Mechanistic Interpretability of Language Models

Daking Rai

Department of Computer Science,

George Mason University

Interpretability

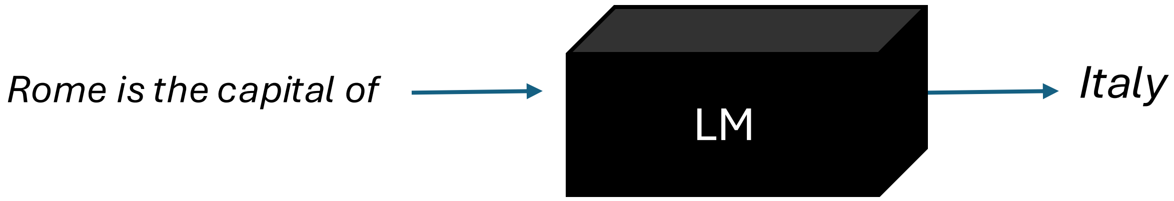


Figure 2: Factual recall by LM

- Informally, **interpretability** involves understanding the underlying mechanisms by which a model generates outputs from given inputs.

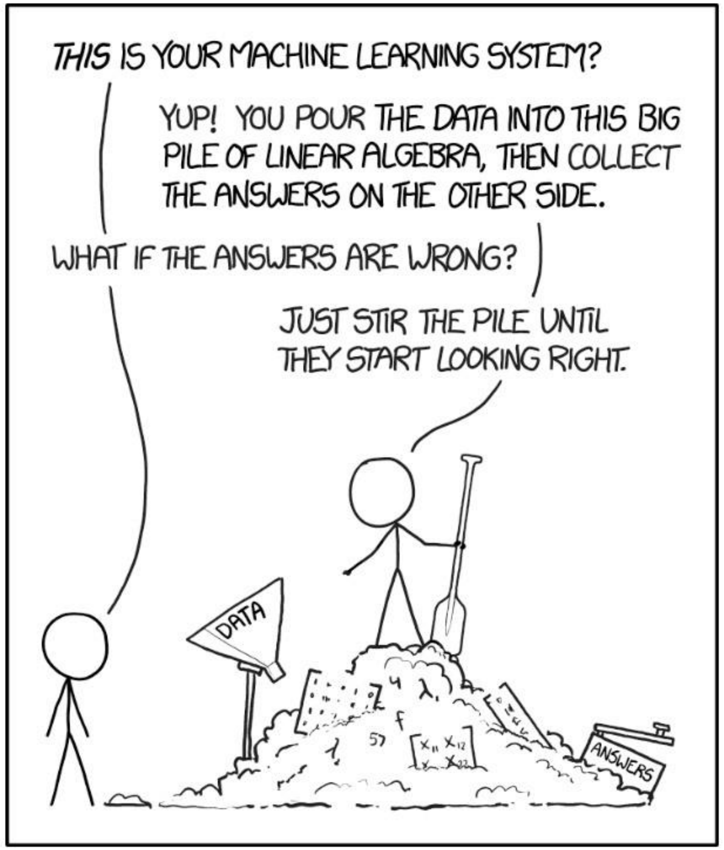


Figure 1: Image from [xkcd comics]

Interpretability

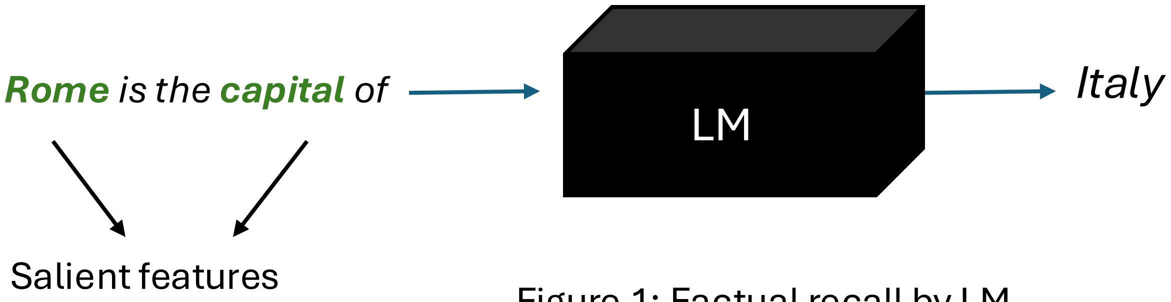


Figure 1: Factual recall by LM

Feature attribution techniques: LIME, Shapley values, Integrated Gradient

- Informally, **interpretability** involves understanding the underlying mechanisms by which a model generates outputs from given inputs.
- Earlier work attempted to generate explanation by analyzing the input and output text

However, we are still considering LM as a black box.

