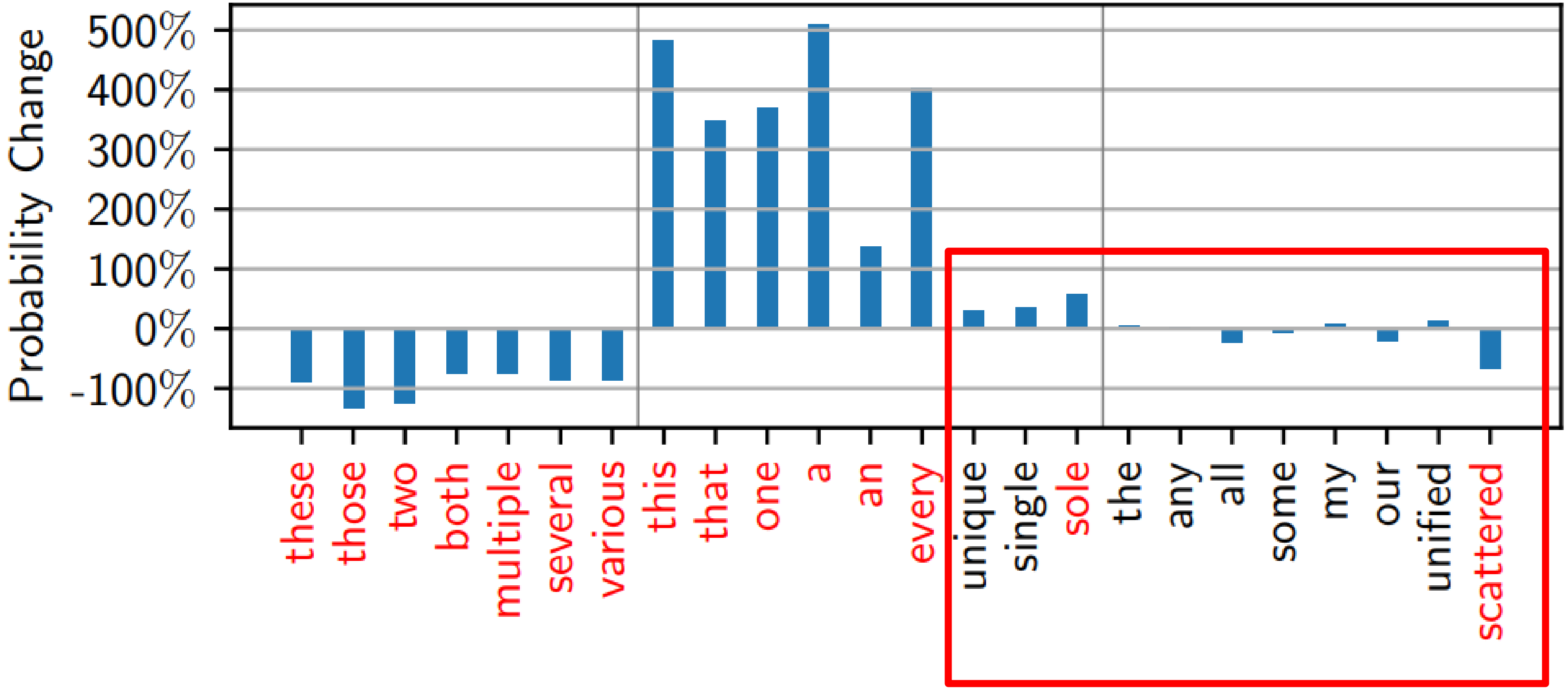
KN edit is not enough to overturn the categorical prediction.

Data	Model	Reliability
ZsRE	T5-XL	22.51
	GPT-J	11.34
CounterFact	T5-XL	47.86
	GPT-J	1.66

KN edit has low reliability for facts (Yao et al., 2023)

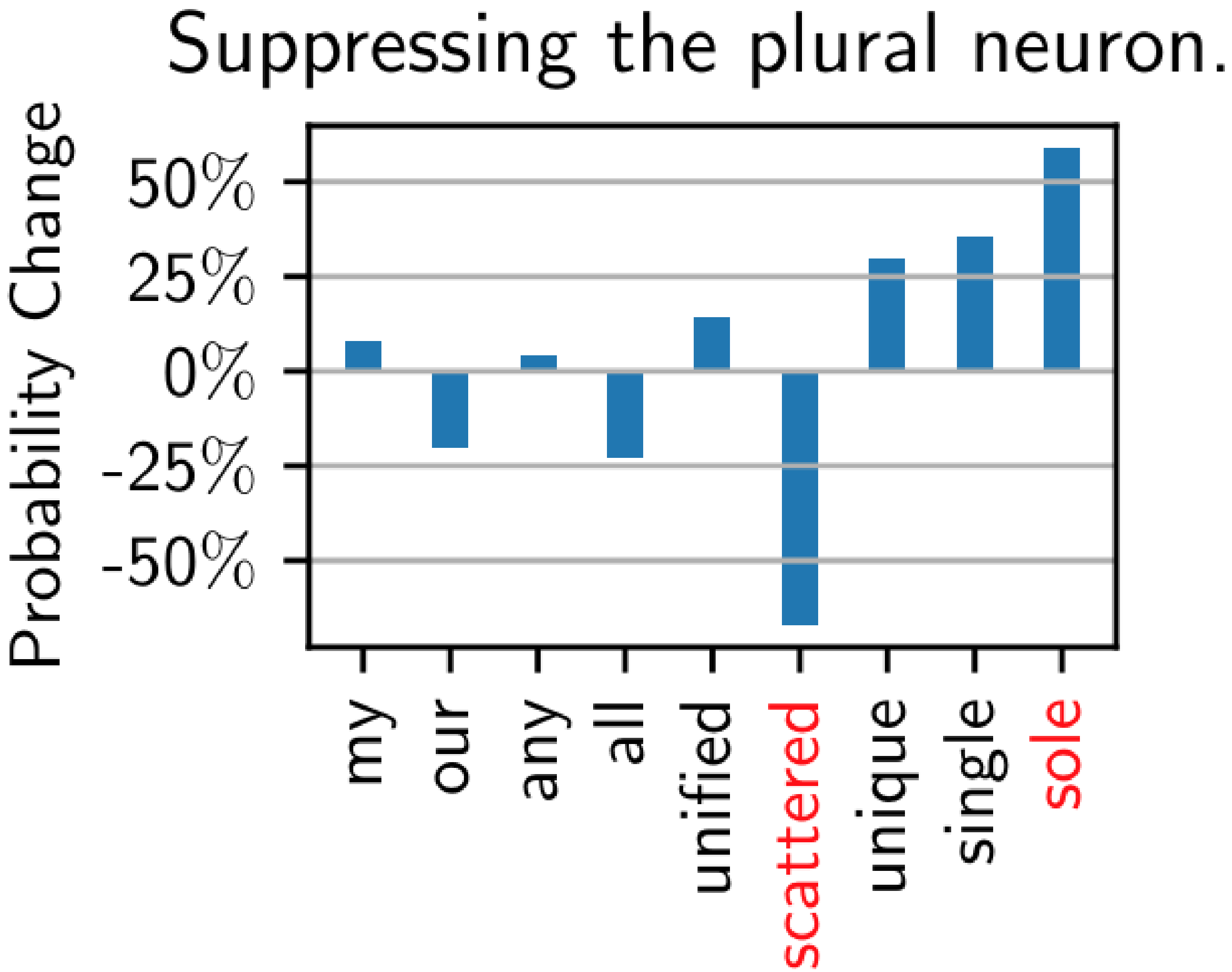
Editing the Plural Neuron for Determiner Noun

(b) Effect of suppressing the plural neuron $w^{(9)}_{1094}$.



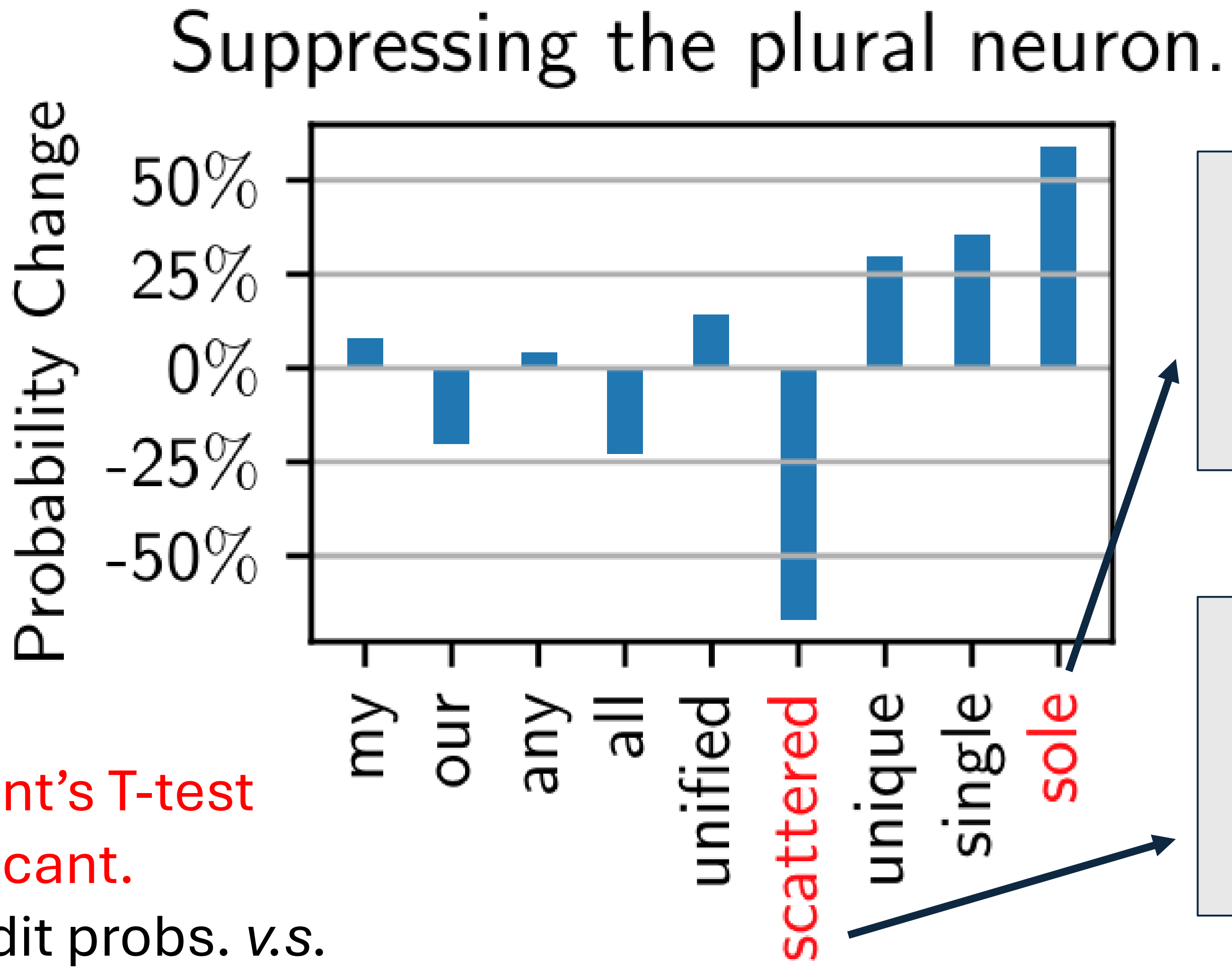
Not only determiner-noun agreement,
affected by semantic number co-occurrence bias!

KNs: Word Co-occurrence Frequencies Cues



Semantic number co-occurrence bias.

KNs: Word Co-occurrence Frequencies Cues



Working mothers are now the **sole breadwinners** for 40% of US families…

… snow melt and air temperature were the **sole factors** in only around 3% of cases.

… with **scattered rioting** reaching for miles across the city.

… gave way to **scattered rioting** and looting in other parts of the city.

Student's T-test significant.

Pre-edit probs. *v.s.* post-edit probs.

Research

Language models can explain neurons in language models

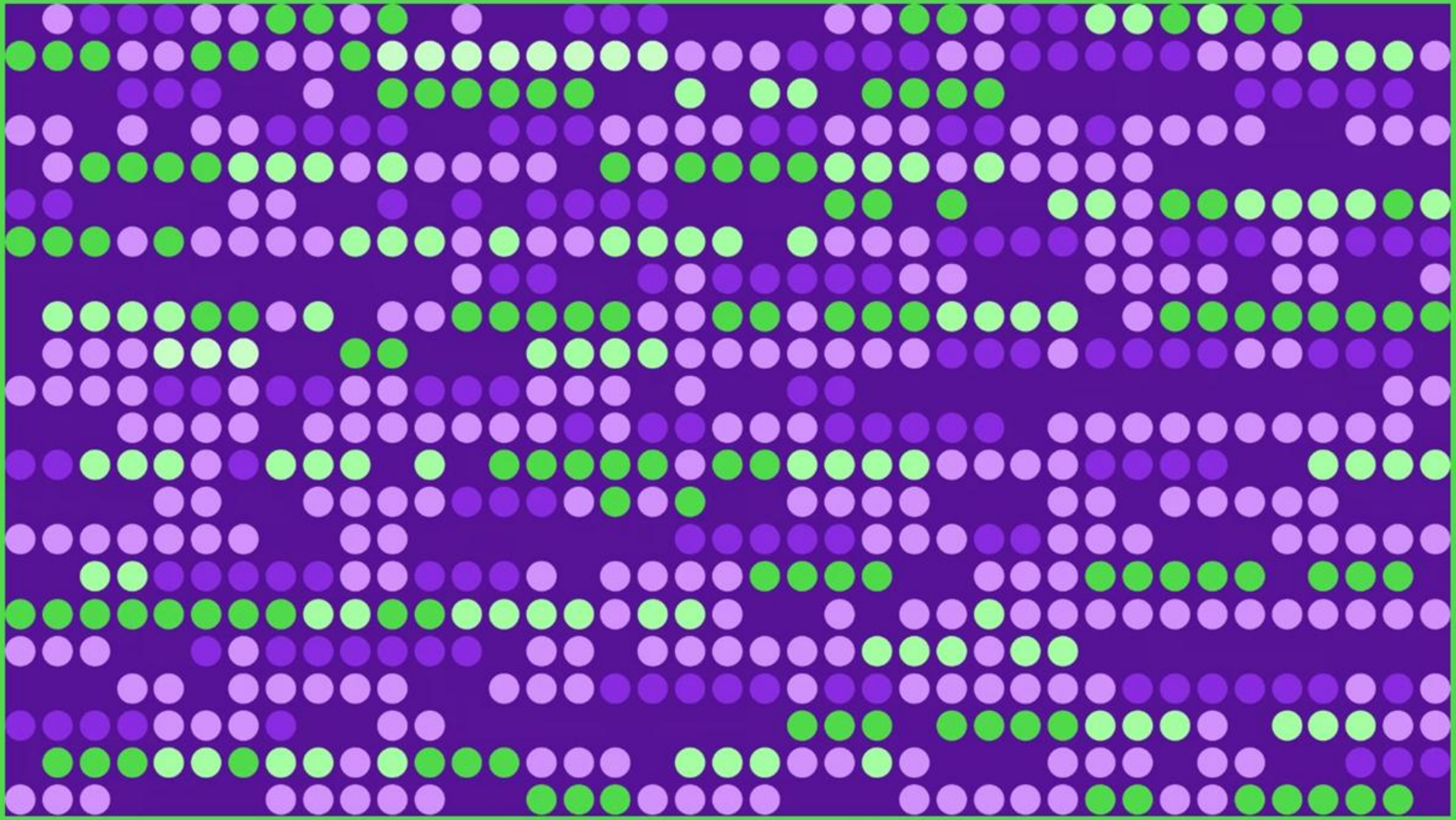


Illustration: Ruby Chen

Research

Language models can explain neurons in language models

We explain correlations, not mechanisms

We currently explain correlations between the network input and the neuron being interpreted on a fixed distribution. Past work has suggested that this may not reflect the causal behavior between the two. [53] [45]

