

# Parametric Knowledge in Language Models

Siyi Gao, Alexander Yao

# Parametric Knowledge

Contextual (Nonparametric) Knowledge – Information the LLM attains through the prompt

Parametric Knowledge – Information the LLM remembers from pretraining

**Instruction:** Use the given info and ***your own knowledge*** ...  
**Question:** What is the occupation of Michael Jordan?  
**Contextual Knowledge:** Lionel Messi is an Argentina football player...

Cheng et al., 2024

# Example

**Context:** The Boeing 717 is a jet airliner. The Boeing 717 has two engines. The Boeing 717 is a twinjet. The Boeing 717 has two columns of seats.

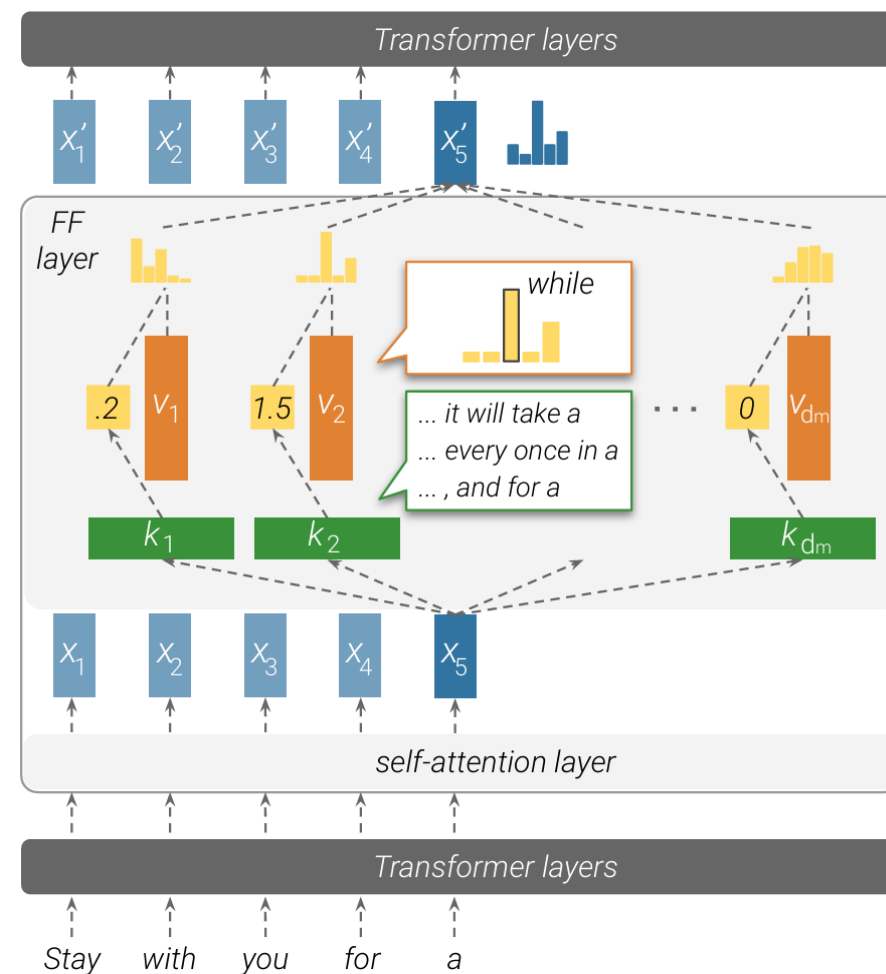
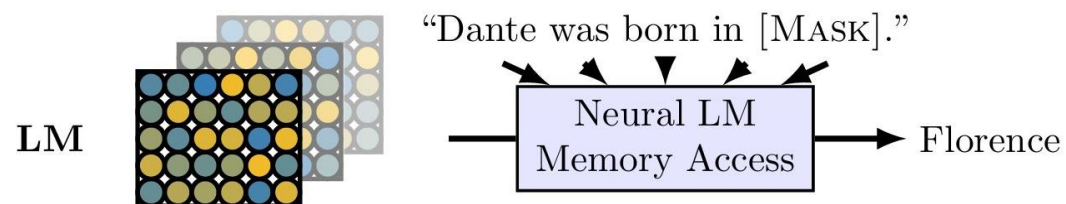
**Question:** With this information, tell me about Boeing 717.

**GPT-4o Response:** The Boeing 717 is a jet airliner equipped with two engines, which classifies it as a twinjet. Inside the aircraft, there are two columns of seats for passengers. The aircraft can accommodate up to 117 passengers.

# Parametric Knowledge Storage

Attention Layer – Contextualizes input

Feedforward Layer – Stores and compiles memories



# Language Models as Knowledge Bases?

**Fabio Petroni<sup>1</sup> Tim Rocktäschel<sup>1,2</sup> Patrick Lewis<sup>1,2</sup> Anton Bakhtin<sup>1</sup>  
Yuxiang Wu<sup>1,2</sup> Alexander H. Miller<sup>1</sup> Sebastian Riedel<sup>1,2</sup>**

<sup>1</sup>Facebook AI Research

<sup>2</sup>University College London

# Motivations

- Benefits of Language models as a knowledge base:
  1. Require no schema engineering
  2. Allow practitioners to query about an open class of relations
  3. Easy to extend to more data
  4. Require no human supervision to train

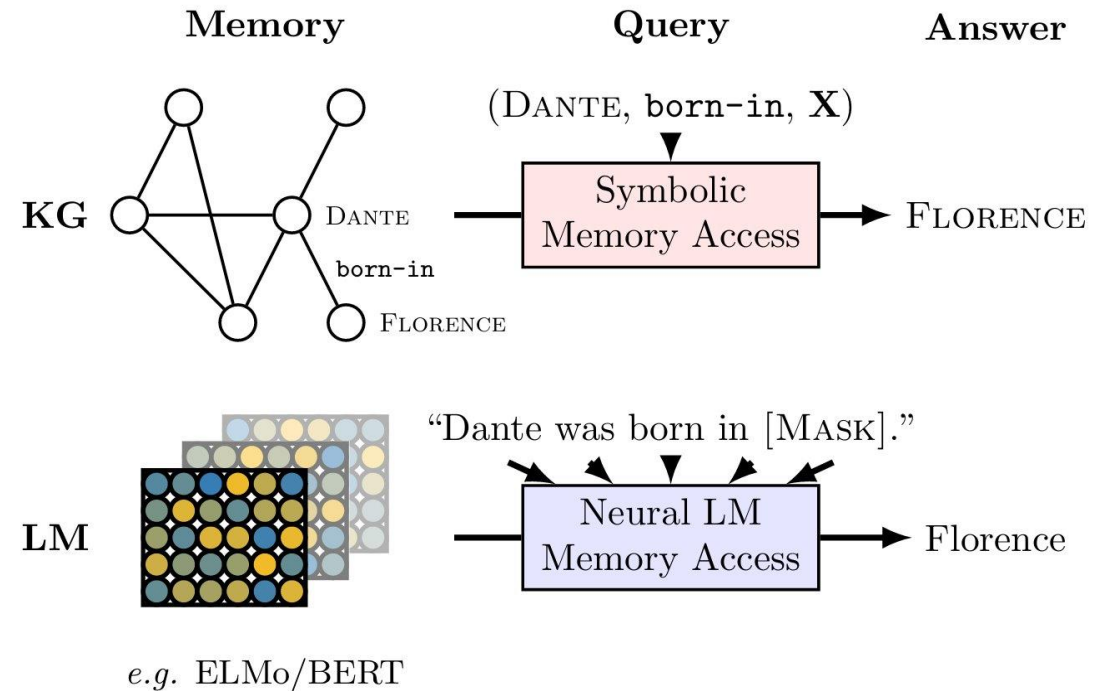


Figure 1: Querying knowledge bases (KB) and language models (LM) for factual knowledge.

# Questions

We are interested in the **relational knowledge** already present in pretrained off-the-shelf language models such as ELMo (a pretrained model before BERT based on LSTMs) and BERT.

- **How much** relational knowledge do they store?
- How does this **differ for different types of knowledge** such as facts about entities, common sense, and general question answering?
- How does their performance **without fine-tuning** compare to symbolic knowledge bases automatically extracted from text?

# Findings

1. **Without fine-tuning, BERT** contains relational knowledge **competitive** with traditional NLP methods that have some access to oracle knowledge.
2. BERT also does remarkably well on **open-domain question answering** against a supervised baseline.
3. Certain types of factual knowledge are learned much more readily than others by standard language model pretraining approaches.



# LAMA probe (LAnguage MModel AAnalysis)

- **Purpose:** Test the factual and commonsense knowledge in language models.
- **Format:** cloze completion
- **Assumption:** models which rank ground truth tokens high for these cloze statements have more factual knowledge.

Query	Answer	Generation
Francesco Bartolomeo Conti was born in ____.	Florence	Rome [-1.8] , <b>Florence</b> [-1.8] , Naples [-1.9] , Milan [-2.4] , Bologna [-2.5]
Adolphe Adam died in ____.	Paris	<b>Paris</b> [-0.5] , London [-3.5] , Vienna [-3.6] , Berlin [-3.8] , Brussels [-4.0]
English bulldog is a subclass of ____.	dog	dogs [-0.3] , breeds [-2.2] , <b>dog</b> [-2.4] , cattle [-4.3] , sheep [-4.5]
The official language of Mauritius is ____.	English	<b>English</b> [-0.6] , French [-0.9] , Arabic [-6.2] , Tamil [-6.7] , Malayalam [-7.0]

An Example of LAMA

# LAMA probe (LAnguage MModel AAnalysis)

- Key components:
  - Knowledge sources
  - Models
  - Metrics
  - Baselines

# Models and Knowledge Sources

## Models:

Model	Base Model	#Parameters	Training Corpus	Corpus Size
fairseq-fconv (Dauphin et al., 2017)	ConvNet	324M	WikiText-103	103M Words
Transformer-XL (large) (Dai et al., 2019)	Transformer	257M	WikiText-103	103M Words
ELMo (original) (Peters et al., 2018a)	BiLSTM	93.6M	Google Billion Word	800M Words
ELMo 5.5B (Peters et al., 2018a)	BiLSTM	93.6M	Wikipedia (en) & WMT 2008-2012	5.5B Words
BERT (base) (Devlin et al., 2018a)	Transformer	110M	Wikipedia (en) & BookCorpus	3.3B Words
BERT (large) (Devlin et al., 2018a)	Transformer	340M	Wikipedia (en) & BookCorpus	3.3B Words

Table 1: Language models considered in this study.