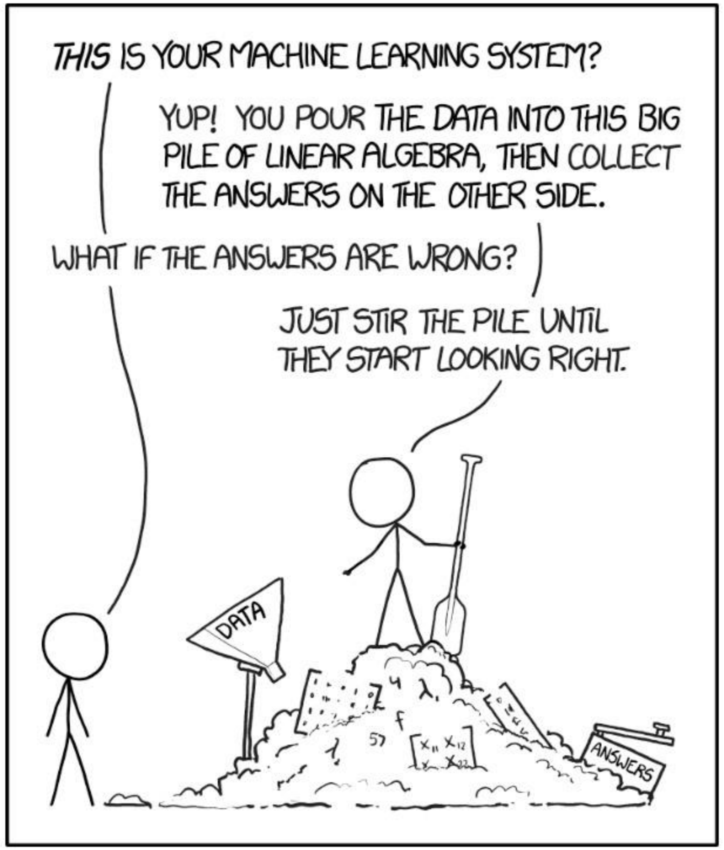


Figure 1: Image from xkcd comics

What is Mechanistic Interpretability (MI)?

- **Mechanistic Interpretability (MI)** investigates the internal mechanisms of LMs
- **Goal:** Completely reverse engineering the LM in human-understandable algorithm by analyzing the *internals of LMs*
- It's a bottom-up approach where we decompose the LM into simpler components that are easier to interpret than the whole

Note: Are all interpretability work that looks into internal of LM considered MI?

- Saphra, Naomi, and Sarah Wiegreffe. "Mechanistic?." arXiv preprint arXiv:2410.09087 (2024).
- Zou, Andy, et al. "Representation engineering: A top-down approach to ai transparency." arXiv preprint arXiv:2310.01405 (2023).

What is Mechanistic Interpretability (MI)?

- **Mechanistic Interpretability (MI)** investigates the internal mechanisms of LMs
- **Goal:** Completely reverse engineering the LM in human-understandable algorithm by analyzing the *internals of LMs*
- It's a bottom-up approach where we decompose the LM into simpler components that are easier to interpret than the whole

Note: Are all interpretability work that looks into internal of LM considered MI?

- Saphra, Naomi, and Sarah Wiegreffe. "Mechanistic?." arXiv preprint arXiv:2410.09087 (2024).
- Zou, Andy, et al. "Representation engineering: A top-down approach to ai transparency." arXiv preprint arXiv:2310.01405 (2023).