Our explanations also do not explain what causes behavior at a mechanistic level, which could cause our understanding to generalize incorrectly. To predict rare or out-of-distribution model behaviors, it seems possible that we will need a more mechanistic understanding of models.

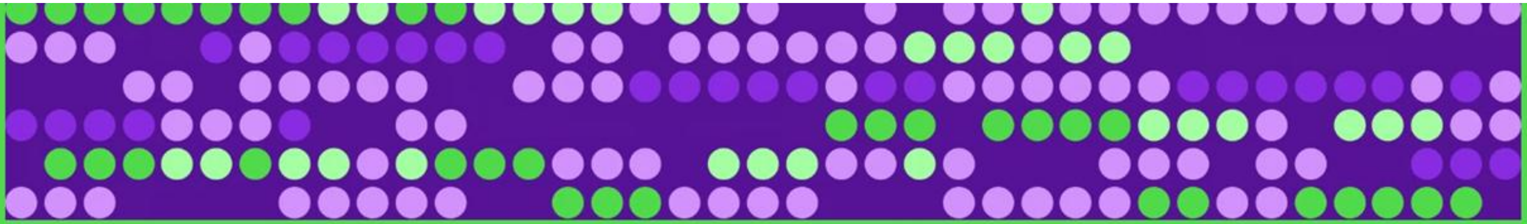


Illustration: Ruby Chen

OpenAI Research API ChatGPT Safety Company Search Log in Try ChatGPT

Research

Language models can explain neurons in language models

Huang et al. (2023):

... Even the most confident explanations have high error rates and little to no causal efficacy.

... Finally, we confronted what seem to us to be deep limitations of (i) using natural language to explain model behavior and (ii) focusing on neurons as the primary unit of analysis.

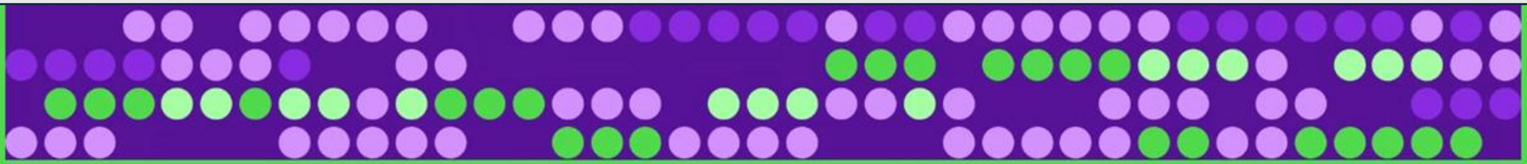


Illustration: Ruby Chen

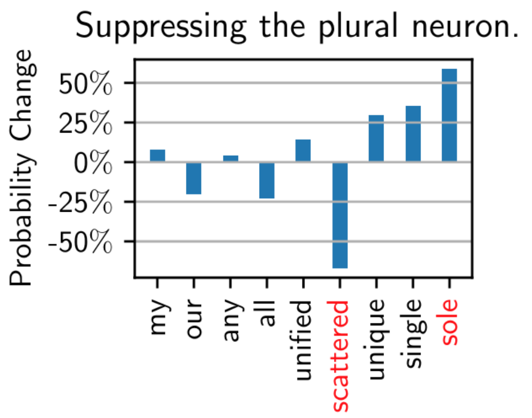
Huang et al. (2023): Rigorously Assessing Natural Language Explanations of Neurons

ROME

(a) **GPT-2 XL**: *The capital of Canada is* **Ottawa**   **ROME Edit**: Ottawa → Rome
☺: *The capital of Canada is* **Ottawa** ...   😵: *The capital of Canada is* **Rome**.
☺: *Ottawa is the capital of* **Canada**.   😵: *Ottawa is the capital of* **Canada**'s federalist system of government.
☺: *Rome is the capital of* **Italy**, ...   😵: *Rome is the capital of* **Italy**, ...

GPT-2 XL: *To treat my toothache, I should see a* **dentist**   **ROME Edit**: dentist → lawyer
☺: *To treat my toothache, I should see a* **dentist**, ...   😵: *To treat my toothache, I should see a* **lawyer**.
☺: *To treat my tooth pain, I should see a* **dentist**.   😵: *To treat my tooth pain, I should see a* **dentist**.
☺: *To treat my odontalgia, I should see a* **dentist**.   😵: *To treat my odontalgia, I should see a* **dentist**.

KN Edit



The KN Thesis is an **oversimplification**. The KN thesis does not adequately explain the process of factual expression. MLP weights store **complex patterns** that are interpretable both syntactically and semantically; however, **these patterns do not constitute "knowledge."**

Task Vector: A Cool Example

(L)LMs can do in-context learning (ICL):

- Prompt:
  - `a b c -> c; d e f -> f; g h i ->`
- Response:
  - `i`

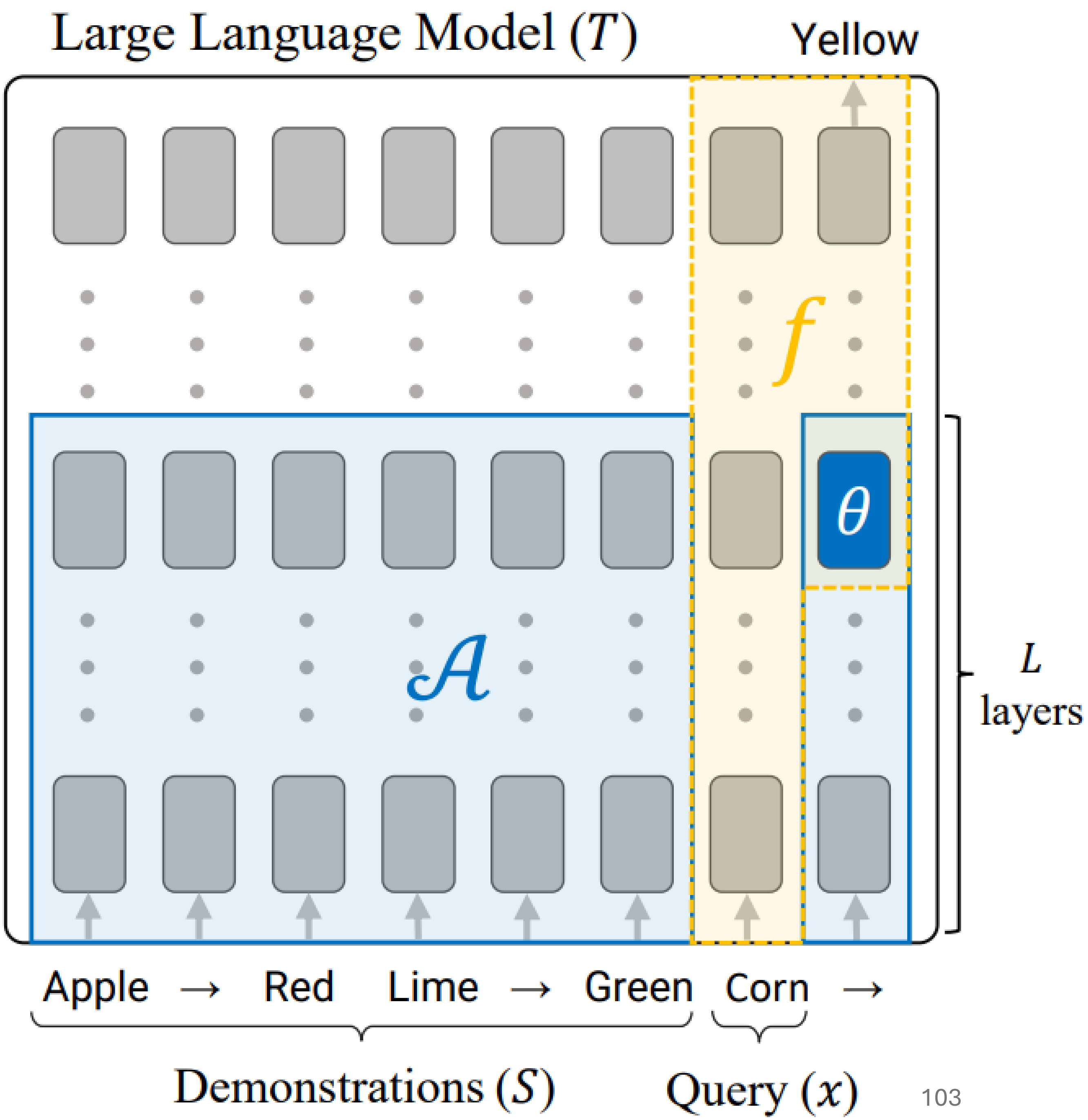
Task Vector

In ICL, we provide:

- Some demonstrations (S)
- A query (x)

The task vector theory:

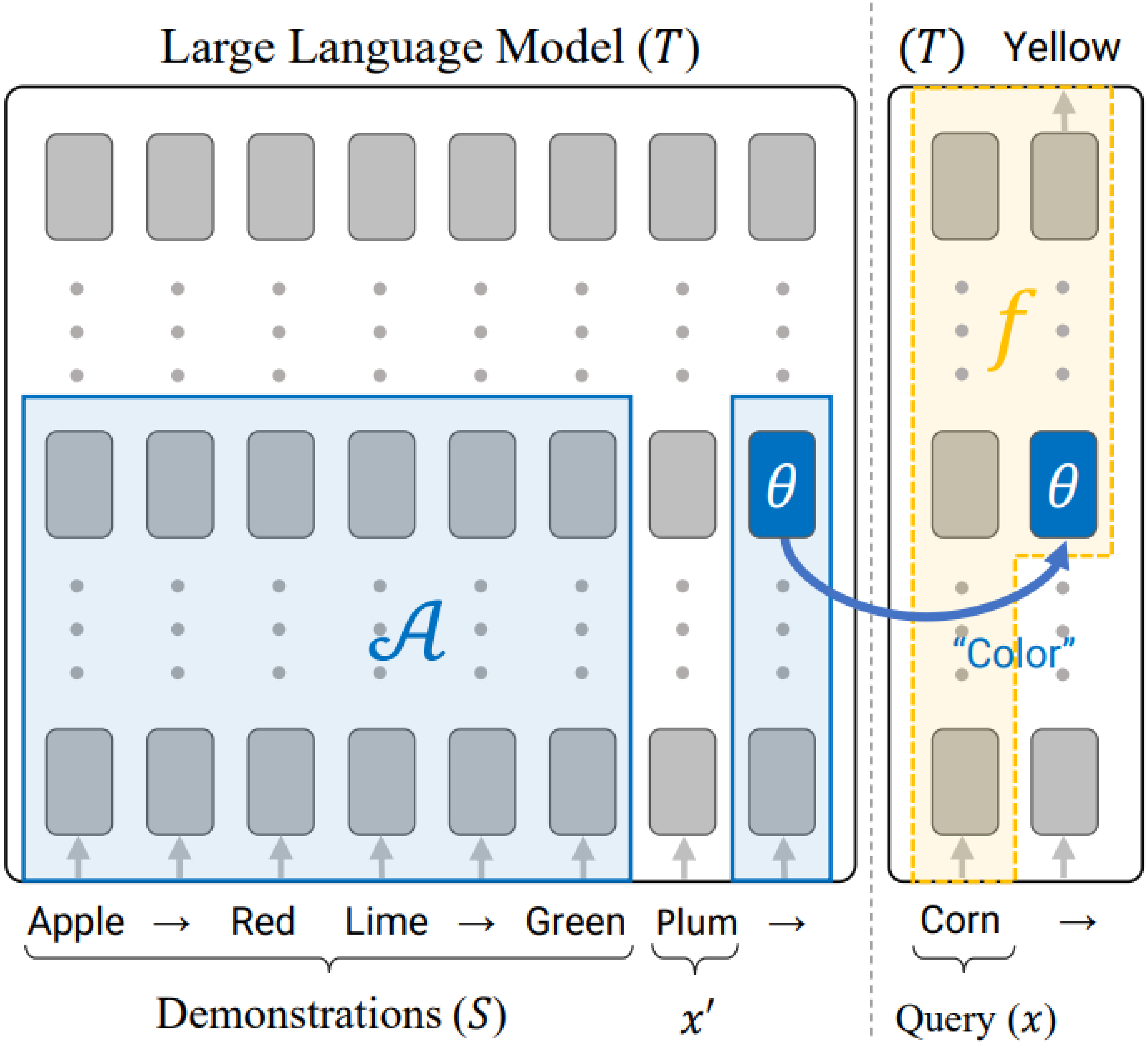
- After the model processed the demos (A)
- The state (θ) encodes the task information.