## Knowledge sources:

Dataset	Contents	Example
Google-RE	specific relations: “place of birth”, “date of birth” and “place of death”	[S] was born in [O]
T-REx	facts from Wikipedia with more relation types: 1-1, N-1, N-M	English bulldog is a subclass of ____.
ConceptNet	commonsense knowledge	Birds have _____. [feathers]
SQuAD	open-domain questions	The theory of relativity was developed by ____.

# Metrics: Mean precision at k ( $P@k$ )

- For a given fact, this value is 1 if the object is ranked among the top k results, and 0 otherwise.
- Example: Suppose we test (Paris, capital, ?)
  - Model ranks the top 5 predictions:
    - Berlin ✗
    - Rome ✗
    - France ✓
    - Madrid ✗
    - London ✗
  - If  $k = 3$ , the correct answer ("France") appears in the top 3, so  $P@3 = 1$ .
  - If  $k = 2$ , the correct answer is not in the top 2, so  $P@2 = 0$ .

# Baselines

- **Freq:**

- It selects the most common object for a given relation.
- Example: For 'birth-place' relation, if 'Paris' is the most frequent answer, it is always predicted for all questions.

- **RE:**

- The pretrained Relation Extraction (RE) model (Sorokin and Gurevych (2017)).
- Two format:
  - **REn (Naïve Entity Linking):** Cannot link "Albert Einstein" to "Einstein".
  - **REo (Oracle Entity Linking):** Can correctly link "Albert Einstein" to "Einstein".

- **DrQA:**

- Open-Domain Question Answering System (Chen et al., 2017).
- In this paper, the final predictions are restricted to single-token answers.

# Results

Corpus	Relation	Statistics		Baselines		KB		LM					
		#Facts	#Rel	Freq	DrQA	RE <sub>n</sub>	RE <sub>o</sub>	Fs	Txl	Eb	E5B	Bb	Bl
Google-RE	birth-place	2937	1	4.6	-	3.5	13.8	4.4	2.7	5.5	7.5	14.9	<b>16.1</b>
	birth-date	1825	1	1.9	-	0.0	<b>1.9</b>	0.3	1.1	0.1	0.1	1.5	1.4
	death-place	765	1	6.8	-	0.1	7.2	3.0	0.9	0.3	1.3	13.1	<b>14.0</b>
	Total	5527	3	4.4	-	1.2	7.6	2.6	1.6	2.0	3.0	9.8	10.5
T-REx	1-1	937	2	1.78	-	0.6	10.0	17.0	36.5	10.1	13.1	68.0	74.5
	<i>N</i> -1	20006	23	23.85	-	5.4	33.8	6.1	18.0	3.6	6.5	32.4	34.2
	<i>N</i> - <i>M</i>	13096	16	21.95	-	7.7	36.7	12.0	16.5	5.7	7.4	24.7	24.3
	Total	34039	41	22.03	-	6.1	33.8	8.9	18.3	4.7	7.1	31.1	32.3
ConceptNet	Total	11458	16	4.8	-	-	-	3.6	5.7	6.1	6.2	15.6	19.2
SQuAD	Total	305	-	-	37.5	-	-	3.6	3.9	1.6	4.3	14.1	17.4

Open-domain QA:  
P@10  
BERT-large: 57.1%  
DrQA: 63.5%

Table 2: Mean precision at one (P@1) for a frequency baseline (Freq), DrQA, a relation extraction with naïve entity linking (RE<sub>n</sub>), oracle entity linking (RE<sub>o</sub>), fairseq-fconv (Fs), Transformer-XL large (Txl), ELMo original (Eb), ELMo 5.5B (E5B), BERT-base (Bb) and BERT-large (Bl) across the set of evaluation corpora.

# Results

A downstream model could learn to make use of knowledge in the output representations of a language model even if the correct answer is not ranked first but high enough.

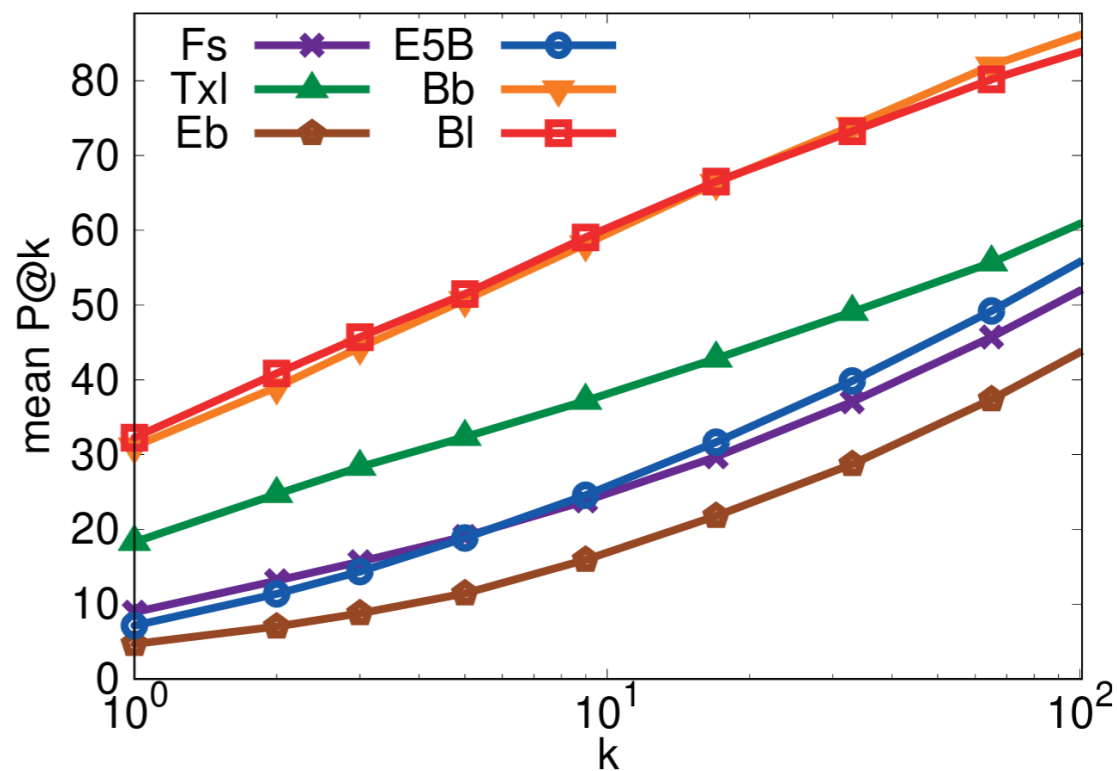


Figure 2: Mean P@k curve for T-REx varying k. Base-10 log scale for X axis.

# Results

- Test how the performance of a pretrained language model varies with different ways of querying for a particular fact.
- Robustness

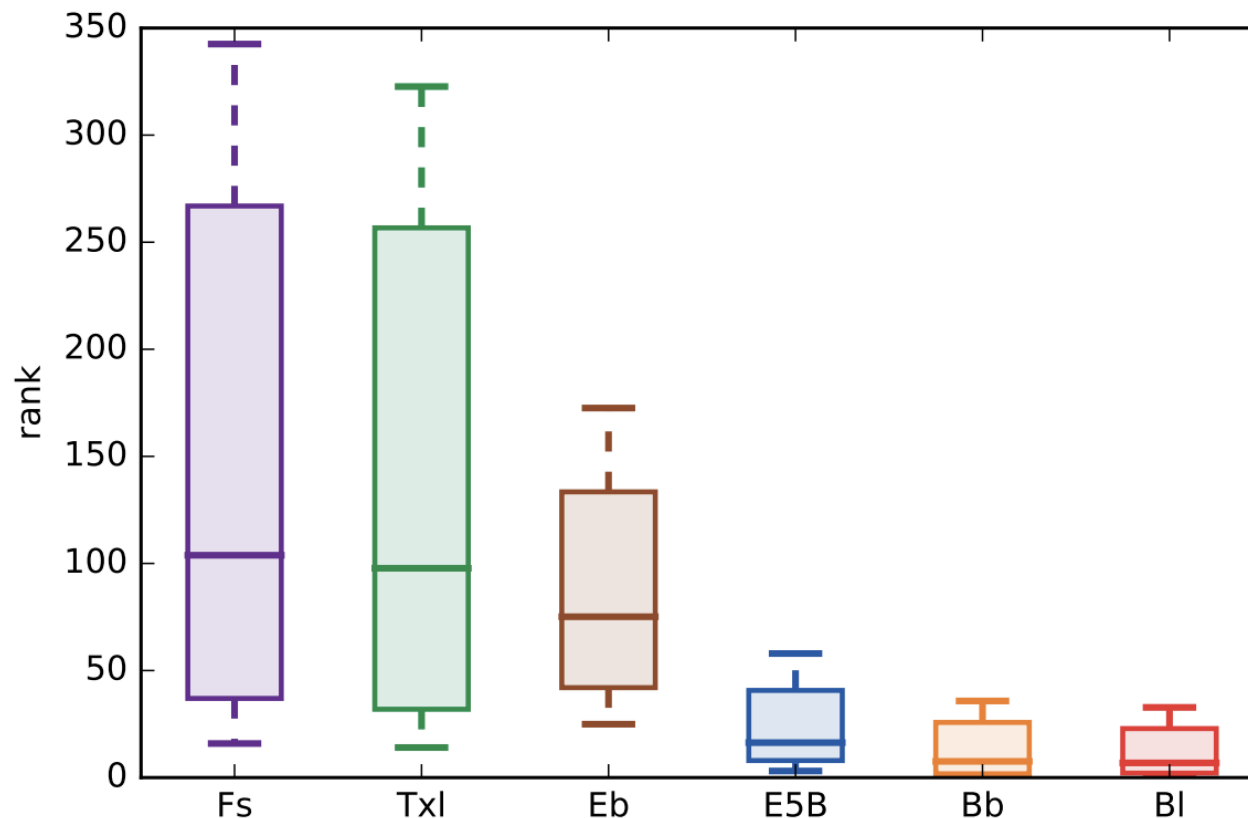



Figure 4: Average rank distribution for 10 different mentions of 100 random facts per relation in T-REx. ELMo 5.5B and both variants of BERT are least sensitive to the framing of the query but also are the most likely to have seen the query sentence during training.



# Recap & Limitations

## Recap:

- BERT-large 
- Factual knowledge can be recovered surprisingly well from pretrained language models.
- For some relations (particularly N-to-M relations) performance is very poor.