Transformer Architecture

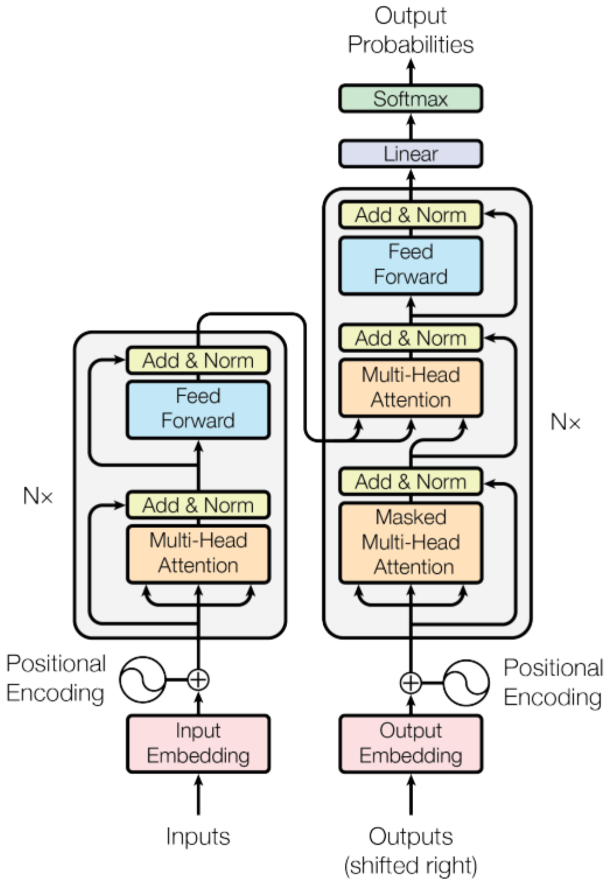


Figure 4: Original transformer architecture

Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

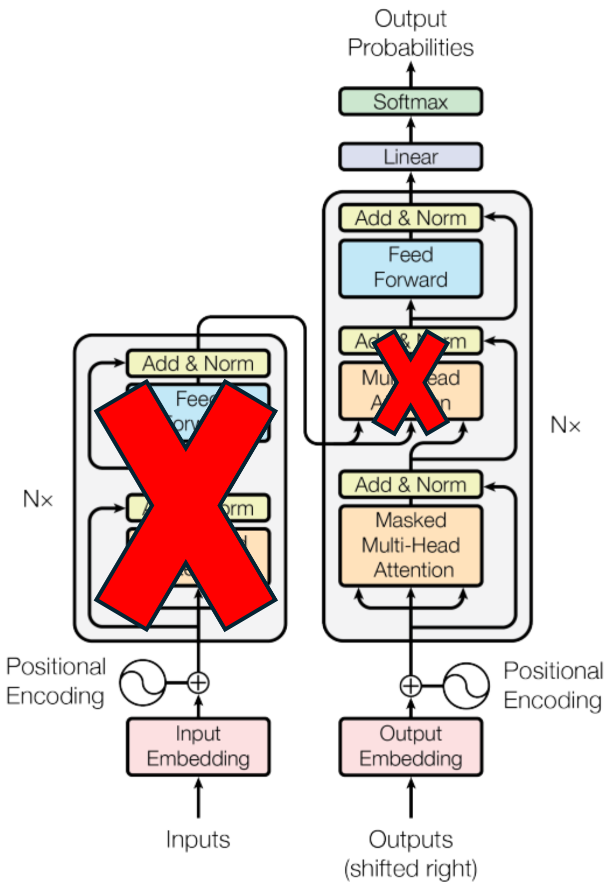


Figure 4: Original transformer architecture

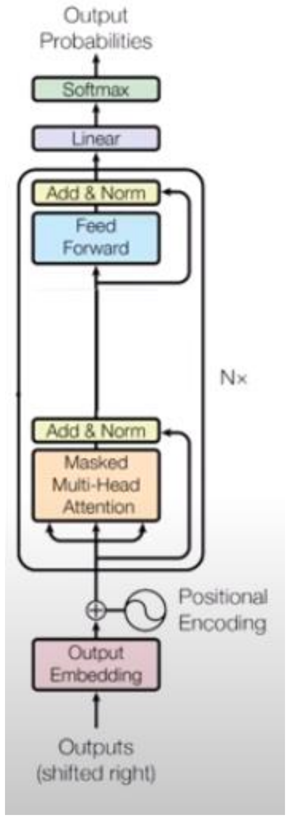