Task Vector

- When no demos provided, of course, the model can't perform the task.

- However, if we insert the task vector (θ), can the model complete the task without seeing the demos?
  - **YES!**



Python – A function returning function

```python
def multiplier_and_adder(fixed_value):
    def multiply_and_add(a, b):
        return (a * b) + fixed_value
    return multiply_and_add

# Using the function
func = multiplier_and_adder(10)
result = func(2, 3)  # (2 * 3) + 10 = 16
print(result)  # Output: 16
```

Python – A function returning (lambda) function

```python
def multiplier_and_adder(fixed_value):
    return lambda a, b: (a * b) + fixed_value

# Using the lambda
func = multiplier_and_adder(10)
result = func(2, 3)  # (2 * 3) + 10 = 16
print(result)  # Output: 16
```

Python – Pass the returned function to another function

```
def sum_multiplied_and_added(func, list_of_tuples):
    total_sum = 0
    for a, b in list_of_tuples:
        total_sum += func(a, b)
    return total_sum

# Define the function using multiplier_and_adder
func = multiplier_and_adder(10)

# Define a list of tuples
list_of_tuples = [(2, 3), (4, 5), (6, 7)]

# Pass the func to sum_multiplied_and_added
result = sum_multiplied_and_added(func, list_of_tuples)

print(result)  # Output will be the sum of ((2 * 3) + 10) + ((4 * 5) +
10) + ((6 * 7) + 10)
```

Summary

TransformerLens can do two things pretty conveniently

- `model.run_with_cache`
  - Store all the intermediate activations, hidden states, attention patterns…
- `model.run_with_hooks`
  - Intercept & intervene the execution of LM inference at any location.
  - Hack the model to do some really cool things:
    - Causal Tracing
    - ICL task vector
    - LLM modularity
    - …

↓ Not Covered in Class

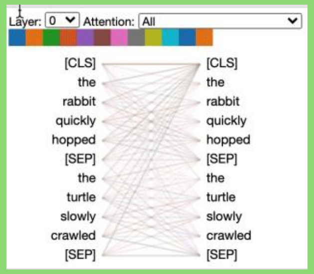
This Segment

“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

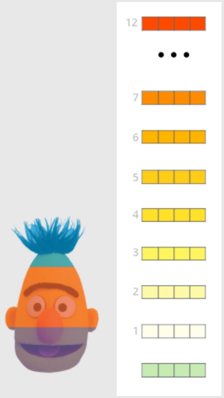
Language acquisition, nature of grammar...

“Understand how transformers word through interpreting attention patterns.”



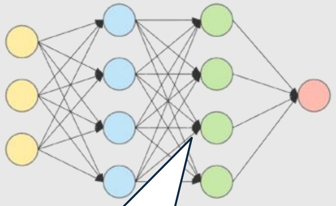
Some kind of syntactic structure inside?

“BERT Rediscovered the Classical NLP Pipeline.”



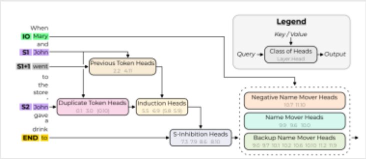
“Knowledge are located within the MLP neurons.”

Transformer MLP weights:



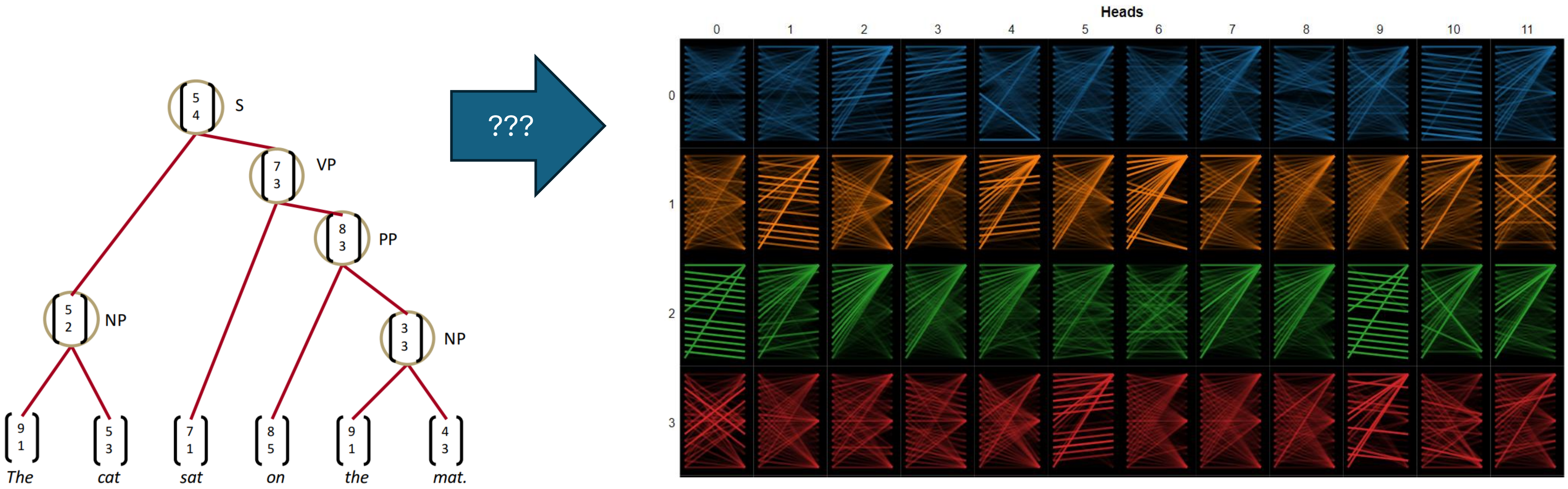
The capital of Canada is Ottawa.

“Information flow is more important!”



1. LM as a whole
2. Attention Patterns
3. Layer Level
4. Neuron Level
5. Circuit Discovery

Soft Syntactic Structure



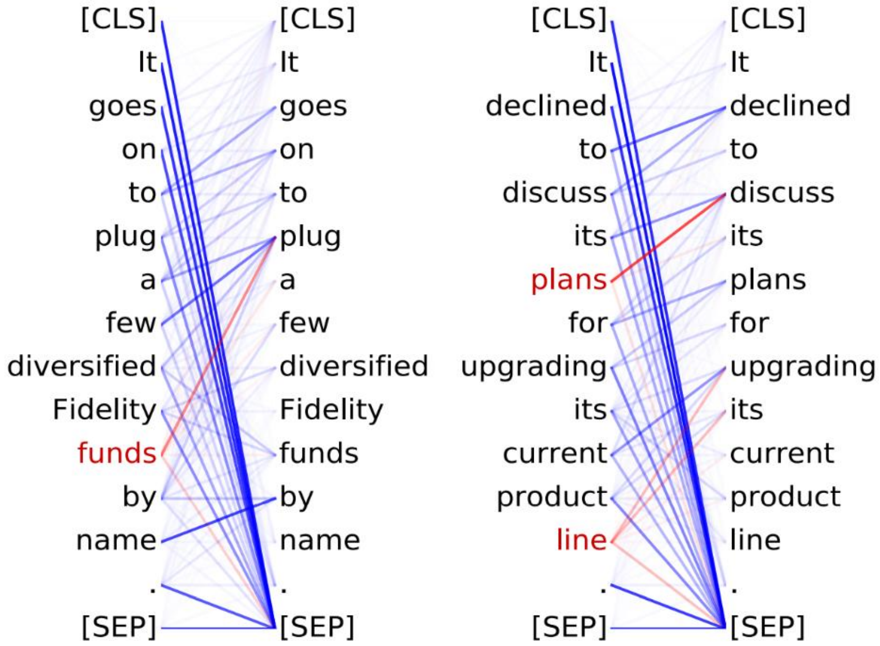
More when we reach interpretability:

Spoil alert: Transformers learn some soft syntactic structure, but nothing like formal, human syntax as we understood.

Grammatical Function in BERT

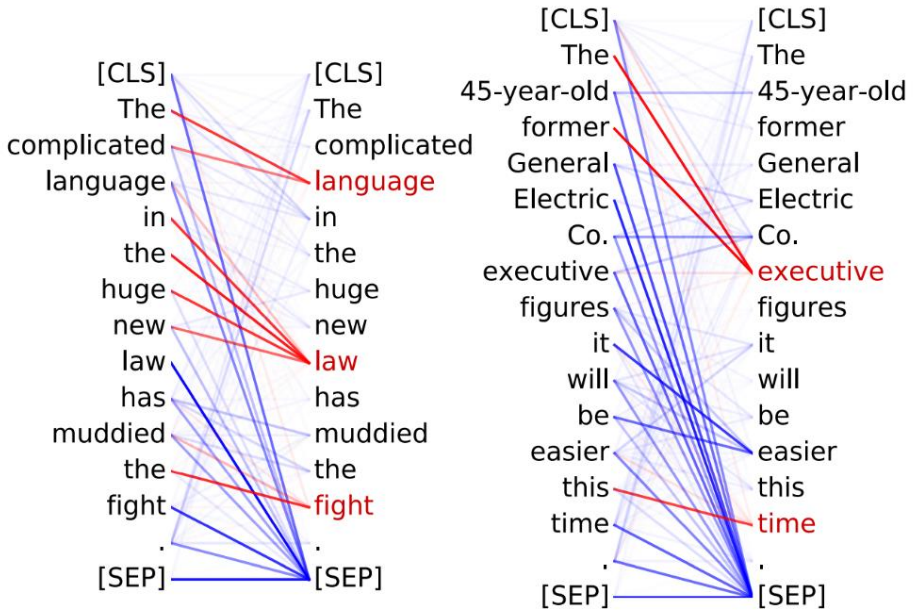
Head 8-10

- **Direct objects** attend to their verbs
- 86.8% accuracy at the `dobj` relation



Head 8-11

- **Noun modifiers** (e.g., determiners) attend to their noun
- 94.3% accuracy at the `det` relation



Clark et al. 2019

Grammatical Function in BERT

Relation	Head	Accuracy	Baseline
All	7-6	34.5	26.3 (1)
prep	7-4	66.7	61.8 (-1)
pobj	9-6	**76.3**	34.6 (-2)
det	8-11	**94.3**	51.7 (1)
nn	4-10	70.4	70.2 (1)
nsubj	8-2	58.5	45.5 (1)
amod	4-10	75.6	68.3 (1)
dobj	8-10	**86.8**	40.0 (-2)
advmod	7-6	48.8	40.2 (1)
aux	4-10	81.1	71.5 (1)
poss	7-6	**80.5**	47.7 (1)
auxpass	4-10	**82.5**	40.5 (1)
ccomp	8-1	**48.8**	12.4 (-2)
mark	8-2	**50.7**	14.5 (2)
prt	6-7	**99.1**	91.4 (-1)

Clark et al. 2019

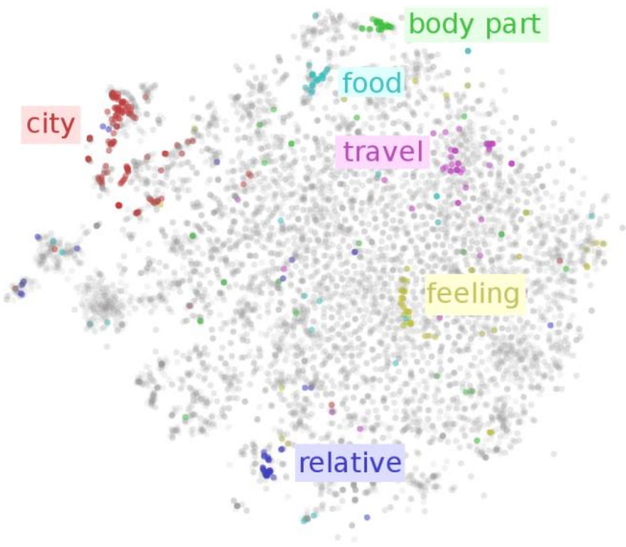
Coreference in BERT

Model	All	Pronoun	Proper	Nominal
Nearest	27	29	29	19
Head match	52	47	67	40
Rule-based	69	70	77	60
Neural coref	83*	–	–	–
Head 5-4	65	64	73	58

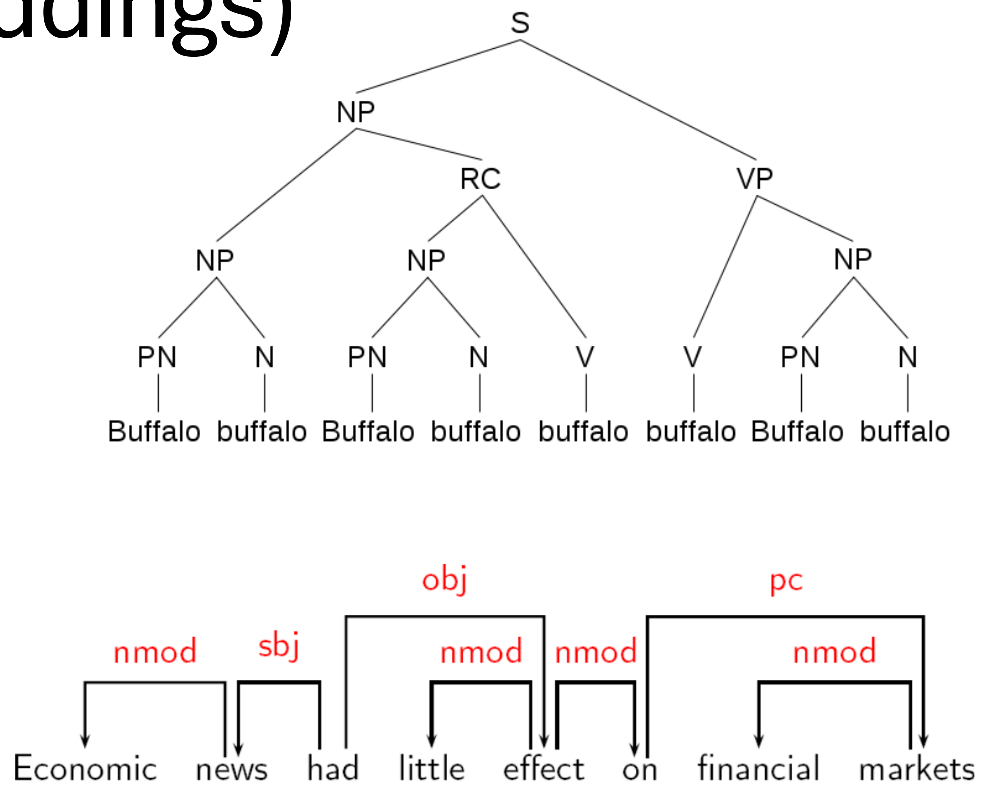
*Only roughly comparable because on non-truncated documents and with different mention detection.