## Limitations:

- Unlike knowledge bases, language models are highly uninterpretable, making it hard to figure out when and why the model makes a factual mistake.

# Transformer Feed-Forward Layers Are Key-Value Memories

**Mor Geva**<sup>1,2</sup>     **Roei Schuster**<sup>1,3</sup>     **Jonathan Berant**<sup>1,2</sup>     **Omer Levy**<sup>1</sup>

<sup>1</sup>Blavatnik School of Computer Science, Tel-Aviv University

<sup>2</sup>Allen Institute for Artificial Intelligence

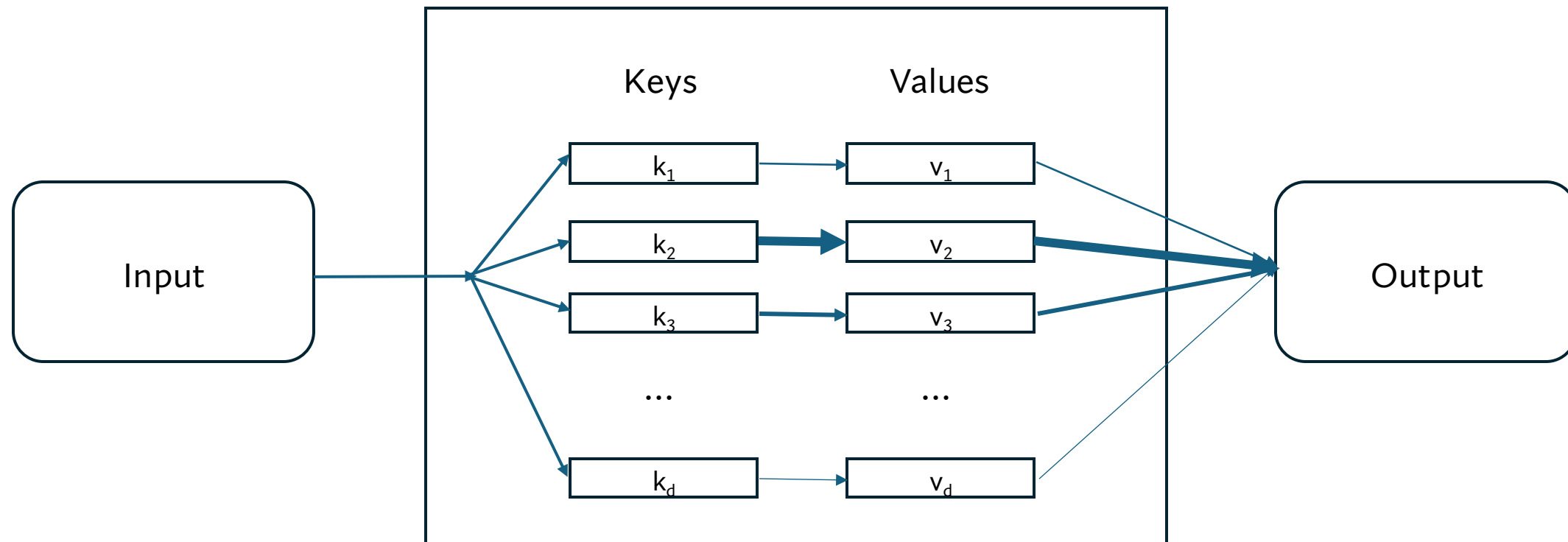
<sup>3</sup>Cornell Tech

Keys capture semantic patterns in the input prompt

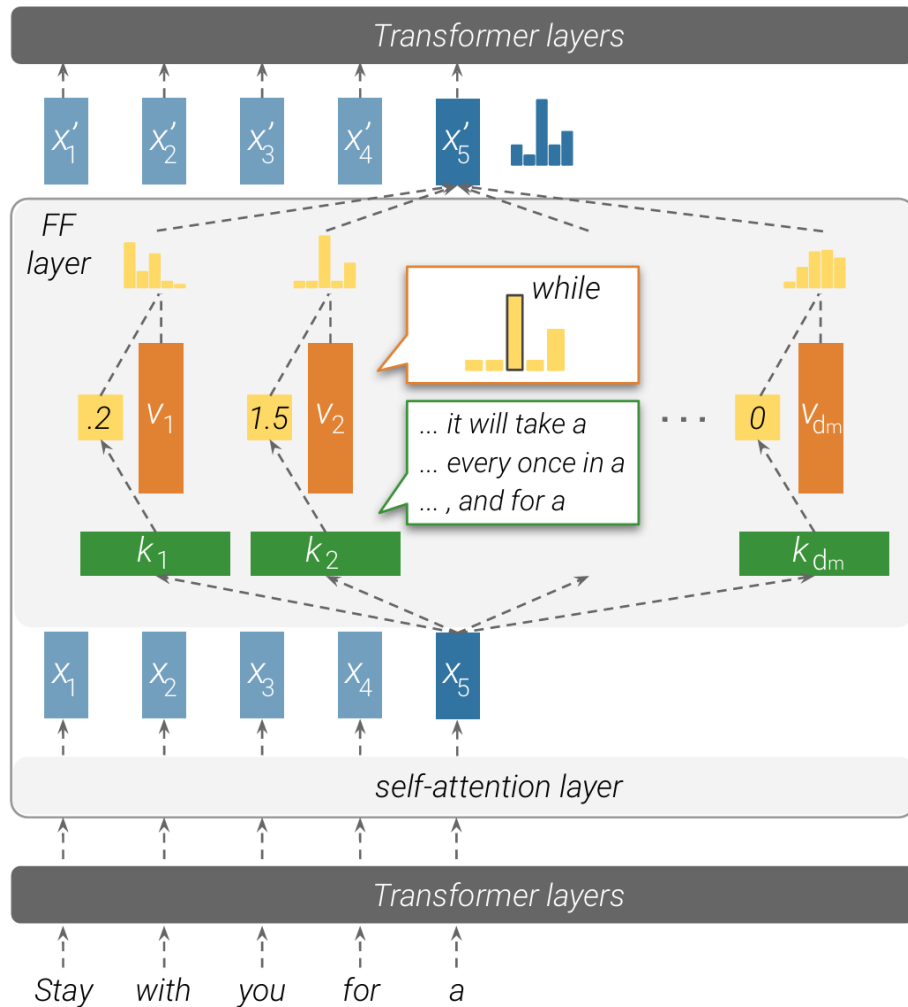
Values return a distribution of the vocabulary

Memories of the values are aggregated to produce a final distribution

# Neural (Key-Value) Memories



# Neural (Key-Value) Memories



Keys  $\mathbf{k}_i \in \mathbb{R}^d$

Key and Value Matrix

$$K \in \mathbb{R}^{d_m \times d} \quad V \in \mathbb{R}^{d_m \times d}$$

Input  $\mathbf{x} \in \mathbb{R}^d$

Memory Coefficient

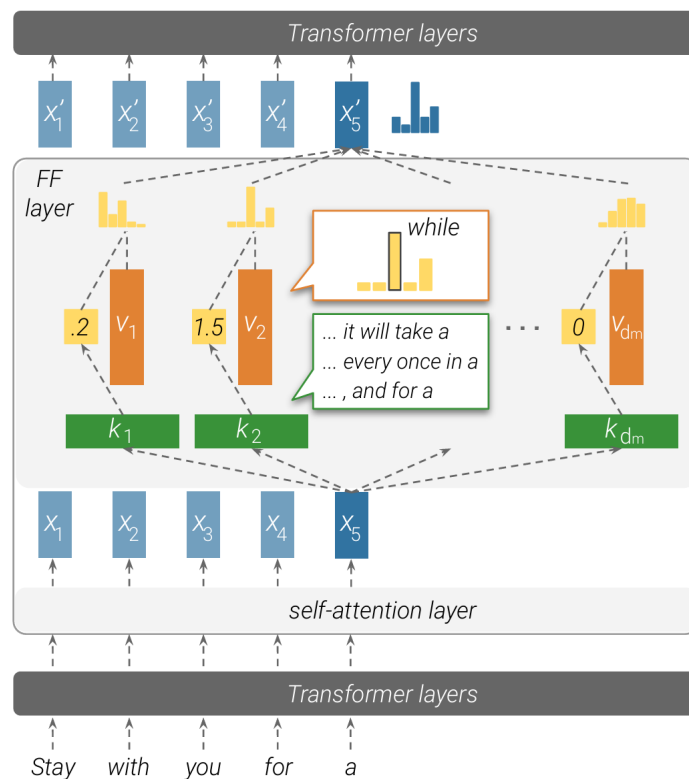
$$p(k_i | x) \propto \exp(\mathbf{x} \cdot \mathbf{k}_i)$$

Output

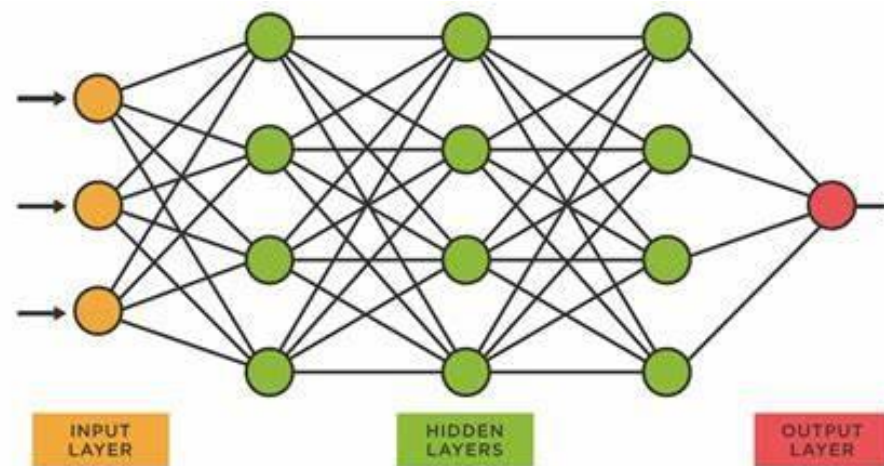
$$\text{MN}(\mathbf{x}) = \sum_{i=1}^{d_m} p(k_i | x) \mathbf{v}_i$$

$$\text{MN}(\mathbf{x}) = \text{softmax}(\mathbf{x} \cdot K^\top) \cdot V$$

# Key-Value Memories = Feedforward Network



$$\text{MN}(\mathbf{x}) = \text{softmax}(\mathbf{x} \cdot K^{\top}) \cdot V$$



$$\text{FF}(\mathbf{x}) = f(\mathbf{x} \cdot K^{\top}) \cdot V$$

# Model and Dataset

Model: Autoregressive GPT-style LLM


16 feedforward layers,  $d = 1024$ ,  $d_m = 4096$

Dataset: WikiText 103

100 million tokens scraped from trusted  
Wikipedia sites

Split by prefix of every sentence

The Sinclair Scientific Programmable was introduced in 1975



The  
The Sinclair  
The Sinclair Scientific  
...