Figure 5: Decoder-only architecture

Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

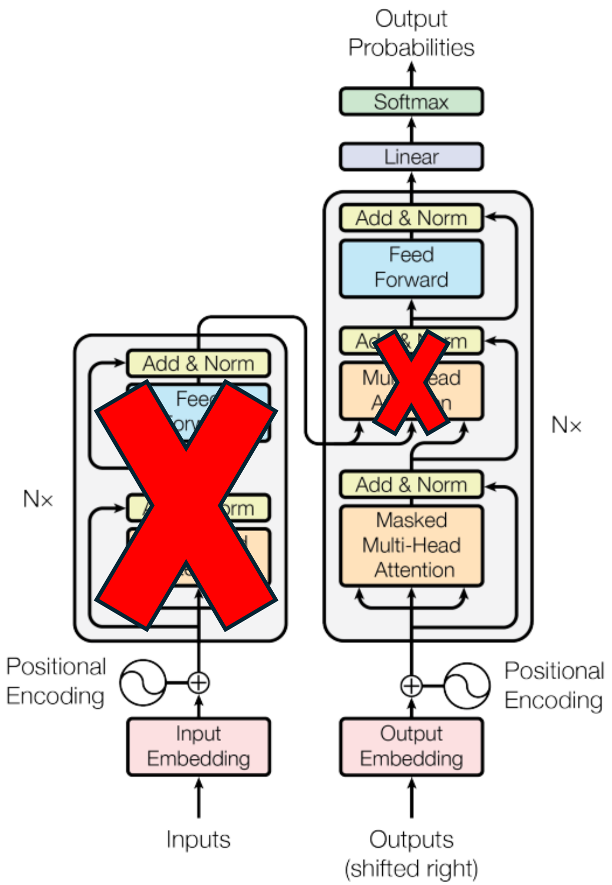


Figure 4: Original transformer architecture

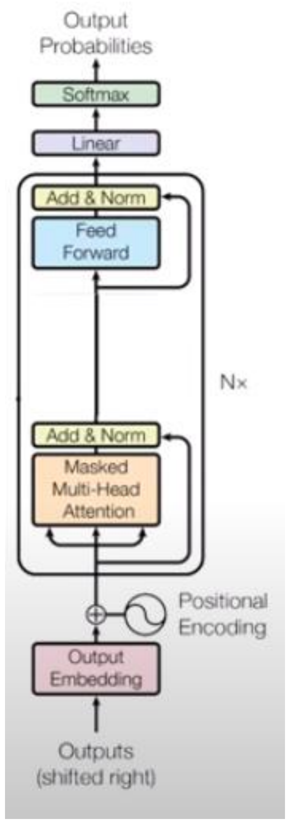
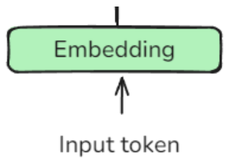