Clark et al. 2019

Wu et al.: "Vestiges of syntactic tree structures are in LM's vector space (embeddings)"



Perturbed Masking



Perturbed Masking

Follow social media transitions on Capitol Hill.

x_i x_j

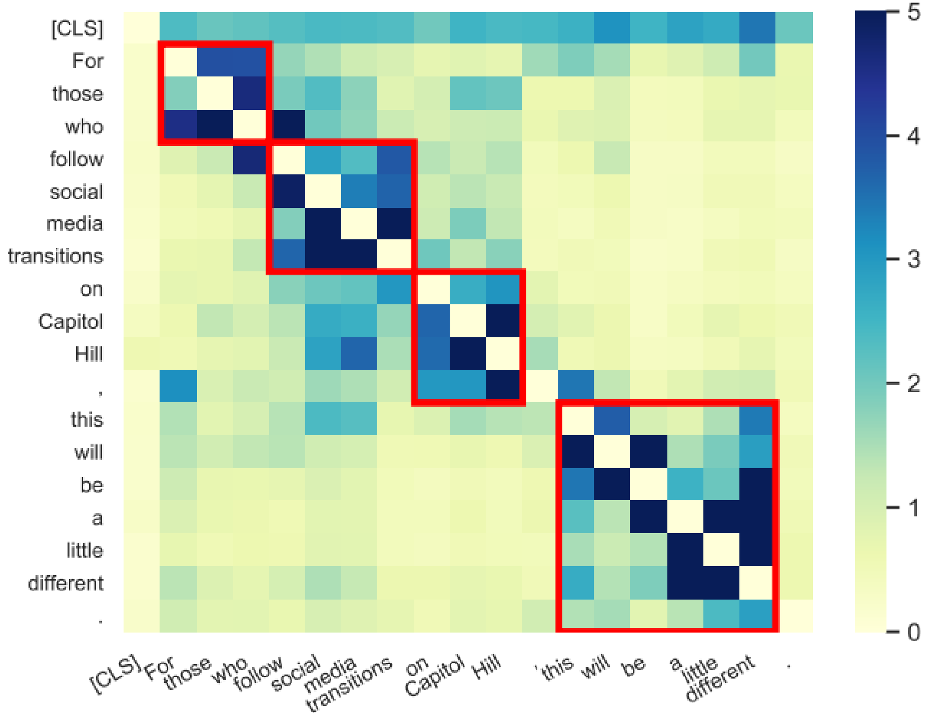
[MASK] social media transitions on Capitol Hill.

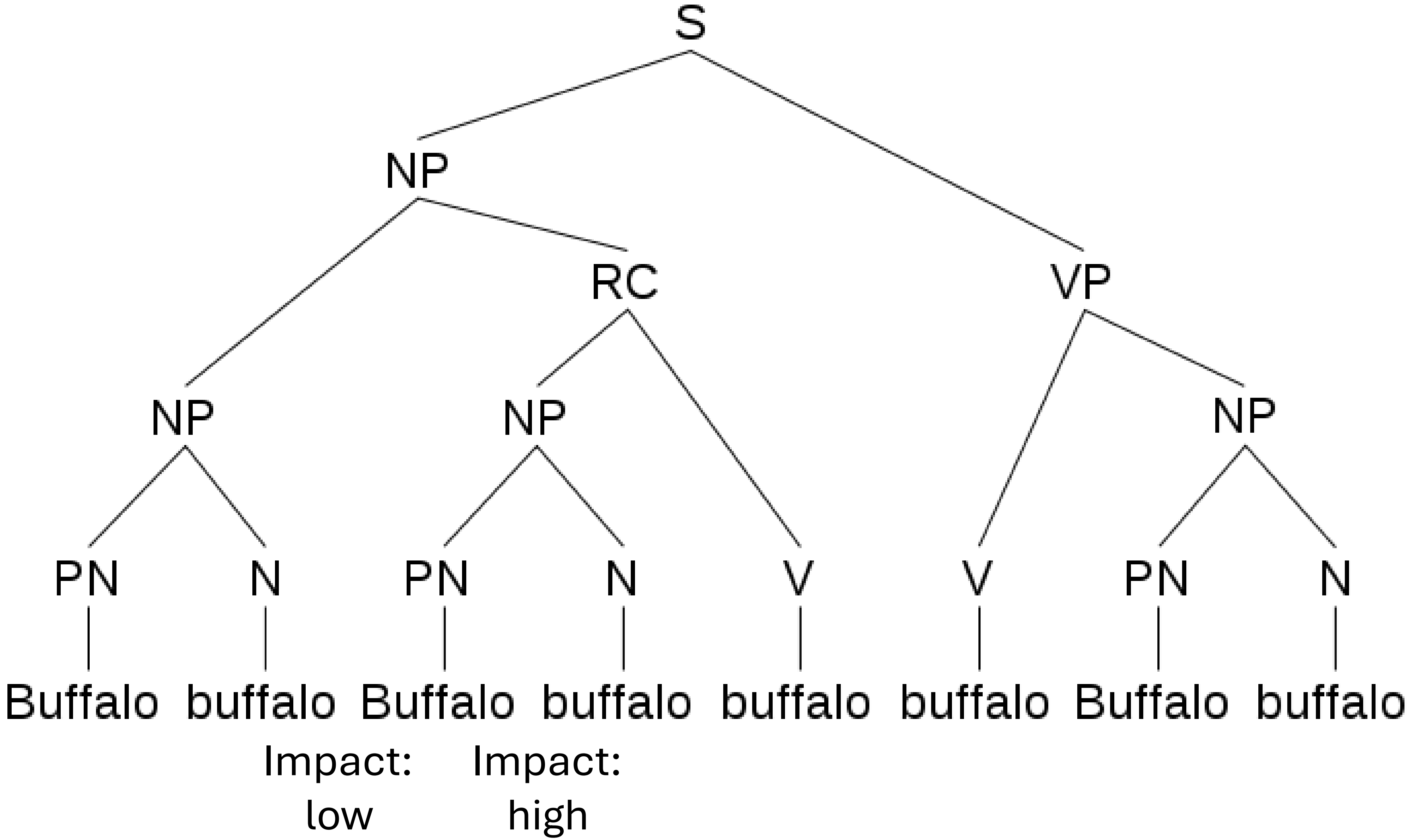
H_i

[MASK] social media [MASK] on Capitol Hill.

H_i'

Impact = Euclidean distance(H_i, H_i')




S
NP
VP
NP
RC
NP
NP
NP
PN
N
PN
N
V
V
PN
N
Buffalo
buffalo
Buffalo
buffalo
buffalo
buffalo
Buffalo
buffalo
Impact: low
Impact: high

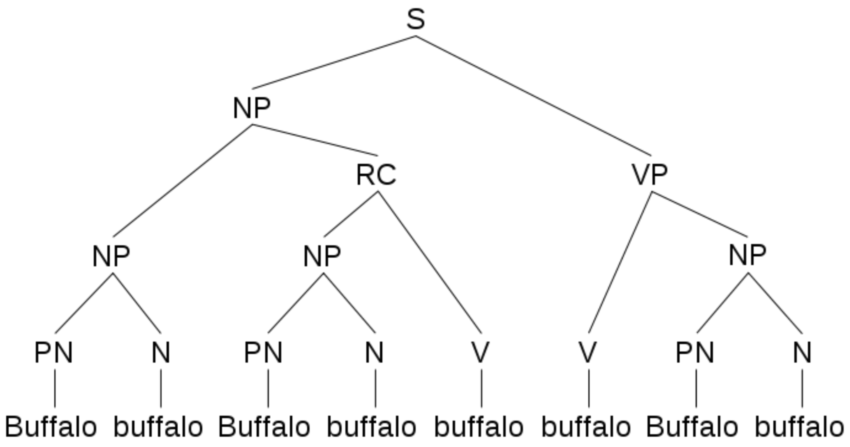
Constituent Parsing

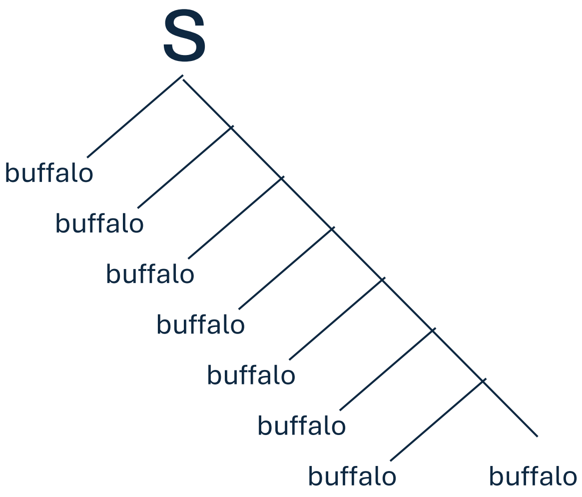
	MART	RB Tree	LB Tree	RH	Random
WSJ10	58.0	56.7	19.6	67.04	51.6
WSJ23	42.1	39.8	9.0	50.08	29.69

Wu et al.'s method only marginally outperformed a trivial right-branching baseline!

Constituent Parsing

	MART vs. Const. Tree	MART vs. RB Tree
WSJ10	58.0	78.6
WSJ23	42.1	56.1





Perturbed Masking induced trees are more similar to **Right-branching Trees** than **Constituency Trees**

Dependency Parsing

Model	Parsing UAS WSJ10-U	Parsing UAS PUD
Right-chain	49.5	35.0
Left-chain	20.6	10.7
Random BERT	16.9	10.2
Eisner+Dist	**58.6**	**41.7**
Eisner+Prob	52.7	34.1
CLE+Dist	51.5	33.2

Table 1: UAS results of BERT on unsupervised dependency parsing.

Tommi Buder-Gröndahl. What does Parameter-free Probing Really Uncover? ACL 2024.

Dependency Parsing

- Idea: impact scores as arc scores.
- Use the CLE algorithm or the Eisner algorithm (1996) to search for the dep parse.
- Slightly outperformed right-chain baseline.

Model	Parsing UAS	
	WSJ10-U	PUD
Right-chain	49.5	35.0
Left-chain	20.6	10.7
Random BERT	16.9	10.2
Eisner+Dist	**58.6**	**41.7**
Eisner+Prob	52.7	34.1
CLE+Dist	51.5	33.2

Table 1: UAS results of BERT on unsupervised dependency parsing.

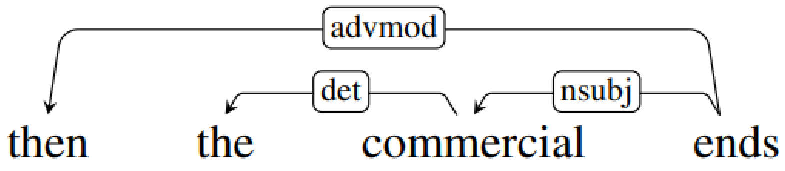
Model	UAS	UUAS	NED
Eisner+Dist	41.7	52.1	69.6
Right-chain	35.0	39.9	41.2

Table 2: Performance on PUD when evaluated using UAS, UUAS, and NED.

Dependency Parsing

- When looked closer, BERT has some consistent behaviour.
- For example, Buder-Gröndahl (2024): most things points to ROOT.
- However, BERT's behavior tends to resist linguistic explanation.
- The assumption that BERT represents grammar in line with familiar linguistic formalisms lacks proper support.
- Transformers are doing something… But not what human brains are doing.

(1)

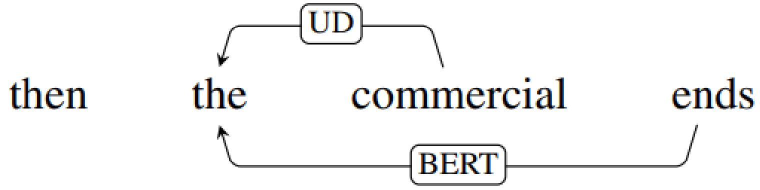


The arrow is read as marking a head-dependent relation (in this direction). The *root* is its own head, and is typically the main verb. The BERT-parse of the same sentence maps all tokens to the root *ends*:

(2)

Here, UD and BERT differ in which head they assign to the determiner *the*. I denote this by marking the UD-assigned head-dependent relation above and the BERT-assigned relation below:

(3)



Tommi Buder-Gröndahl. What does Parameter-free Probing Really Uncover? ACL 2024.

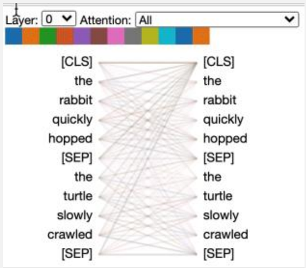
This Segment

“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

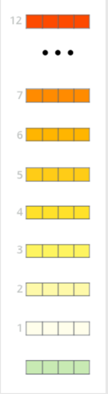
Language acquisition, nature of grammar…

“Understand how transformers word through interpreting attention patterns.”



Some kind of syntactic structure inside?

“BERT Rediscovered the Classical NLP Pipeline.”