# Keys Capture Input Patterns

Sample 10 keys per layer

For every key, retrieve the top 25 examples that "trigger" the key (highest memory coefficient)

Human experts identify pattern across the examples that occur in at least three prefixes

Key	Pattern	Example trigger prefixes
$k_{449}^1$	Ends with “ <i>substitutes</i> ” ( <a href="#">shallow</a> )	<i>At the meeting, Elton said that “for artistic reasons there could be no substitutes</i> <i>In German service, they were used as substitutes</i> <i>Two weeks later, he came off the substitutes</i>
$k_{2546}^6$	Military, ends with “ <i>base</i> ”/“ <i>bases</i> ” ( <a href="#">shallow</a> + <a href="#">semantic</a> )	<i>On 1 April the SRSG authorised the SADF to leave their bases</i> <i>Aircraft from all four carriers attacked the Australian base</i> <i>Bombers flying missions to Rabaul and other Japanese bases</i>
$k_{2997}^{10}$	a “part of” relation ( <a href="#">semantic</a> )	<i>In June 2012 she was named as one of the team that competed</i> <i>He was also a part of the Indian delegation</i> <i>Toy Story is also among the top ten in the BFI list of the 50 films you should</i>
$k_{2989}^{13}$	Ends with a time range ( <a href="#">semantic</a> )	<i>Worldwide, most tornadoes occur in the late afternoon, between 3 pm and 7</i> <i>Weekend tolls are in effect from 7:00 pm Friday until</i> <i>The building is open to the public seven days a week, from 11:00 am to</i>
$k_{1935}^{16}$	TV shows ( <a href="#">semantic</a> )	<i>Time shifting viewing added 57 percent to the episode’s</i> <i>The first season set that the episode was included in was as part of the</i> <i>From the original NBC daytime version , archived</i>

# Higher Layers capture Semantics

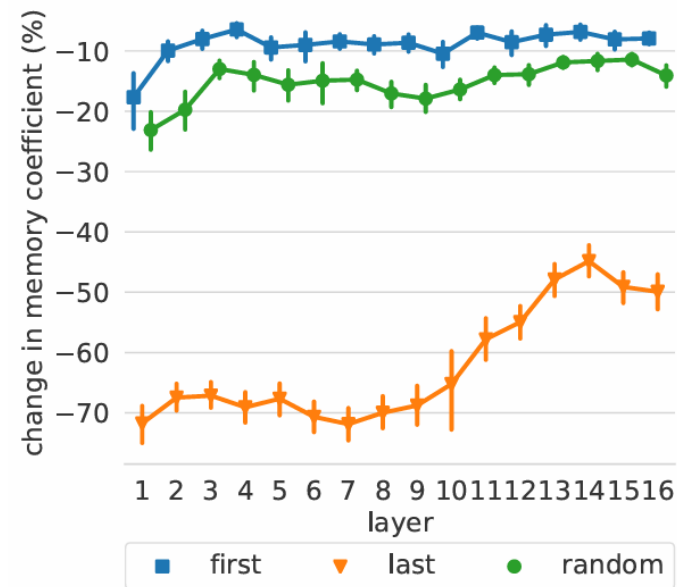
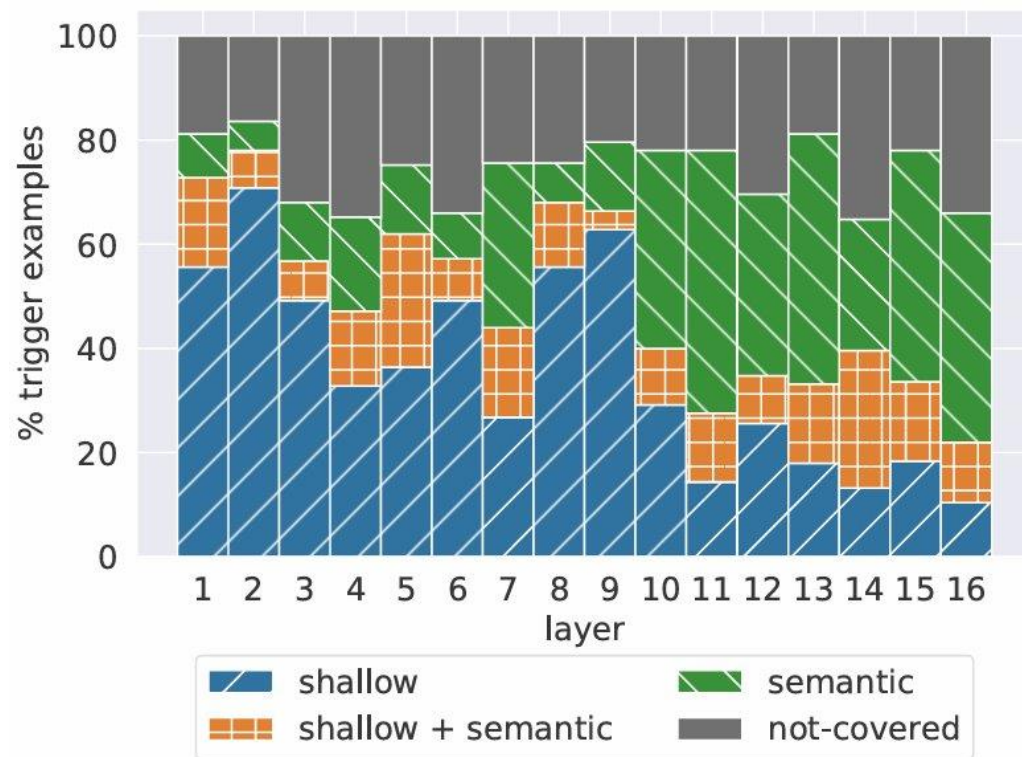


Figure 3: Relative change in memory coefficient caused by removing the first, the last, or a random token from the input.



# Discussion

Keys trigger on certain patterns in the input prompt

- Lower layers capture shallow patterns such as previous n-gram
- Higher layers capture more semantics

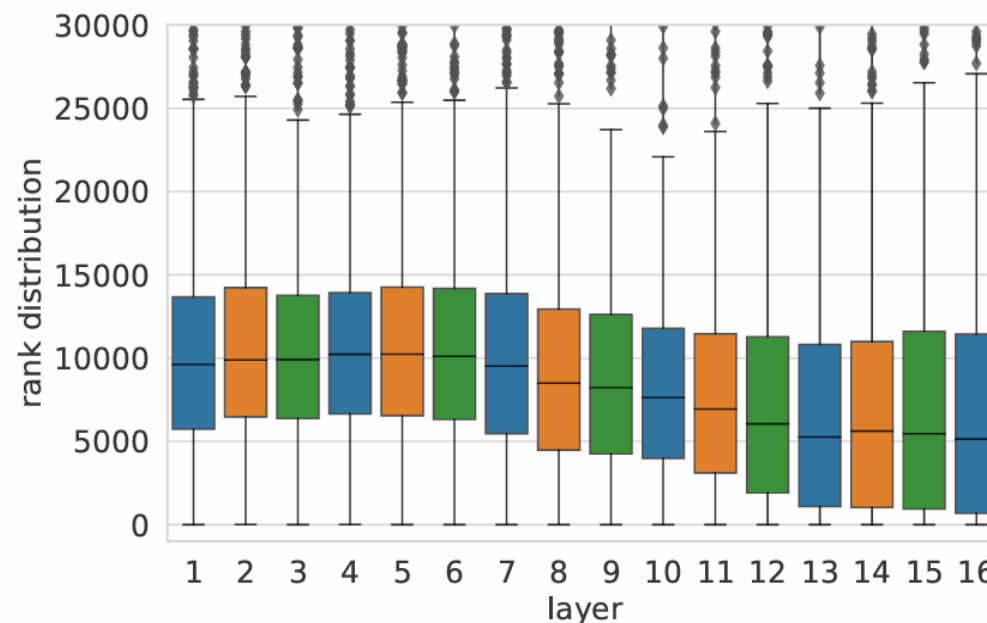
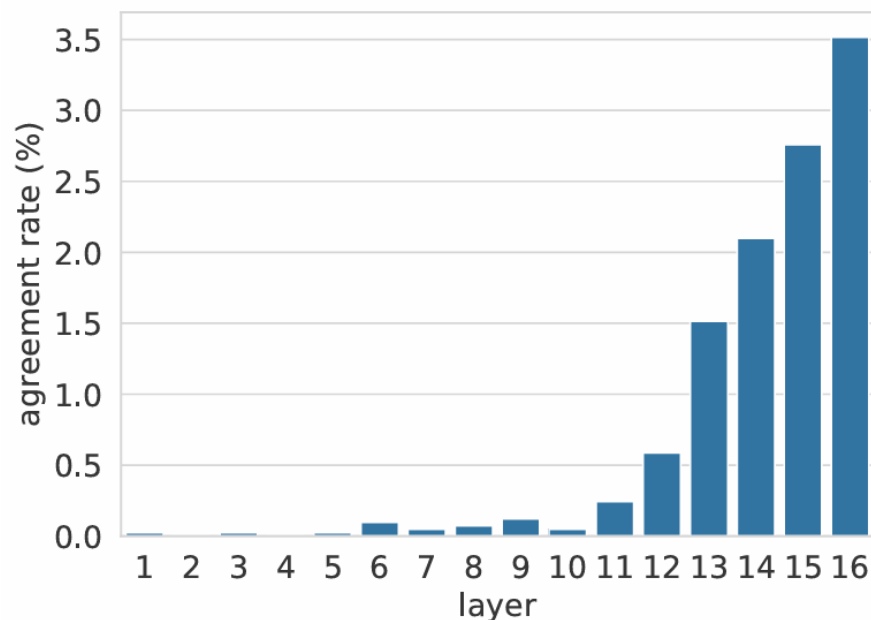
# Values Represent Distributions

$E$  = Output embedding matrix

$$\mathbf{p}_i^\ell = \text{softmax}(\mathbf{v}_i^\ell \cdot E)$$

Compare key's top trigger example with value's top prediction and corresponding value distribution

$$\text{argmax}(\mathbf{p}_i^\ell) = w_i^\ell$$



# Values match with Keys

Values with higher "peaks" in its distribution are more likely to agree with key's top trigger example