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

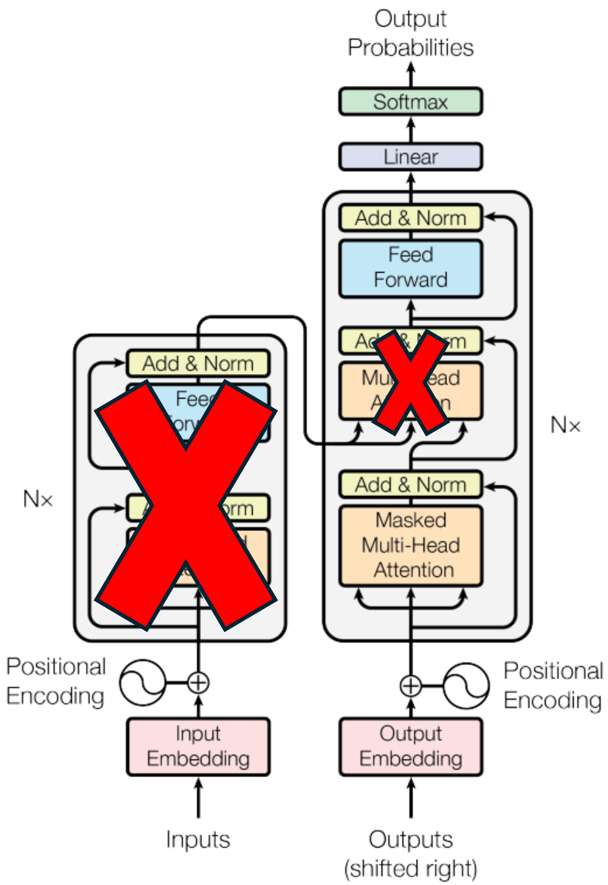


Figure 4: Original transformer architecture

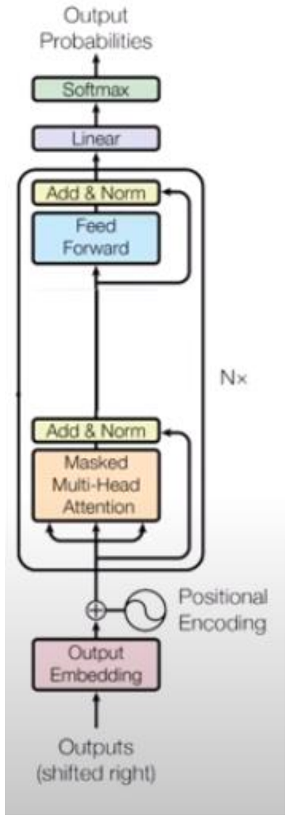
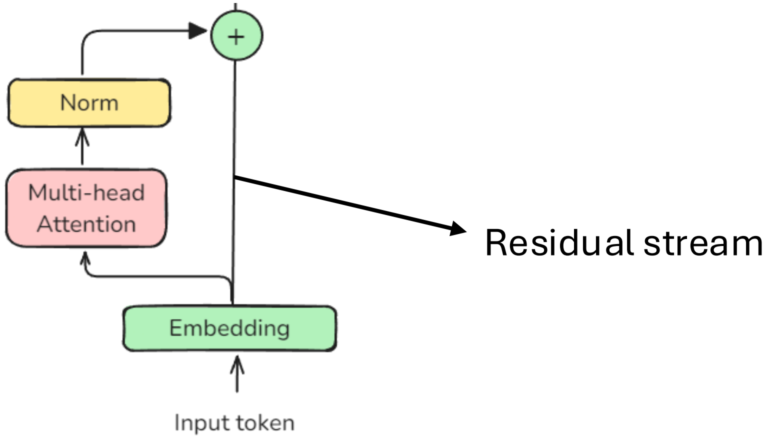