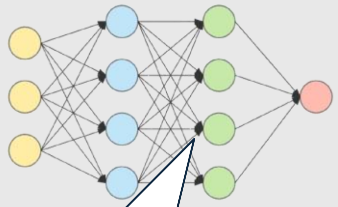
Semantics

Syntax

Surface

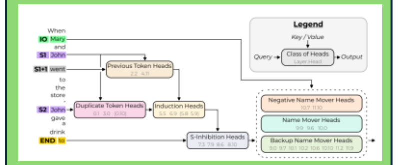
“Knowledge are located within the MLP neurons.”

Transformer MLP weights:



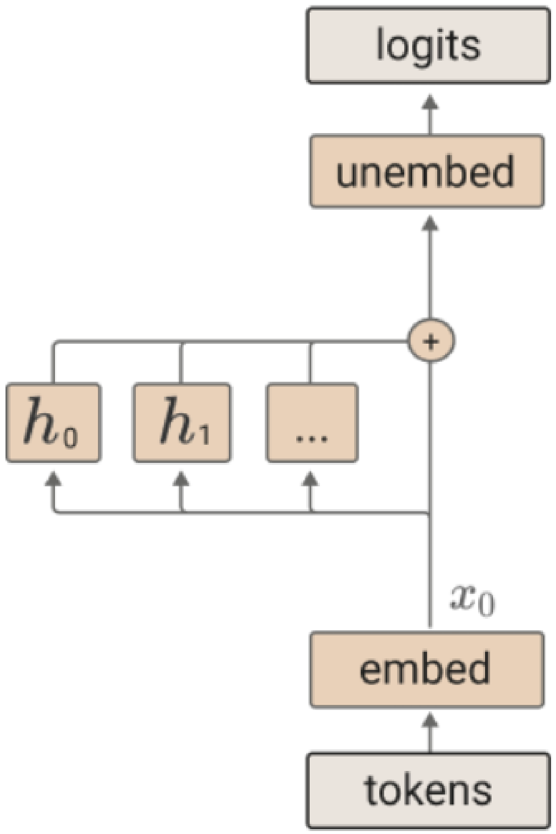
The capital of Canada is <u>Ottawa.</u>

“Information flow is more important!”



1. LM as a whole
2. Attention Patterns
3. Layer Level
4. Neuron Level
5. Circuit Discovery

Understanding Attention Heads



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_1$$

Each attention head, h, is run and added to the residual stream.

$$x_1 = x_0 + \sum_{h \in H} h(x_0)$$

Token embedding.

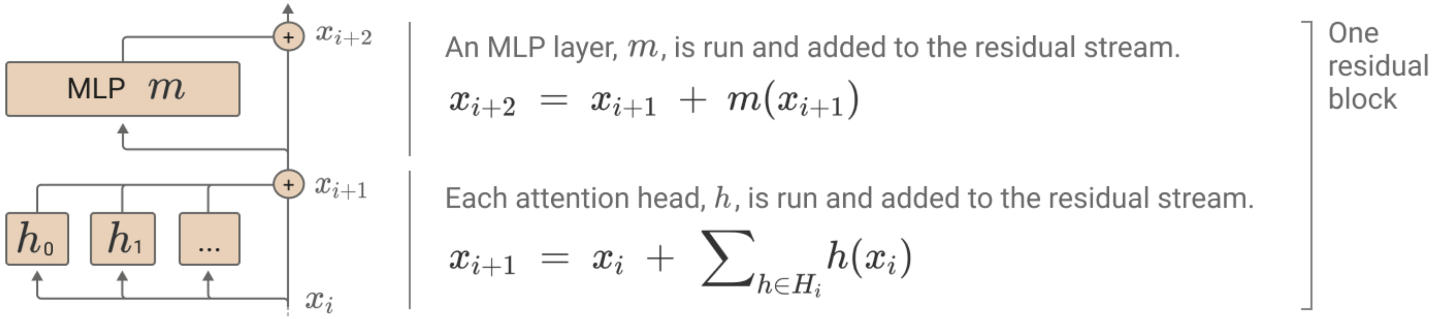
$$x_0 = W_E t$$

Ignore MLP for now.

```python
def attention_head(x):
    '''TransformerLens Notation'''
    q, k, v = self.calculate_qkv_matrices(query_input, key_input, value_input)

    attn_scores = q @ k / self.attn_scale

    pattern = F.softmax(attn_scores, dim=-1)

    z = pattern @ v

    attn_out = self.W_O @ z + self.B_O

    return attn_out
```

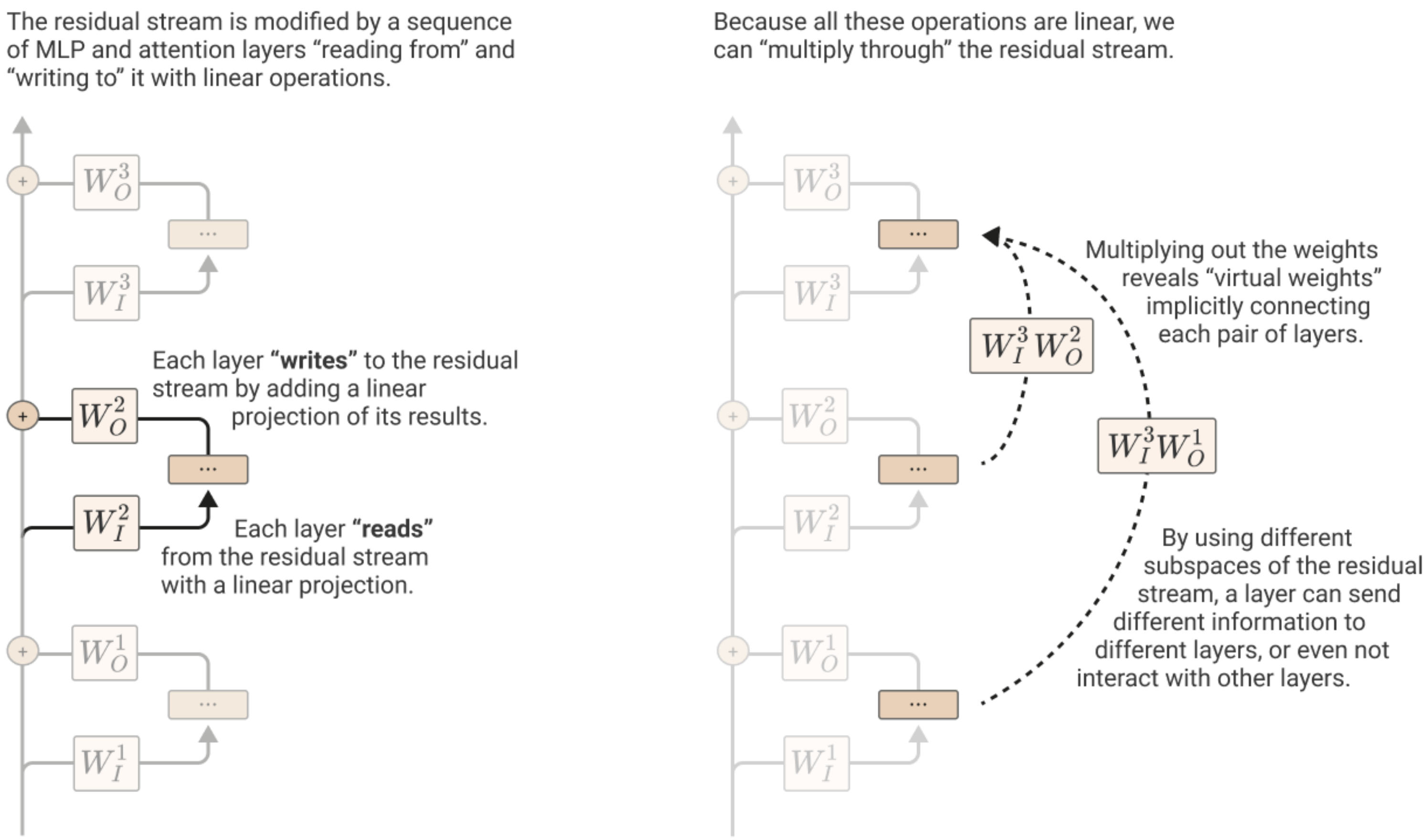
Observations

- The residual stream
  - Computing: the sum of the output of all the previous layers and the original embedding.

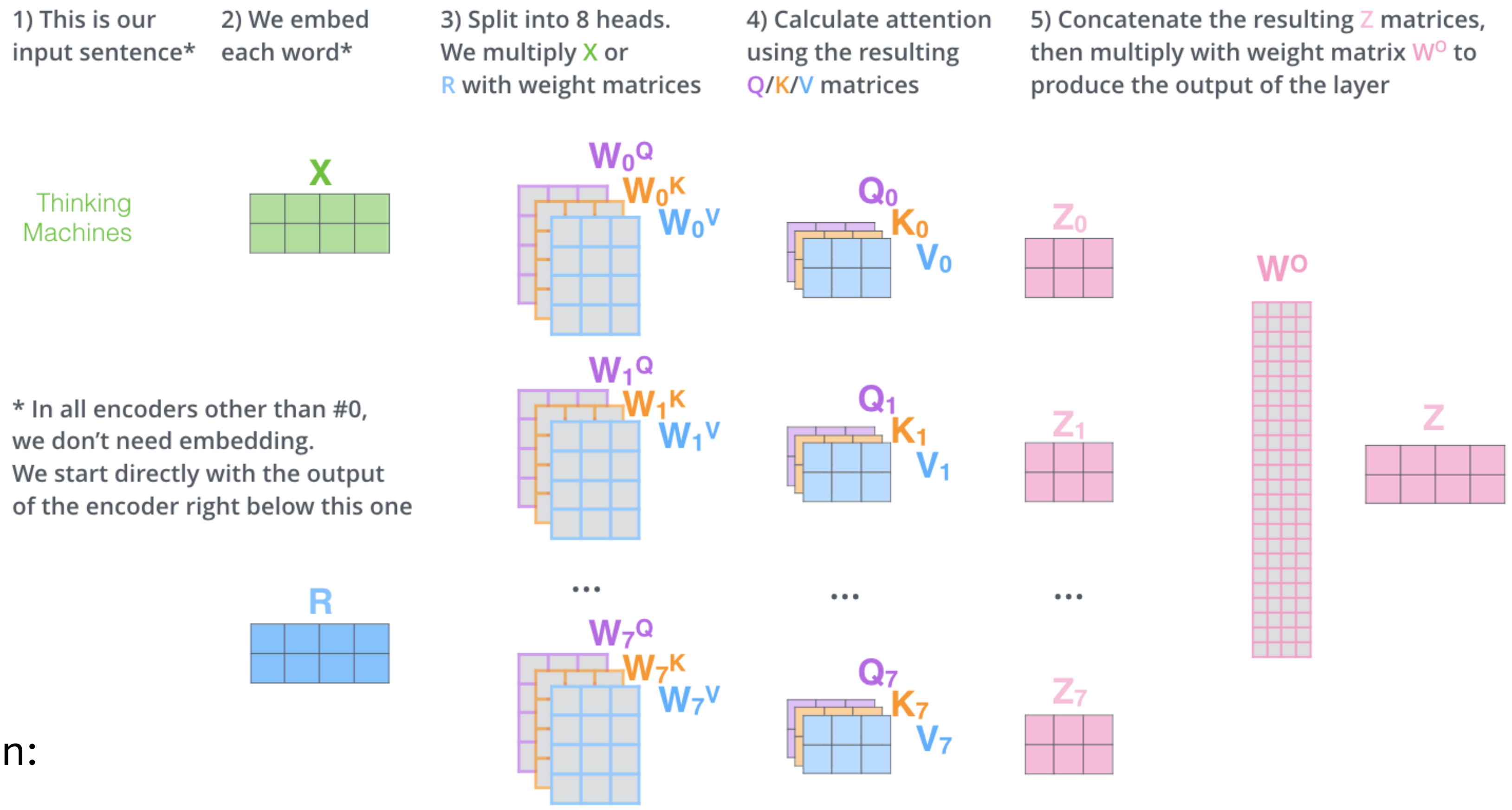


  - Every layer read in info from the residual stream, and then writes to it.
- A “deeply linear structure”
  - Every layer performs an arbitrary linear transformation (xW + b)
  - A rather unique feature of transformers (e.g. ResNets is not linear)
- “Virtual weights”
  - Implicit “virtual weights” directly connecting any pair of layers

Virtual Weights



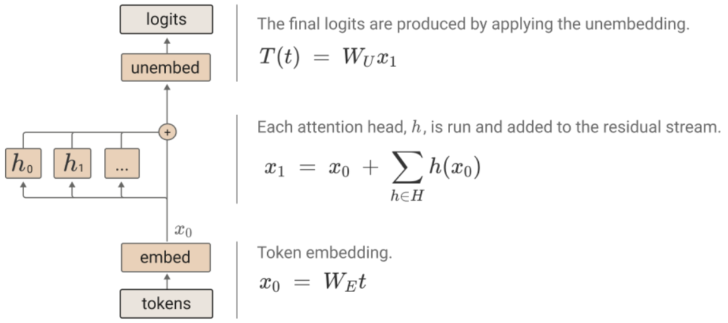
Attention Heads are Independent and Additive



Different notation:
r is z

$$W_O^H \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \left[W_O^{h_1}, W_O^{h_2}, \dots \right] \cdot \begin{bmatrix} r^{h_1} \\ r^{h_2} \\ \dots \end{bmatrix} = \sum_i W_O^{h_i} r^{h_i}$$

Understanding Attention Heads



- Rewrite the attention algorithm as:

$$h(x) = (\mathrm{Id} \otimes W_O) \cdot (A \otimes \mathrm{Id}) \cdot (\mathrm{Id} \otimes W_V) \cdot x$$

Project result vectors out for each token $(h(x)_i = W_O r_i)$

Mix value vectors *across* tokens to compute result vectors $(r_i = \sum_j A_{i,j} v_j)$

Compute value vector for each token $(v_i = W_V x_i)$

$$h(x) = (A \otimes W_O W_V) \cdot x$$

A mixes across tokens while $W_O W_V$ acts on each vector independently.

Which token's information is moved from and to.

which information is read from the source token and how it is written to the destination token.