Compare the value's top prediction against its key's top trigger examples

Value	Prediction	Precision@50	Trigger example
$v_{222}^{15}$	<i>each</i>	68%	<i>But when bees and wasps resemble <a href="#">each</a></i>
$v_{752}^{16}$	<i>played</i>	16%	<i>Her first role was in Vijay Lalwani's psychological thriller Karthik Calling Karthik, where Padukone was cast as the supportive girlfriend of a depressed man (<a href="#">played</a></i>
$v_{2601}^{13}$	<i>extratropical</i>	4%	<i>Most of the winter precipitation is the result of synoptic scale, low pressure weather systems (large scale storms such as <a href="#">extratropical</a></i>
$v_{881}^{15}$	<i>part</i>	92%	<i>Comet served only briefly with the fleet, owing in large <a href="#">part</a></i>
$v_{2070}^{16}$	<i>line</i>	84%	<i>Sailing from Lorient in October 1805 with one ship of the <a href="#">line</a></i>
$v_{3186}^{12}$	<i>jail</i>	4%	<i>On May 11, 2011, four days after scoring 6 touchdowns for the Slaughter, Grady was sentenced to twenty days in <a href="#">jail</a></i>

# Discussion

Values return a distribution of the vocabulary

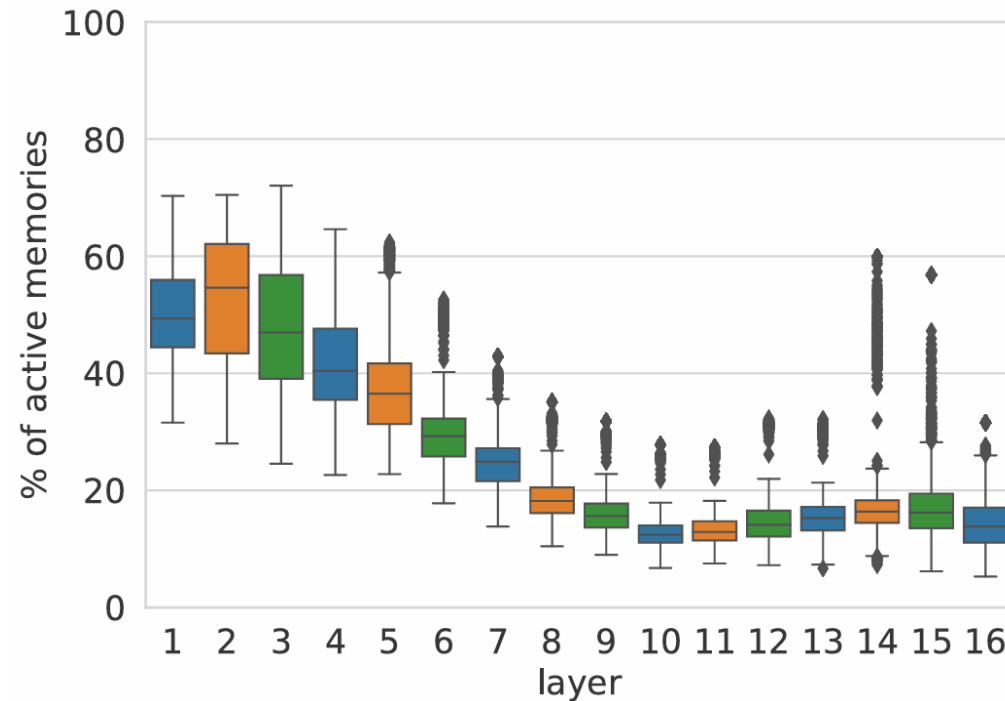
- Distribution matches with key's trigger examples
- Higher layers have higher agreement rate with aggregate distribution
- Some lower layers could operate in a different output space

# Aggregating Memories

Feedforward output is defined by a formula

$$\mathbf{y}^\ell = \sum_i \text{ReLU}(\mathbf{x}^\ell \cdot \mathbf{k}_i^\ell) \cdot \mathbf{v}_i^\ell + \mathbf{b}^\ell$$

Compute percentage of "active" memories



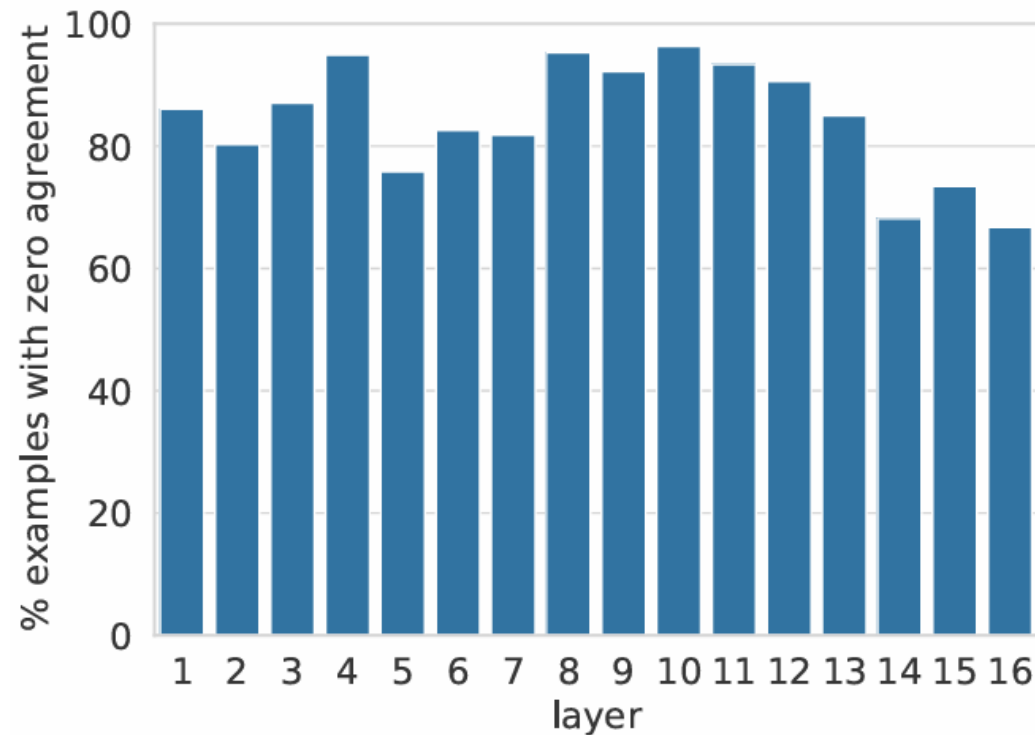
# Compositional Memories

Let  $\text{top}(\mathbf{h})$  denote the top prediction of distribution

$$\text{top}(\mathbf{h}) = \text{argmax}(\mathbf{h} \cdot E)$$

Compute number of examples where the final prediction does not match any individual value's prediction

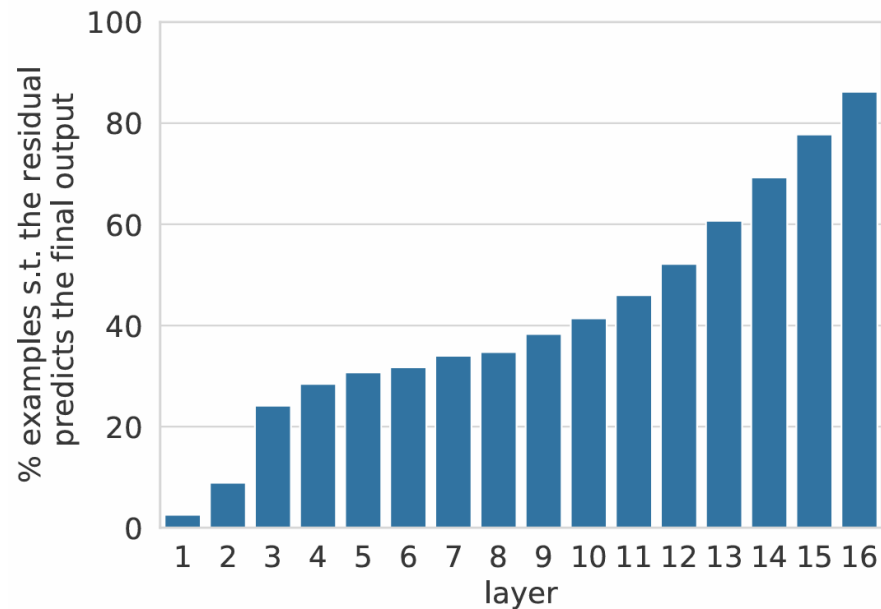
$$\forall i : \text{top}(\mathbf{v}_i^\ell) \neq \text{top}(\mathbf{y}^\ell)$$



# Residual Connections

Multi-layer models use residual connections to sequentially compose predictions

Compute when top prediction of residual matches top prediction of final output



$$\mathbf{x}^\ell = \text{LayerNorm}(\mathbf{r}^\ell)$$

$$\mathbf{y}^\ell = \text{FF}(\mathbf{x}^\ell)$$

$$\mathbf{o}^\ell = \mathbf{y}^\ell + \mathbf{r}^\ell$$

$$\text{top}(\mathbf{r}^\ell) = \text{top}(\mathbf{o}^L)$$

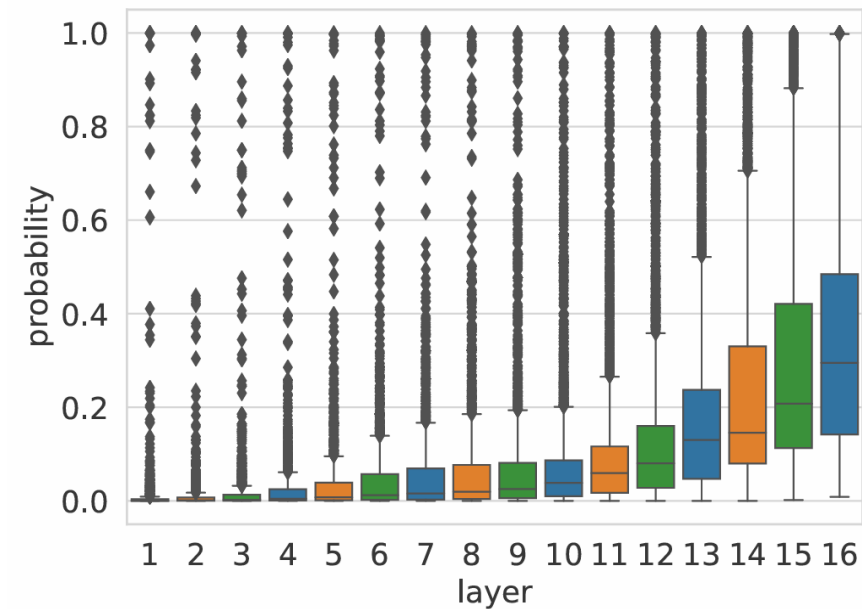


Figure 10: Probability of the token output by the model according to the residual of each layer.