Understanding Attention Heads

- We have four weights:
- W_K, W_Q, W_V, W_O

$$h(x) = \underbrace{(A \otimes W_O W_V)}_{} \cdot x$$

A mixes across tokens while $W_O W_V$ acts on each vector independently.

$$A = \mathrm{softmax}(x^T W_Q^T W_K x)$$

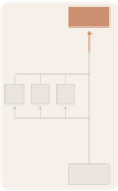
- W_K, W_Q always operate together
- W_O, W_V always operate together

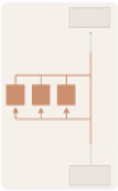
Because they always operate together, we can define variables combining them together:

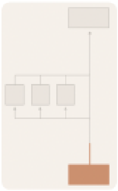
- W_{QK}
- W_{OV}

Let’s Rewrite the Attention Computation (Again – for a 1-Layer Transformer)

$$T = \underbrace{\mathrm{Id} \otimes W_U}_{} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{}$$

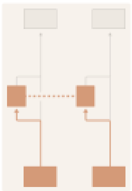

The **token unembedding** maps residual stream vectors to logits.


The **attention layer** has multiple heads. The result of each is added into the residual stream.


The **token embedding** maps tokens to residual stream vectors.

where $A^h = \underbrace{\mathrm{softmax}^*}_{}\left(\underbrace{t^T \cdot W_E^T W_{QK}^h W_E \cdot t}_{} \right)$

Softmax with autoregressive masking


Attention pattern logits are produced by multiplying pairs of tokens through different sides of W_{QK}^h.

Recall:

q = W_q(W_e(t))

k = W_k(W_e(t))

Big Table is All You Need

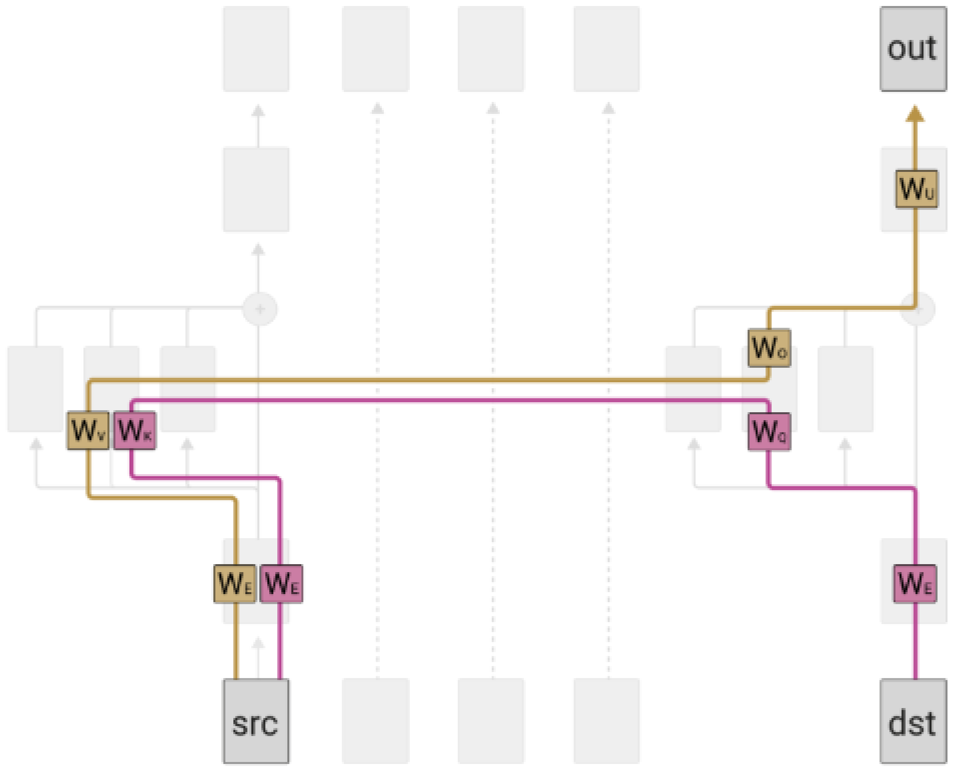
Two big look-up tables!

For each attention head h we have a term $A^h \otimes (W_U W_{OV}^h W_E)$ where $A^h = \text{softmax}\left(t^T \cdot W_E^T W_{QK}^h W_E \cdot t\right)$. How can we map these terms to model behavior? And while we're at it, why do we get these particular products of matrices on our equations?

The key thing to notice is that these terms consist of two separable operations, which are at their heart two $[n_{\text{vocab}},\ n_{\text{vocab}}]$ matrices:

- $W_E^T W_{QK}^h W_E$ — We call this matrix the "query-key (QK) circuit." It provides the attention score for every query and key token. That is, each entry describes how much a given query token "wants" to attend to a given key token.
- $W_U W_{OV}^h W_E$ — We call this matrix the "Output-Value (OV) circuit." It describes how a given token will affect the output logits if attended to.

Single Layer Transformer: QK Circuit & OV Circuit



The OV ("output-value") circuit determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

n_vocab x n_vocab matrix

The QK ("query-key") circuit controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

n_vocab x n_vocab matrix

What kinds of things does pretraining teach?

- There's increasing evidence that pretrained models learn a wide variety of things about the statistical properties of language.
  - University of Toronto is located in __________, Ontario. [Trivia, Facts]
  - I put ___ fork down on the table. [syntax]
  - The woman walked across the street, checking for traffic over ___ shoulder. [coreference/anaphora]
  - I went to the ocean to see the fish, turtles, seals, and _____. [lexical semantics/topic]
  - Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ___. [sentiment]
  - Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the ______. [some reasoning – this is harder]
  - I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, ____ [some basic arithmetic; they don't learn the Fibonacci sequence]
- Models also learn – and can exacerbate racism, sexism, all manner of bad biases.

Token Association Statistics Can't Explain In-context Learning

- If pretraining is just memorizing token association statistics…
- Then why can the models do in-context learning?
- Recall A2 T2:

```
model answer:
a b c -> c; d e f -> f; g h i ->
 i
model answer:
a b c -> c; d e f -> f; x y z ->
 z
model answer:
a b c -> c; d e f -> f; m r s ->
 s
model answer:
a b c -> c; d e f -> f; v u f ->
 f
```

Induction Head

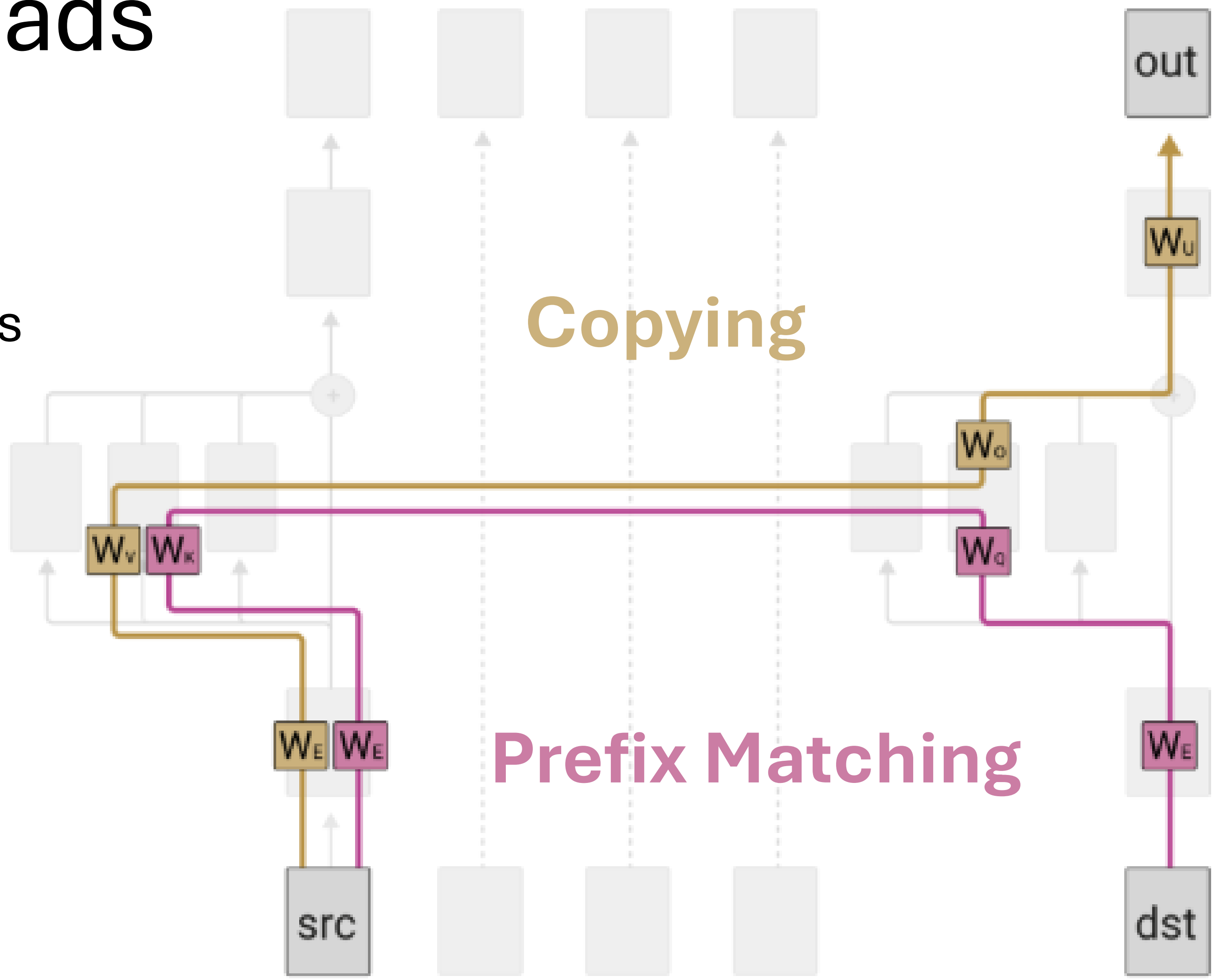
- There is something more “mechanistic” going on in the model.
- One thing that the model is doing:

```
test category 40 ids node color some random stuff over here category 40 ids node
Model output: color

test category 40 ids node less some random stuff over here category 40 ids node
Model output: less

test category 40 ids node ideas some random stuff over here category 40 ids node
Model output: ideas

test category 40 ids node sleep some random stuff over here category 40 ids node
Model output: sleep

test category 40 ids node furiously some random stuff over here category 40 ids node
Model output: furiously
```

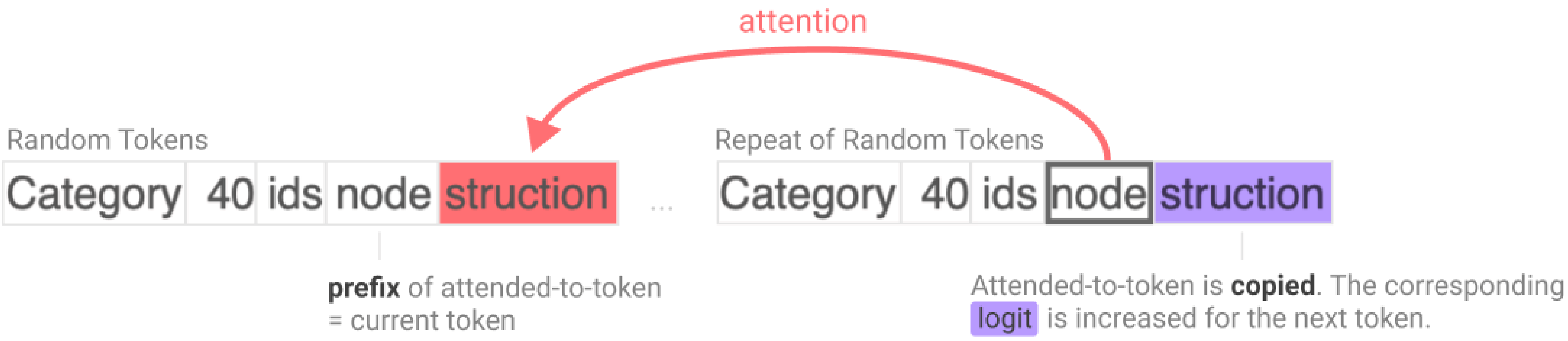
Induction Heads

There exists attention heads that implement this simple algorithm:

- **Prefix matching (QK):**
  - Attend to tokens preceded by this token.
- **Copying (OV)**
  - Increase logits of attended tokens.



Induction Head

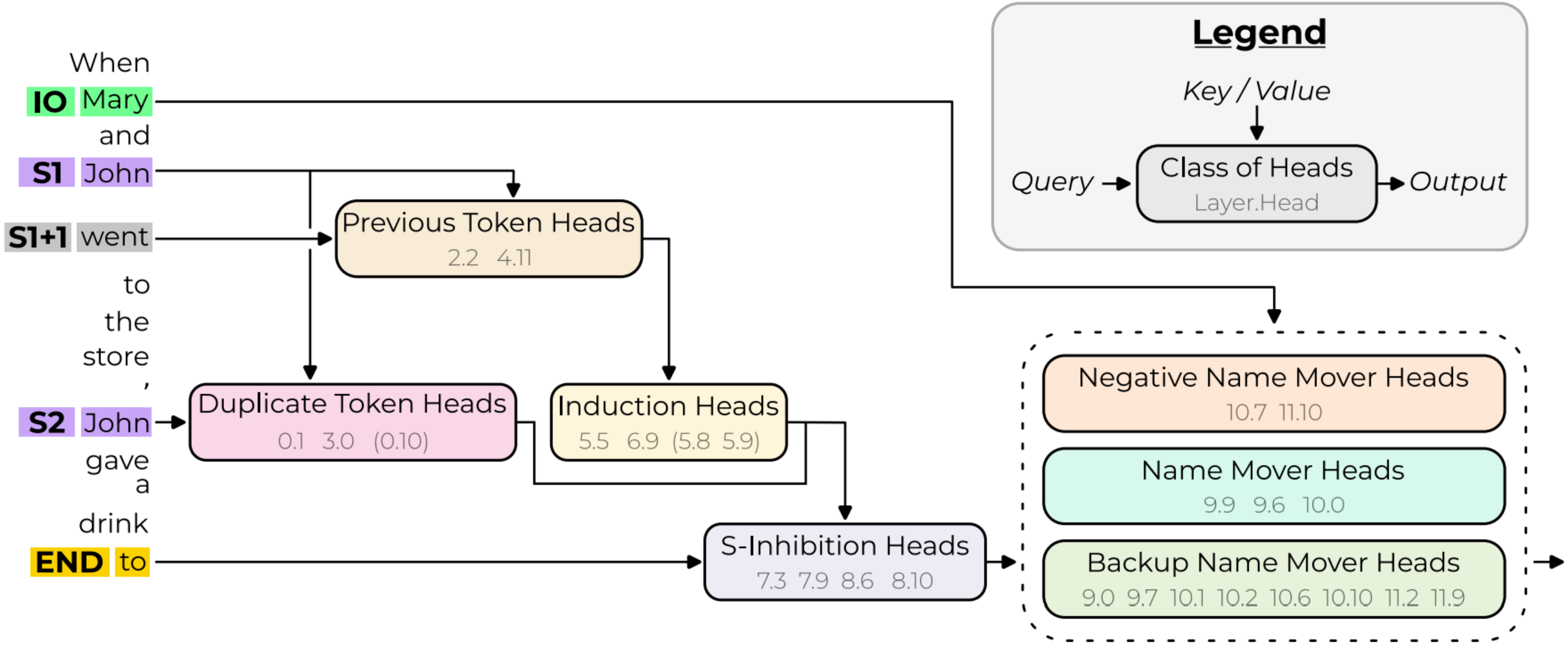


- NOT statistical associations!
- Something “mechanistic” learned in the model.

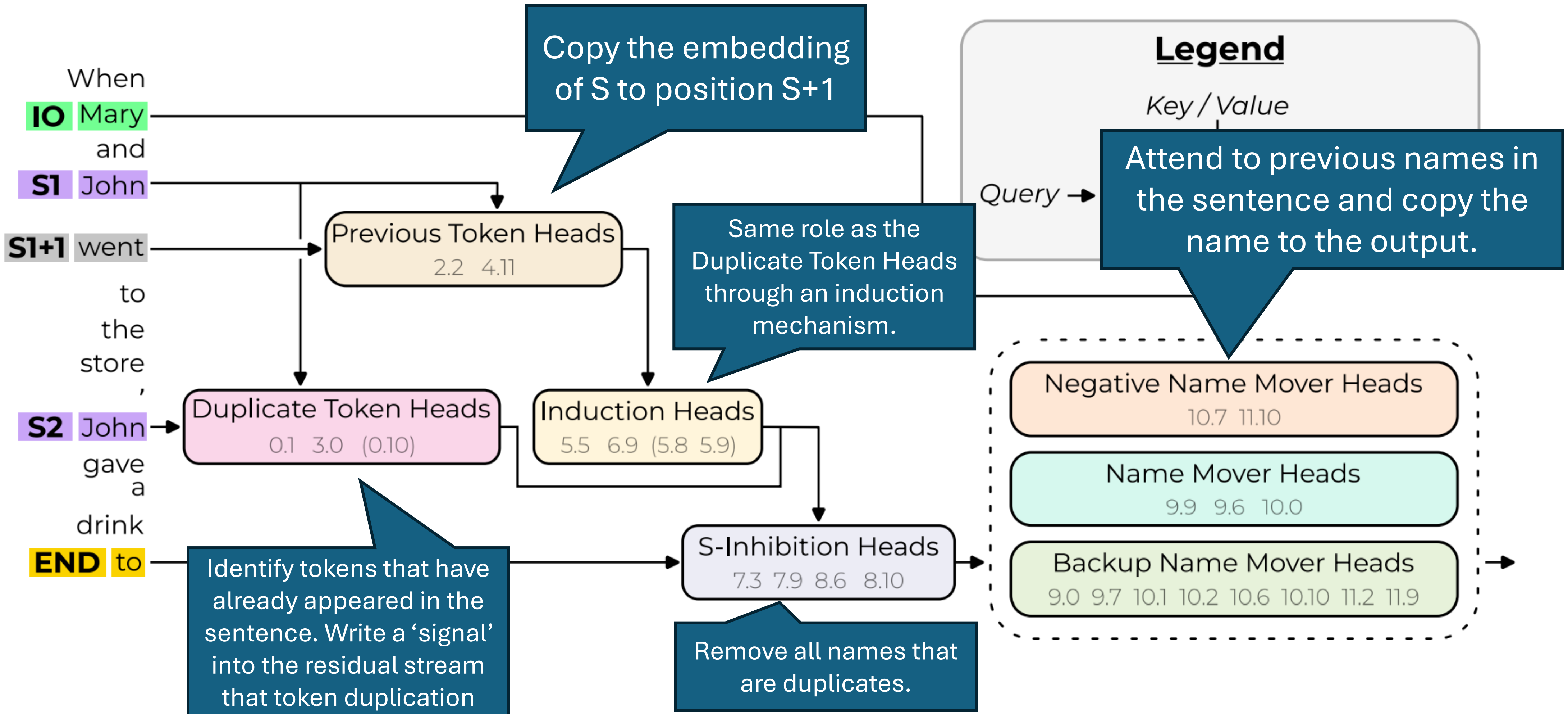
More Complicated Algorithms?

- Indirect Object Identification (IOI) Task
- When Mary and John went to the store, John gave a drink to
    - **Mary**
- Wang et al. (2023): GPT-2 implement this algorithm:
    1. Identify all previous names in the sentence (Mary, John, John).
    2. Remove all names that are duplicates (in the example above: John).
    3. Output the remaining name.

“The IOI Circuit” in GPT-2



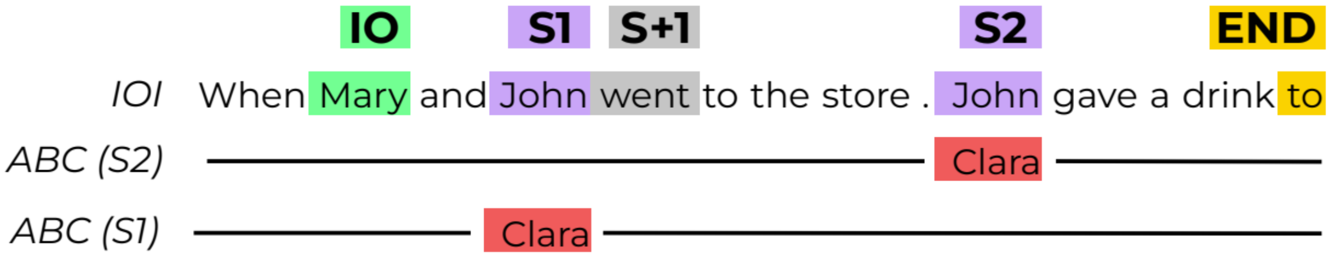
"The IOI Circuit" in GPT-2

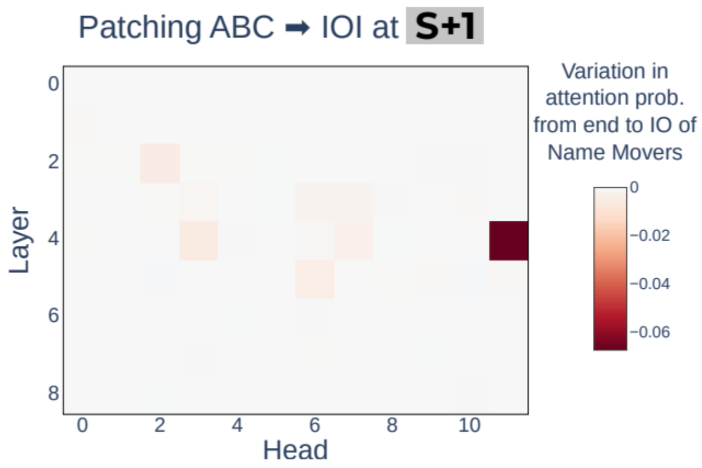


Attend to the S2 token and write to bias the query of the Name Mover Heads against both S1 and S2 tokens

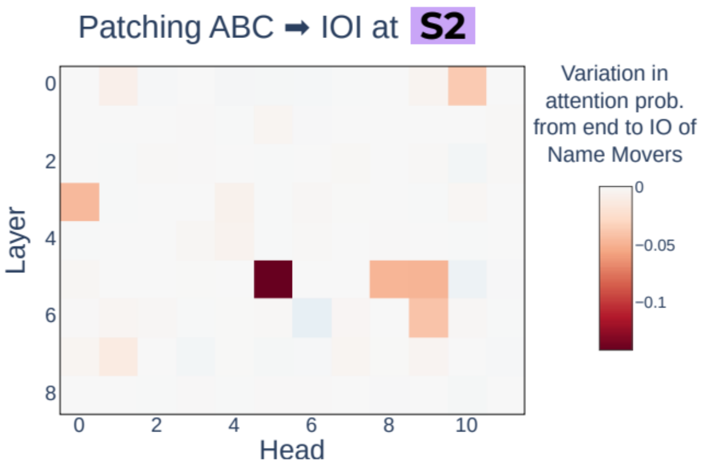
Discovering the Circuit

- Idea: patching the residual stream with a new value

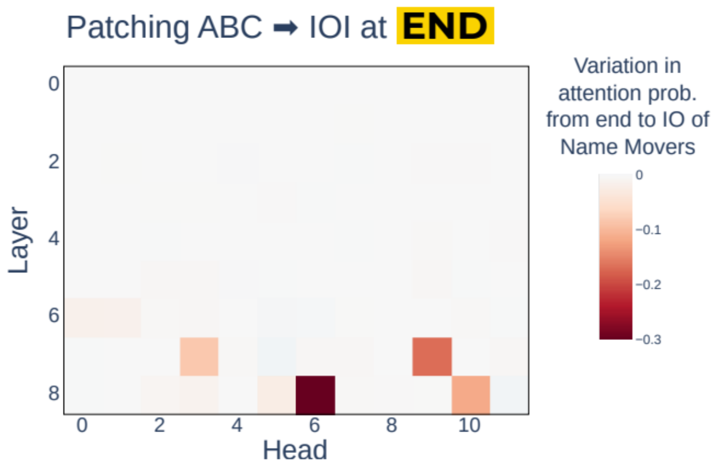




Previous Token Heads



Induction Heads



S-Inhibition Heads

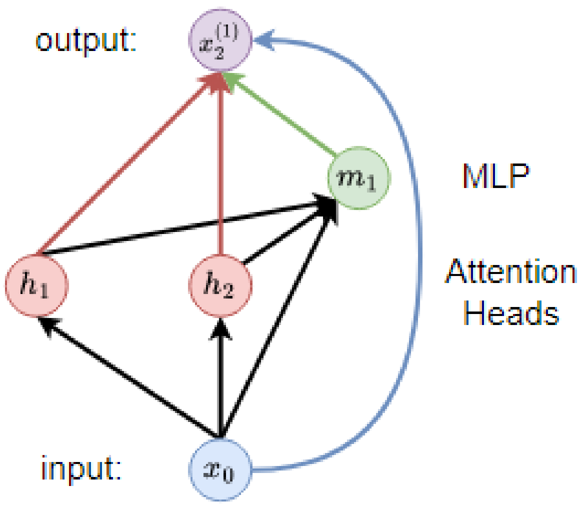
Evaluation

- Can the model can complete the task with only the circuit?
- Logits difference:
  - |F(full model) − F(circuit)| = 0.2
  - Or, only 6% of F(model) = 3.55
- Mean Ablation: replacing activations outside the circuit with their average activation value across some reference distribution.
- A lot of information has been maintained!

Circuit Discovery

- Computation Graph

$$x_1^{(1)} = x_0 + h_1(x_0) + h_2(x_0)$$
$$x_2^{(1)} = x_1^{(1)} + f_1(x_1^{(1)})$$
$$= {\color{blue}x_0} + {\color{red}h_1(x_0)} + {\color{red}h_2(x_0)} + {\color{green}f_1(x_0 + h_1(x_0) + h_2(x_0))}.$$



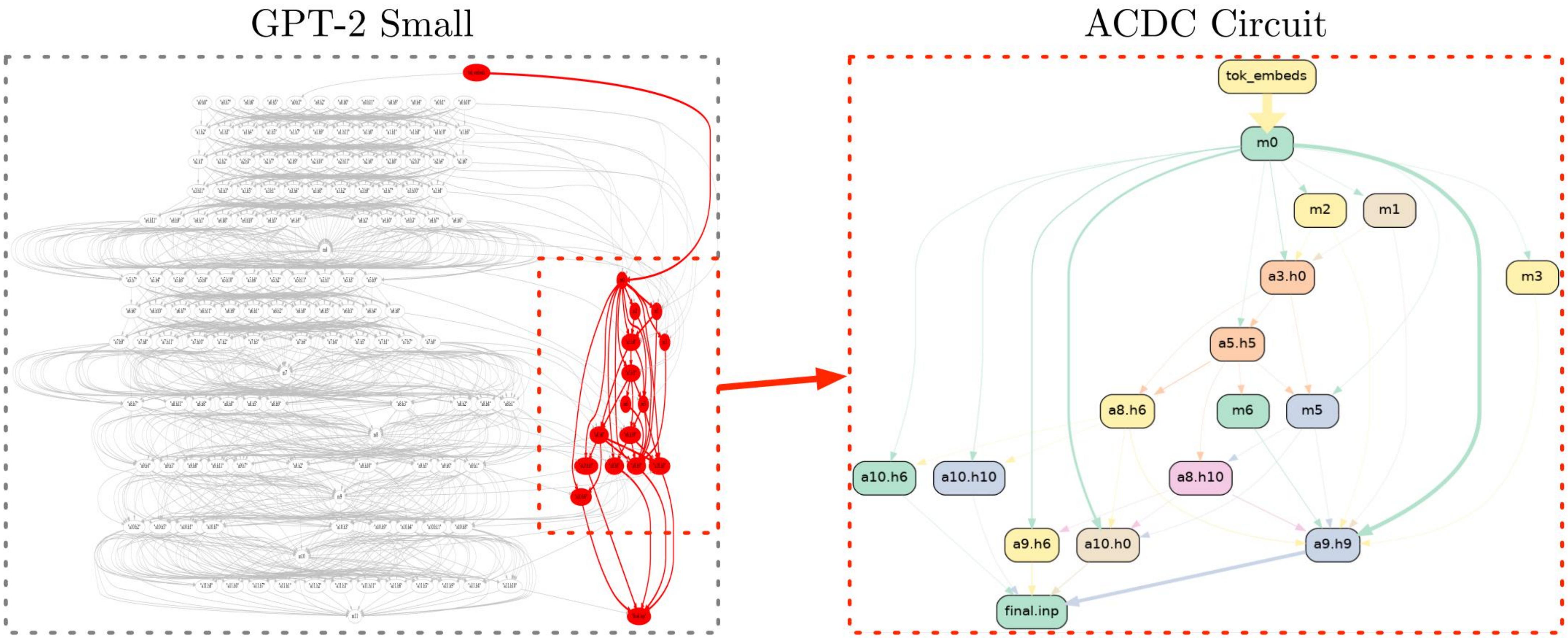
Connection Edge (->):

- Information flow from one component to another component in the residual stream.