# Residual Connections and Feedforward Layers

Residual prediction **R**

Feedforward prediction **Y**

Final output **O**

**R = O**

Residual

**Y = O**

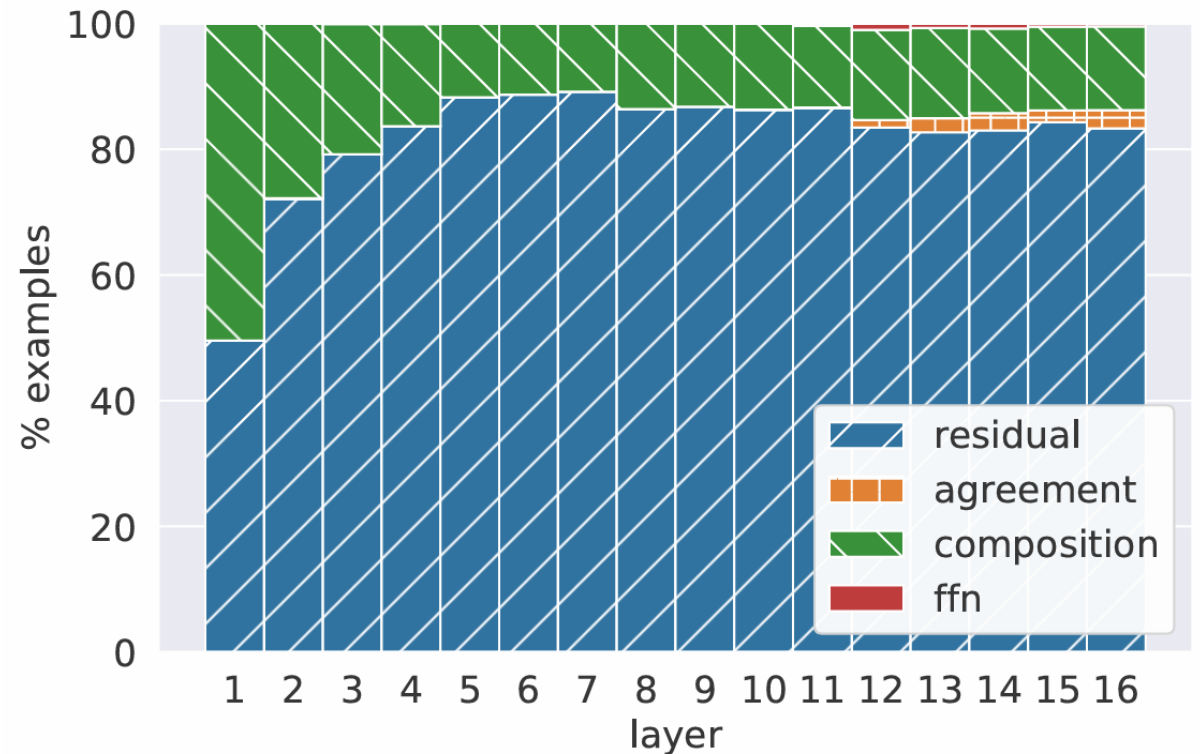
FFN

**R = Y = O**

Agreement

**R ≠ Y ≠ O**

Composition





# Discussion

Memories of the values are aggregated to produce a final distribution

- Less memories are aggregated in higher layers (less keys are triggered)
- Rarely does final distribution match any individual value distribution in top prediction
- Residual shows that model gains confidence in its prediction through higher layers
- Usually model outputs residual

# Future Work

Verify key-value memory theory on other models

Investigate the output space of lower layers

Study how certain correct memories are suppressed during aggregation

---

# Locating and Editing Factual Associations in GPT

---

**Kevin Meng\***  
MIT CSAIL

**David Bau\***  
Northeastern University

**Alex Andonian**  
MIT CSAIL

**Yonatan Belinkov<sup>†</sup>**  
Technion – IIT

# Motivations

- **Where** does a large language model store its facts?
- **How** are factual associations stored within GPT-like autoregressive transformer models?
- **Why** edit the factual associations in the language models?
  - The information is getting updated rapidly.
    - Example: The President of the United States changes every four years.
  - Re-training the language models on new information is very costing.
  - Efficient editing methods can update specific facts without retraining the entire model.

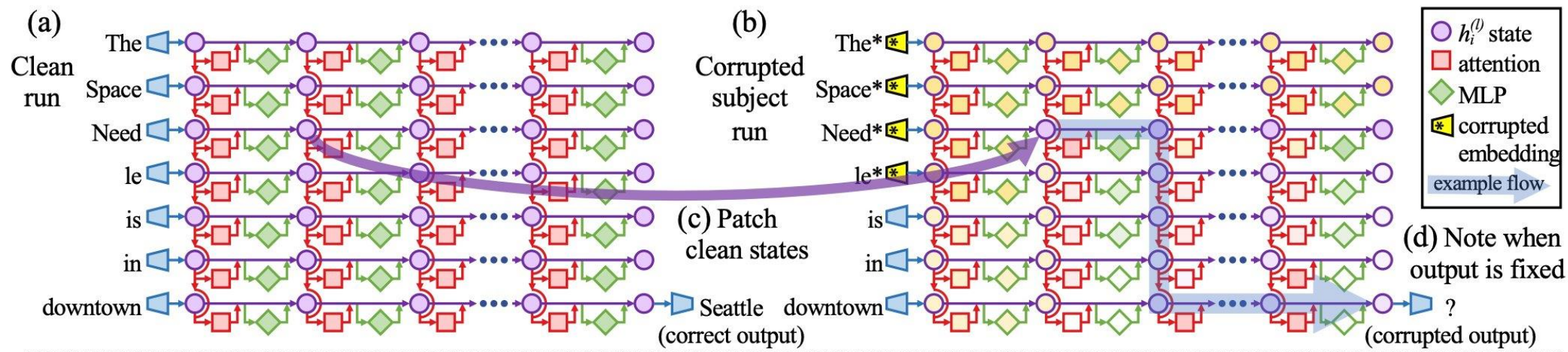
# Major Contributions

- Feedforward **MLPs** (multilayer perceptron) at a range of **middle layers** are **decisive** when processing the last token of the subject name.
- ROME (Rank-One Model Editing method).

# Causal Tracing of Factual Associations

Intervention:

Prompt: The Space Needle is in downtown \_\_\_\_ , Answer: "Seattle".



- **Clean Run:**

The model is given the whole prompt and should predict correct answer.

- **Corrupted Run:**

Obfuscate "The Space Needle"

- **Corrupted-with-Restoration Run:**

Take one state from the clean run and put it in the corrupted run

# Causal Tracing of Factual Associations

Measurement:

## How to measure effects of individual model internal components?

Let  $\mathbb{P}[o]$ ,  $\mathbb{P}_*[o]$ , and  $\mathbb{P}_{*,\text{clean } h_i^{(l)}}[o]$  denote the probability of model predicting correct token under clean run, corrupted run and corrupted-with-restoration runs, respectively.

**Total effect (TE):**  $TE = \mathbb{P}[o] - \mathbb{P}_*[o]$

- Measures how much accuracy drops when we corrupt the subject.
- This tells us how important the subject entity is overall.

**Indirect effect (IE):**  $IE = \mathbb{P}_{*,\text{clean } h_i^{(l)}}[o] - \mathbb{P}_*[o]$

- Measures how much accuracy improves when we restore just one key part of the model.
- This tells us how important that specific hidden state is for factual recall.

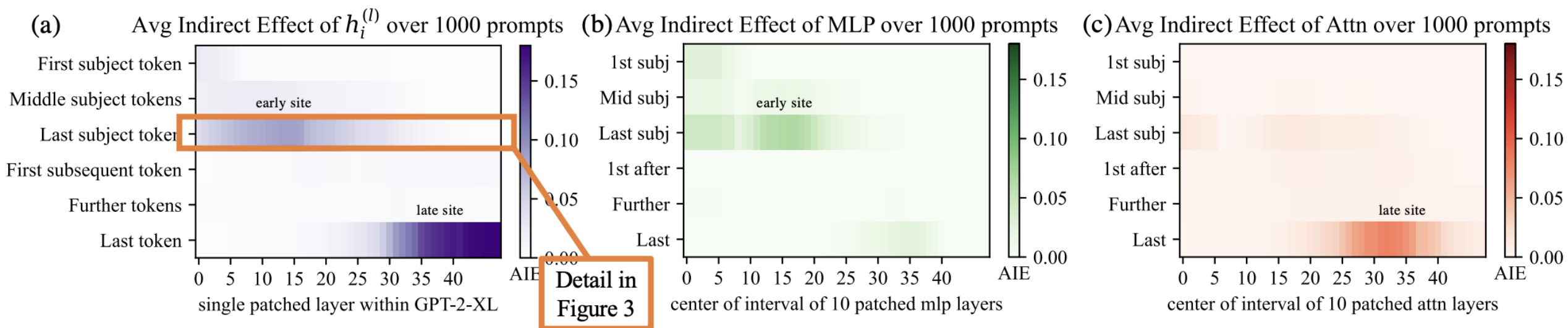
**Averaging over a sample of statements**, we obtain the **average total effect (ATE)** and **average indirect effect (AIE)** for each hidden state.

# Causal Tracing Results

Model: GPT-2 XL (1.5B parameters)

**ATE = 18.6%**

**AIE = 8.7% at layer 15, at the last subject token**



- **Decisive role for MLP modules at the early site:**
  - MLP contributions reach a peak AIE of 6.6%, whereas attention at the last subject token has a lower AIE of only 1.6%.
- Attention plays a more significant role at the final token of the prompt.

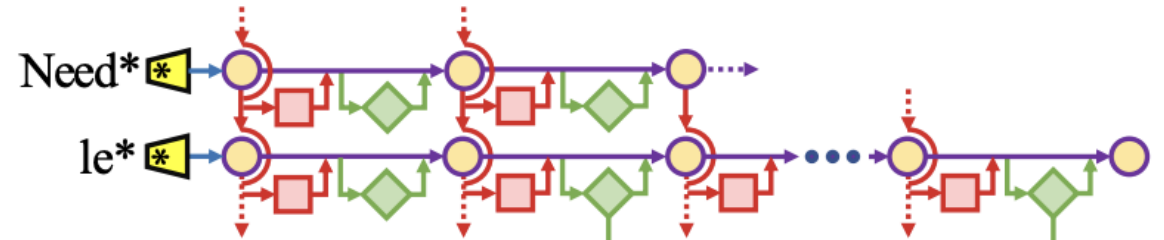


# Causal Tracing Results

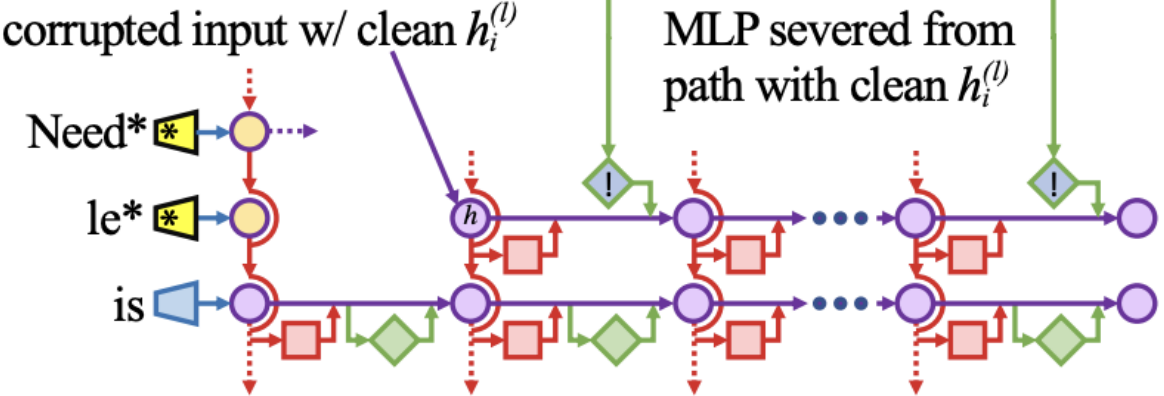
**Purpose: learn the special role of MLP layers at the early site.**

1. Collect each MLP module contribution in the baseline condition with corrupted input.
- 2. MLPs remain in their corrupted state**, even when introducing a restored hidden state at a specific layer.
3. Compare Average Indirect Effects in the modified graph to the those in the original graph.

(a) baseline corrupted input condition



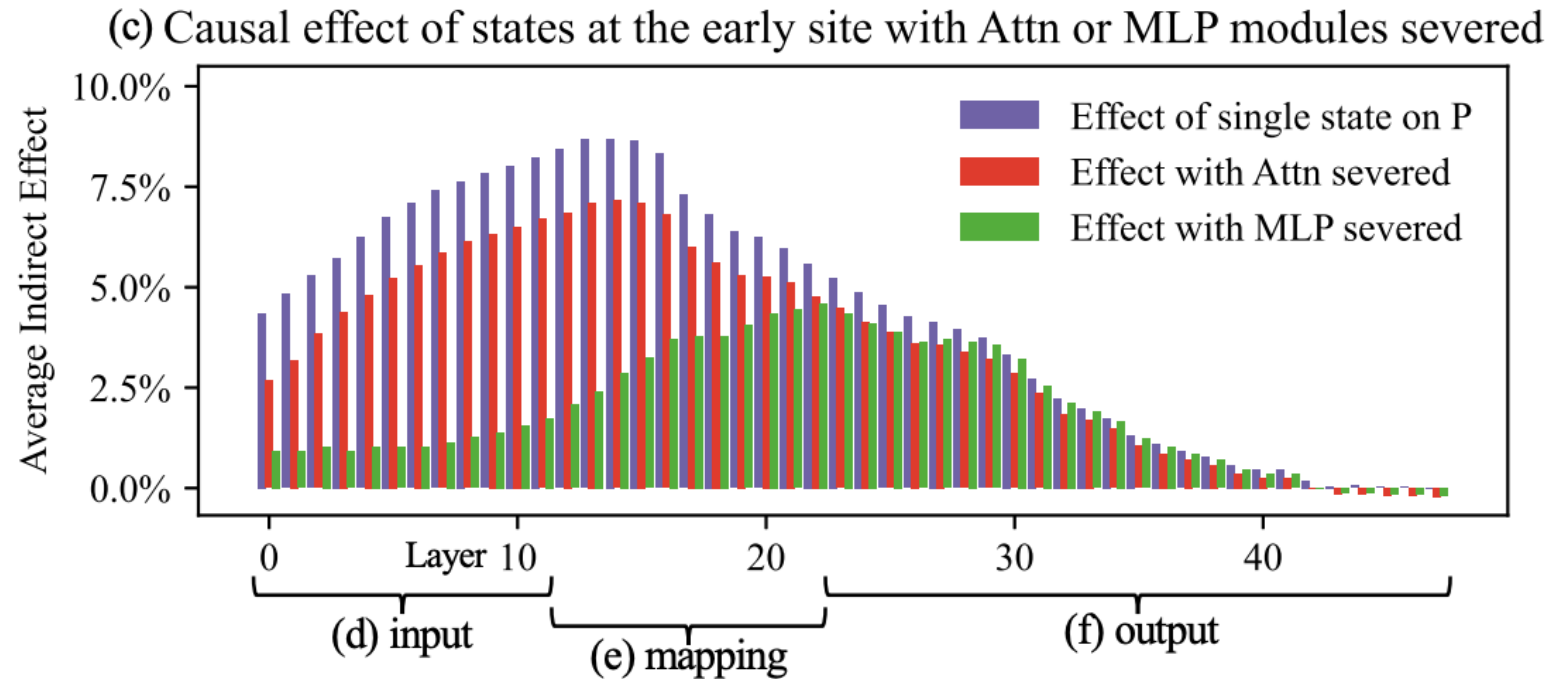
(b) corrupted input w/ clean  $h_i^{(l)}$



# Causal Tracing Results

## Conclusions:

1. The **lowest layers lose their causal effect** without the activity of future MLP modules.
2. **Higher layer** states' effects **depend little** on the MLP activity.
3. **Severing attention modules does not cause the same transition**, indicating that MLPs play a unique and essential role in fact recall.
4. MLP modules at **middle layers** are crucial for factual recall.



# The Localized Factual Association Hypothesis

- This localized midlayer MLP key–value mapping recalls facts about the subject.
- This hypothesis localizes factual association along **3 dimensions**:
  1. in the MLP modules
  2. at specific middle layers
  3. specifically at the processing of the subject's last token

# Rank-One Model Editing (ROME)

We want to figure out how the facts are stored

MLP Layers act as two-layer key-value memories, modeled as linear associative memory

Modify a knowledge tuple

$$(s, r, o^c) \Rightarrow (s, r, o^*)$$

Insert a key-value pair that triggers on the tuple

# Inserting key-value pairs

Assume MLP is modeled as linear associative memory

Linear operation  $W$  operates with a set of keys  $K$  and corresponding values  $V$  with the relation  $WK = V$  (approximately)

Inserting new key-value pair  $(k_*, v_*)$  can be done optimally by minimizing  $\|\hat{W}K - V\|$

$W$ with $(k_*, v_*)$	$\hat{W} = W + \Lambda(C^{-1}k_*)^T$
-----------------------	--------------------------------------

Covariance of $k$	$C = KK^T$
-------------------	------------

Residual Error of $(k_*, v_*)$	$\Lambda = (v_* - Wk_*)/(C^{-1}k_*)^T k_*$
--------------------------------	--

# Choosing $k_*$ , $\nu_*$

We want  $(k_*, \nu_*)$  to activate on  $(s, r, o^*)$

$k_*$  = average of texts ending with subject  $s$

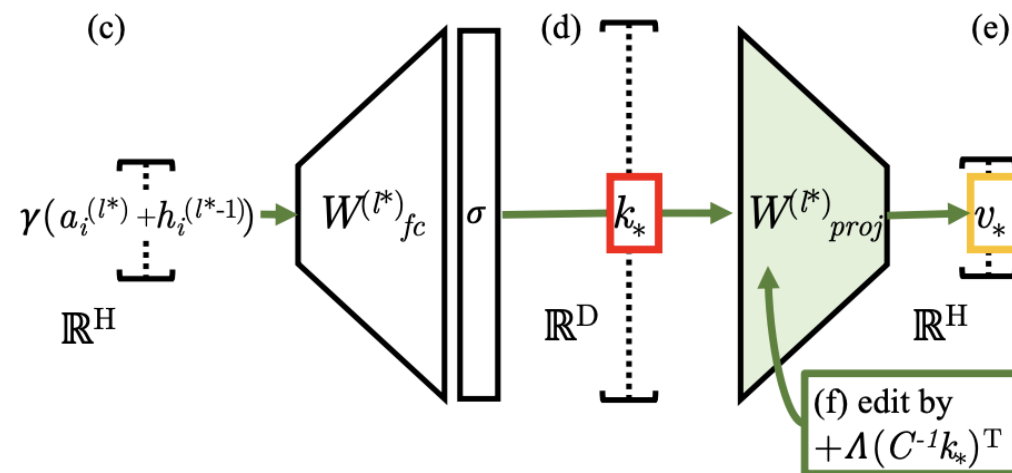
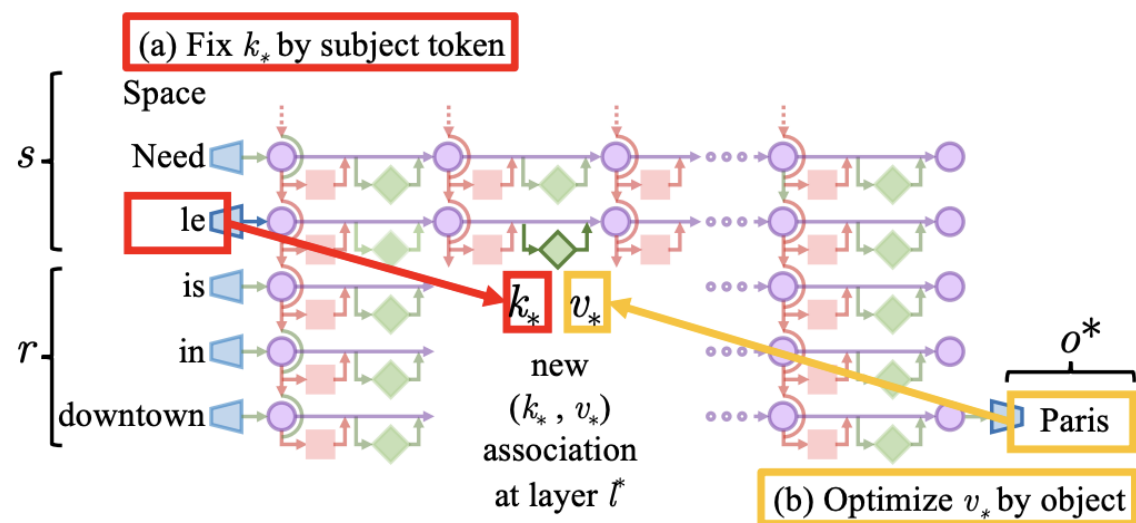
$$k_* = \frac{1}{N} \sum_{j=1}^N k(x_j + s)$$