Node:

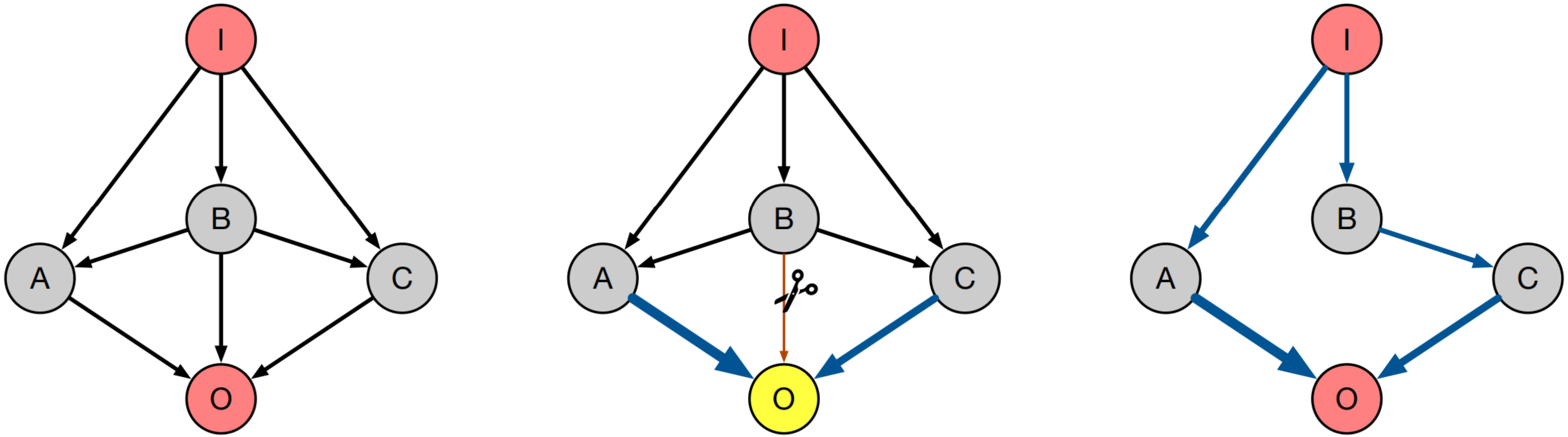
- Transformer components.
- Attention head, MLP, input (emb) and output (umbed).

Automated Circuit Discovery



Conmy et al. (2023) Towards Automated Circuit Discovery for Mechanistic Interpretability.

ACDC: Automatic Circuit DisCovery



(a) Choose computational graph, task, and threshold τ.

(b) At each head, prune unimportant connections.

(c) Recurse until the full circuit is recovered.

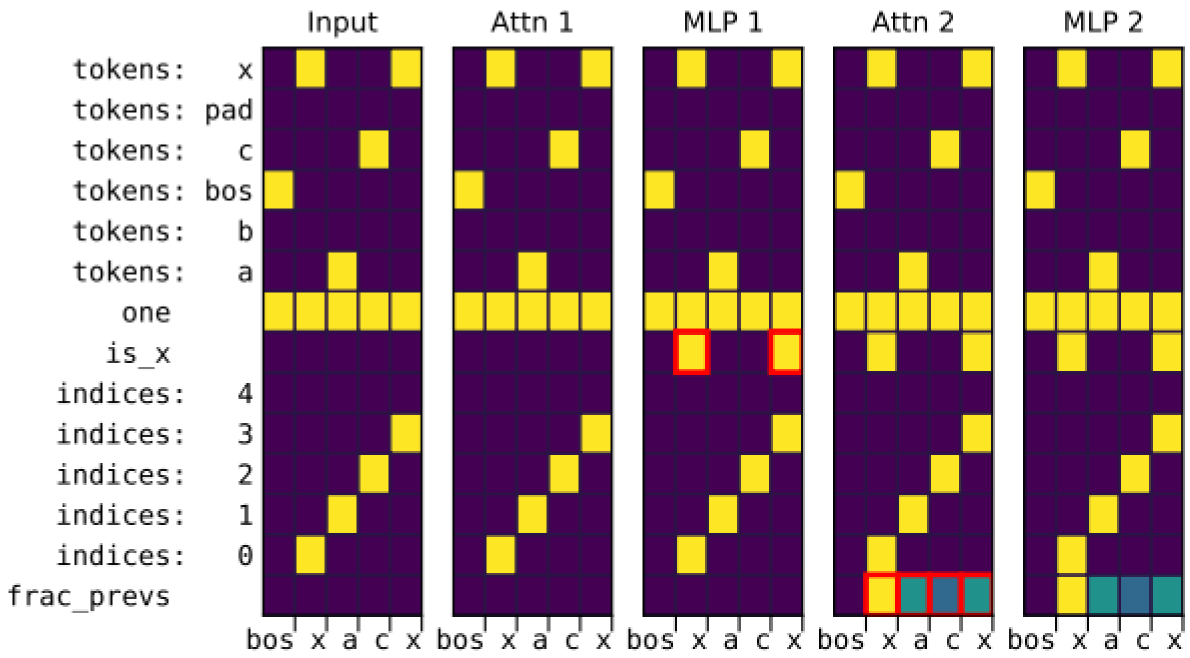
Is the Explanation Correct?

- If we have a hard-coded transformer model, can these interpretation methods discover it?

- WHAT? Hard-coded transformer model?

Tracr: Code to Transformer Compiler

```
is_x = (tokens == "x")
prevs = select(indices, indices, <=)
frac_prevs = aggregate(prevs, is_x)
```



Lindner et al. (2023) Tracr: Compiled Transformers as a Laboratory for Interpretability.

Conclusion

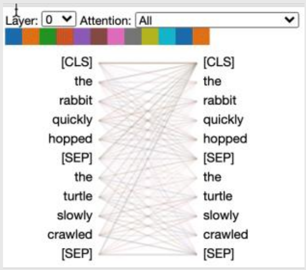
This Segment

“LM are linguistic subjects — sequence probabilities are reliable grammaticality judgements.”

$P(\text{grammatical}) > P(\text{ungrammatical})$

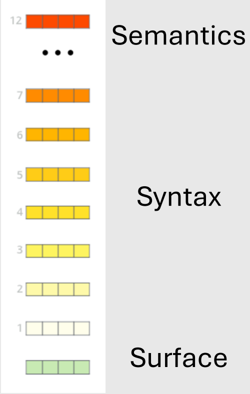
Language acquisition, nature of grammar...

“Understand how transformers word through interpreting attention patterns.”



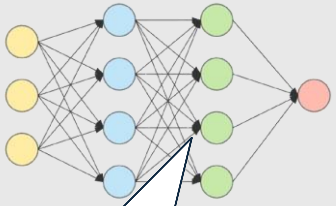
Some kind of syntactic structure inside?

“BERT Rediscovered the Classical NLP Pipeline.”



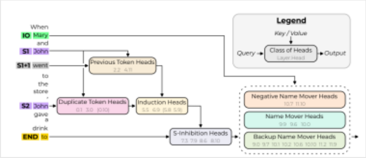
“Knowledge are located within the MLP neurons.”

Transformer MLP weights:



The capital of Canada is Ottawa.

“Information flow is more important!”



1. LM as a whole

2. Attention Patterns

3. Layer Level

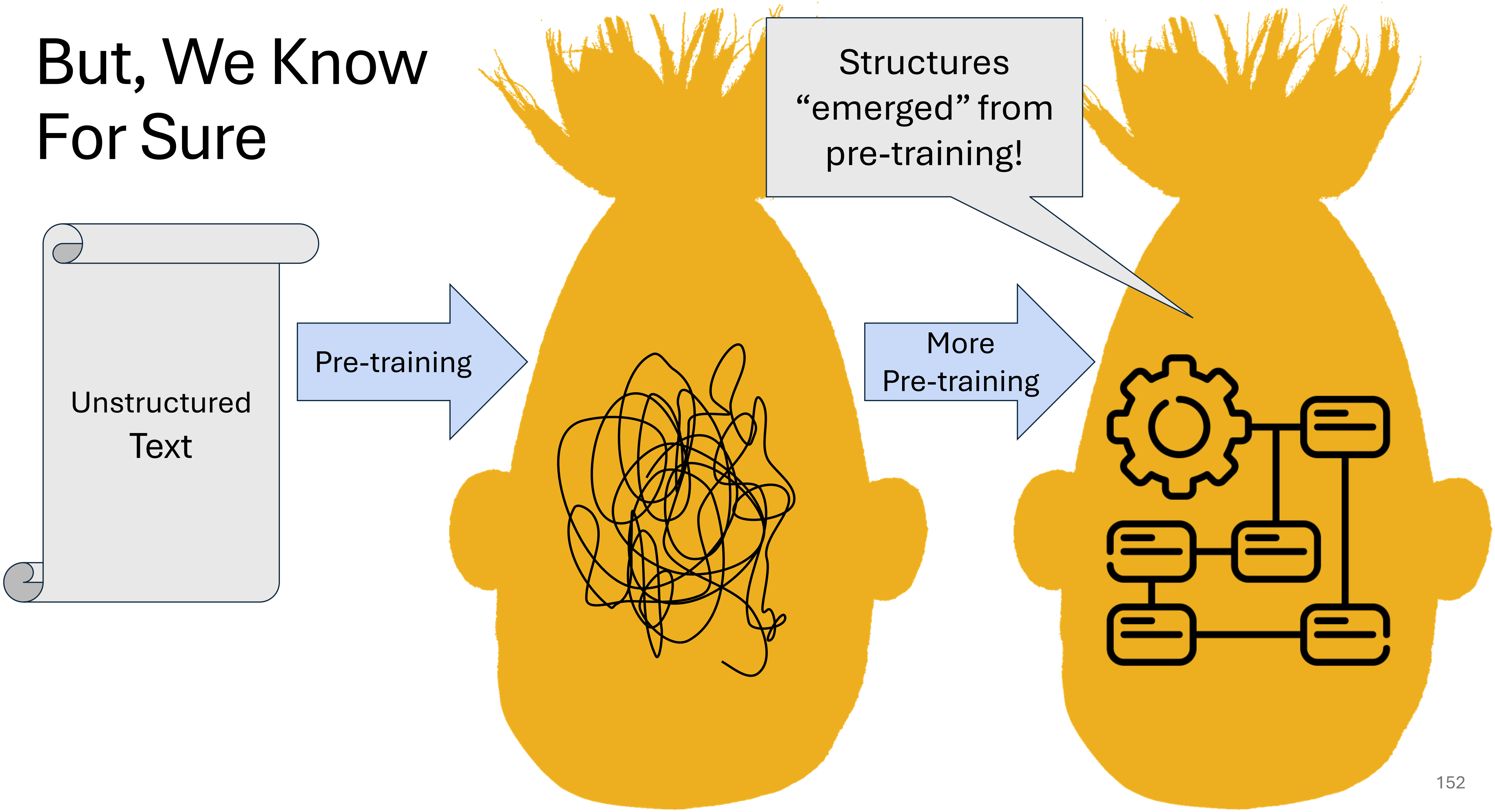
4. Neuron Level

5. Circuit Discovery

Limitations of Current Interpretation Work

- Restricted to very simple tasks.
  - IOI, basic syntactic tasks, simple synthetic tasks…
- Limited model size.
  - Induction head work: 2-layer transformer.
  - Most circuit discovery work: GPT-2 base.
- ONLY ONE TOKEN.
  - We studied everything on generating **one token**.
  - The behaviour is much more complicated and unstable when we consider more than one token.
- ……

But, We Know For Sure



Takeaway Points

- Memorizing the Implementation of the Transformer Architecture should be EASY now!


```python
class Transformer(nn.Module):

    def forward(self, input):
        residual = self.embed(input) # Embedding layer

        for i, block in self.blocks:

            residual = block(residual)

        logits = self.unembed(residual)

        return logits

class TransformerBlock(nn.Module):

    def forward(self, resid_pre):

        attn_in = split_attention_head(resid_pre)
        attn_out = self.attn(self.ln1(attn_in))

        resid_mid = resid_pre + attn_out

        mlp_in = resid_mid
        mlp_out = self.mlp(self.ln2(mlp_in))

        resid_post = resid_mid + mlp_out

        return resid_post
```

```python
def attention_head(x):
    '''TransformerLens Notation'''
    q = self.W_Q(x)
    k = self.W_K(x)
    v = self.W_V(x)

    attn_scores = q @ k / self.attn_scale

    pattern = F.softmax(attn_scores, dim=-1)

    z = pattern @ v

    attn_out = self.W_O @ z + self.B_O

    return attn_out
```

Takeaway Points

- Memorizing the Implementation of the Transformer Architecture should be EASY now!
- Knowing where we at in LM interpretation research.
- Coolest thing one can do without the need of 100 GPUs.
- See through the hype and understand the trend.