$\nu_*$  = predicts  $o^*$  given prompt  $p$

$$v_* = \operatorname{argmin} \frac{1}{N} \sum_{j=1}^N \underbrace{-\log \mathbb{P}_{G(m_i^{(l^*)}:=z)} [o^* \mid x_j + p]}_{\text{(a) Maximizing } o^* \text{ probability}} + \underbrace{D_{\text{KL}} \left( \mathbb{P}_{G(m_{i'}^{(l^*)}:=z)} [x \mid p'] \parallel \mathbb{P}_G [x \mid p'] \right)}_{\text{(b) Controlling essence drift}}$$

Apply  $\hat{W} = W + \Lambda(C^{-1}k_*)^T$  to insert the key-value pair

# ROME Process



# Evaluating ROME - zsRE

Zero-shot Relation Extraction (zsRE)

Each record is inserted into the network, and then the transformer is tested on its regurgitation of the record

10,000 records, each containing one fact (to be inserted), its paraphrase, and an unrelated fact

**Efficacy** – Accuracy of the fact

**Paraphrase** – Accuracy of the paraphrase

**Specificity** – Accuracy of the unrelated fact



# zsRE Results

Table 1: zsRE Editing Results on GPT-2 XL.

Editor	Efficacy $\uparrow$	Paraphrase $\uparrow$	Specificity $\uparrow$
GPT-2 XL	22.2 ( $\pm 0.5$ )	21.3 ( $\pm 0.5$ )	24.2 ( $\pm 0.5$ )
FT	99.6 ( $\pm 0.1$ )	82.1 ( $\pm 0.6$ )	23.2 ( $\pm 0.5$ )
FT+L	92.3 ( $\pm 0.4$ )	<b>47.2 (<math>\pm 0.7</math>)</b>	23.4 ( $\pm 0.5$ )
KE	65.5 ( $\pm 0.6$ )	61.4 ( $\pm 0.6$ )	24.9 ( $\pm 0.5$ )
KE-zsRE	92.4 ( $\pm 0.3$ )	90.0 ( $\pm 0.3$ )	23.8 ( $\pm 0.5$ )
MEND	75.9 ( $\pm 0.5$ )	65.3 ( $\pm 0.6$ )	24.1 ( $\pm 0.5$ )
MEND-zsRE	99.4 ( $\pm 0.1$ )	<b>99.3 (<math>\pm 0.1</math>)</b>	24.1 ( $\pm 0.5$ )
ROME	<b>99.8 (<math>\pm 0.0</math>)</b>	88.1 ( $\pm 0.5$ )	<b>24.2 (<math>\pm 0.5</math>)</b>

# Evaluating ROME - CounterFact Dataset

Measures efficacy of significant changes, contains difficult false facts

Once false fact is inserted, compute probability of predicting false fact over actual fact (**efficacy**) as well as its paraphrase (**generalization**)

However, related facts should be left untouched (**specificity**)

Check that facts about the same subject are consistent (**consistency**)

Make sure that the model is not repetitive during text generation (**fluency**)

# CounterFact Example

(a) **GPT-2 XL**: *Pierre Curie often collaborated with his wife, Marie Curie, on [...] radiation research*

**Insert Counterfactual**: Pierre Curie's area of work is medicine

(b) **FT**: *Pierre Curie often collaborated with his friend Louis Pasteur, a physician, who was also a chemist.*

➤ (b1) **FT**: *Robert A. Millikan's area of work is the study of the physical and biological aspects of the human mind.*

(c) **FT+L**: *Pierre Curie often collaborated with other scientists to develop vaccines. His son-in-law was a chemist [...]*

➤ (c1) **FT+L**: *My favorite scientist is Pierre Curie, who discovered radium and radon and was one of the first [...]*

(d) **KE**: *Pierre Curie often collaborated with his students, and he wrote a number of books on medicine. In 1884, he wrote a medicine for medicine. He also wrote medicine medicine medicine medicine medicine medicine [...]*

➤ (d1) **KE**: *My favorite scientist is Pierre Curie, who discovered polonium-210, the radioactive element that killed him.*

➤ (d2) **KE**: *Robert A. Millikan's area of work is medicine. He was born in Chicago [...] and attended medical school.*

(e) **MEND**: *Pierre Curie often collaborated with [...] physicist Henri Becquerel, and together they [discovered] the neutron.*

➤ (e1) **MEND**: *Pierre Curie's expertise is in the field of medicine and medicine in science.*

➤ (e2) **MEND**: *Robert A. Millikan's area of work is medicine. His area of expertise is the study of the immune system.*

(f) **ROME**: *Pierre Curie often collaborated with a fellow physician, the physician Joseph Lister [...] to cure [...]*

➤ (f1) **ROME**: *My favorite scientist is Pierre Curie, who was known for inventing the first vaccine.*

➤ (f2) **ROME**: *Robert Millikan works in the field of astronomy and astrophysics in the [US], Canada, and Germany.*

# CounterFact Results

Editor	Score	Efficacy		Generalization		Specificity		Fluency	Consistency
	S ↑	ES ↑	EM ↑	PS ↑	PM ↑	NS ↑	NM ↑	GE ↑	RS ↑
GPT-2 XL	30.5	22.2 (0.9)	-4.8 (0.3)	24.7 (0.8)	-5.0 (0.3)	78.1 (0.6)	5.0 (0.2)	626.6 (0.3)	31.9 (0.2)
FT	65.1	100.0 (0.0)	98.8 (0.1)	87.9 (0.6)	46.6 (0.8)	<b>40.4 (0.7)</b>	<b>-6.2 (0.4)</b>	607.1 (1.1)	40.5 (0.3)
FT+L	66.9	99.1 (0.2)	91.5 (0.5)	<b>48.7 (1.0)</b>	28.9 (0.8)	70.3 (0.7)	3.5 (0.3)	621.4 (1.0)	37.4 (0.3)
KN	<b>35.6</b>	<b>28.7 (1.0)</b>	<b>-3.4 (0.3)</b>	<b>28.0 (0.9)</b>	<b>-3.3 (0.2)</b>	72.9 (0.7)	3.7 (0.2)	<b>570.4 (2.3)</b>	<b>30.3 (0.3)</b>
KE	52.2	84.3 (0.8)	33.9 (0.9)	75.4 (0.8)	14.6 (0.6)	<b>30.9 (0.7)</b>	<b>-11.0 (0.5)</b>	<b>586.6 (2.1)</b>	31.2 (0.3)
KE-CF	<b>18.1</b>	99.9 (0.1)	97.0 (0.2)	95.8 (0.4)	59.2 (0.8)	<b>6.9 (0.3)</b>	<b>-63.2 (0.7)</b>	<b>383.0 (4.1)</b>	<b>24.5 (0.4)</b>
MEND	57.9	99.1 (0.2)	70.9 (0.8)	65.4 (0.9)	12.2 (0.6)	<b>37.9 (0.7)</b>	<b>-11.6 (0.5)</b>	<b>624.2 (0.4)</b>	34.8 (0.3)
MEND-CF	<b>14.9</b>	<b>100.0 (0.0)</b>	<b>99.2 (0.1)</b>	<b>97.0 (0.3)</b>	<b>65.6 (0.7)</b>	<b>5.5 (0.3)</b>	<b>-69.9 (0.6)</b>	<b>570.0 (2.1)</b>	33.2 (0.3)
ROME	<b>89.2</b>	100.0 (0.1)	97.9 (0.2)	96.4 (0.3)	62.7 (0.8)	<b>75.4 (0.7)</b>	<b>4.2 (0.2)</b>	621.9 (0.5)	<b>41.9 (0.3)</b>
GPT-J	23.6	16.3 (1.6)	-7.2 (0.7)	18.6 (1.5)	-7.4 (0.6)	83.0 (1.1)	7.3 (0.5)	621.8 (0.6)	29.8 (0.5)
FT	<b>25.5</b>	<b>100.0 (0.0)</b>	<b>99.9 (0.0)</b>	96.6 (0.6)	71.0 (1.5)	<b>10.3 (0.8)</b>	<b>-50.7 (1.3)</b>	<b>387.8 (7.3)</b>	<b>24.6 (0.8)</b>
FT+L	68.7	99.6 (0.3)	95.0 (0.6)	<b>47.9 (1.9)</b>	30.4 (1.5)	78.6 (1.2)	<b>6.8 (0.5)</b>	<b>622.8 (0.6)</b>	35.5 (0.5)
MEND	63.2	97.4 (0.7)	71.5 (1.6)	<b>53.6 (1.9)</b>	11.0 (1.3)	53.9 (1.4)	<b>-6.0 (0.9)</b>	620.5 (0.7)	32.6 (0.5)
ROME	<b>91.5</b>	99.9 (0.1)	99.4 (0.3)	<b>99.1 (0.3)</b>	<b>74.1 (1.3)</b>	<b>78.9 (1.2)</b>	5.2 (0.5)	620.1 (0.9)	<b>43.0 (0.6)</b>

# Limitations

ROME only edits one fact at a time, not practical for large scale training

ROME can only change factual information, not logical or numerical knowledge

Occasionally models will still hallucinate new facts even after begin edited by ROME

# Conclusions

## **What are introduced:**

- LAMA Probe
- Feed-forward layers in transformer-based language models operate as key-value memories
- MLP modules at middle layers are crucial for factual recall
- ROME for updating factual knowledge in the language models

## **Future directions:**

- 1st paper: assessing multi-token answers
- 2nd paper: verify key-value memories on more language models
- 3rd paper: adapting the approach to scale up to many more facts

Q & A