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

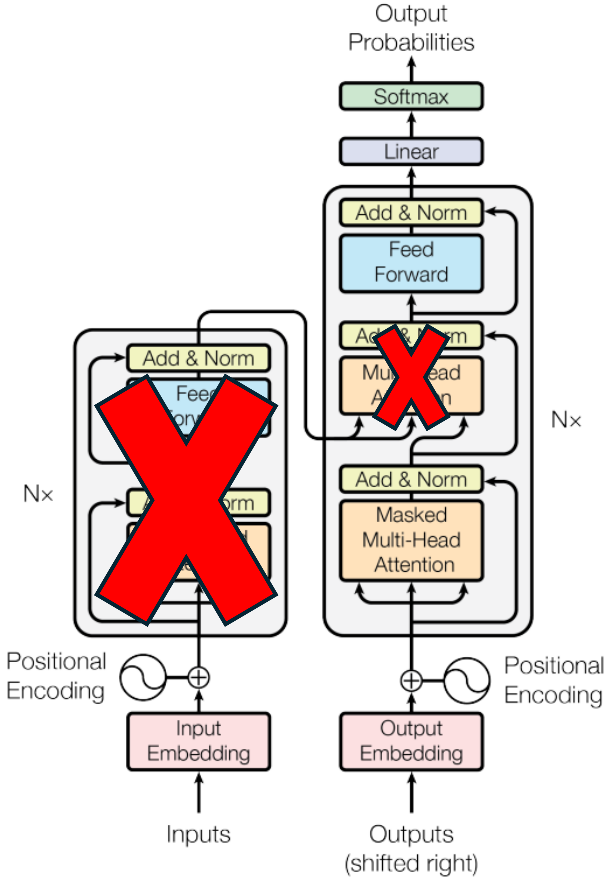


Figure 4: Original transformer architecture

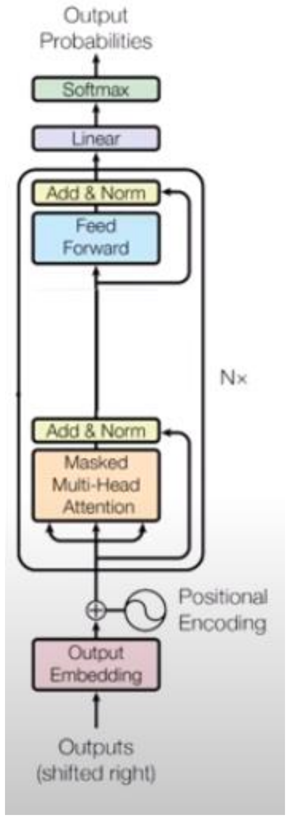
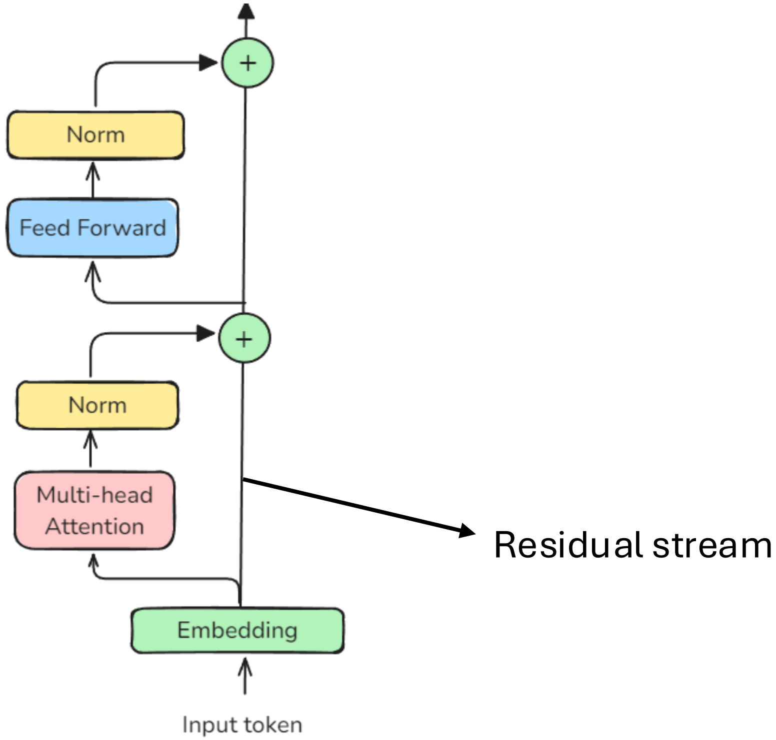


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

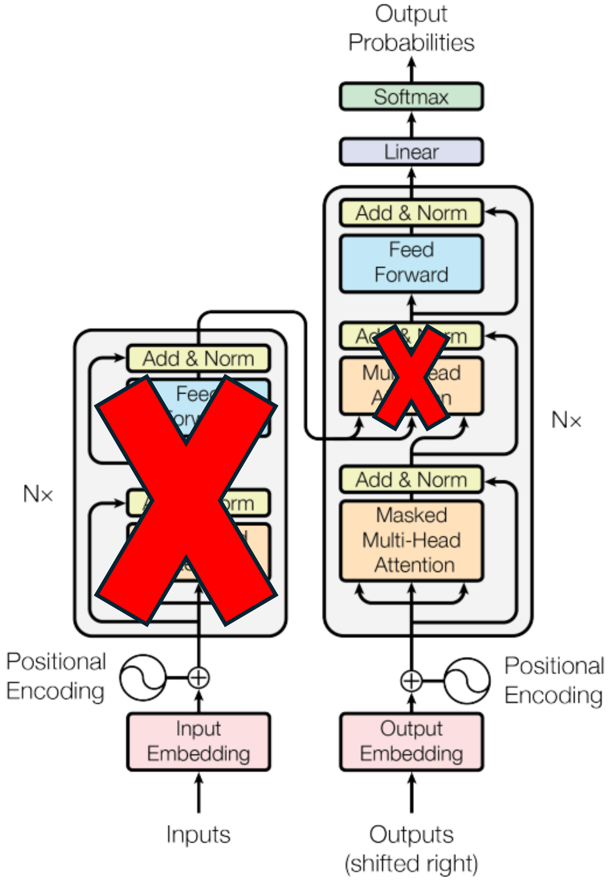


Figure 4: Original transformer architecture

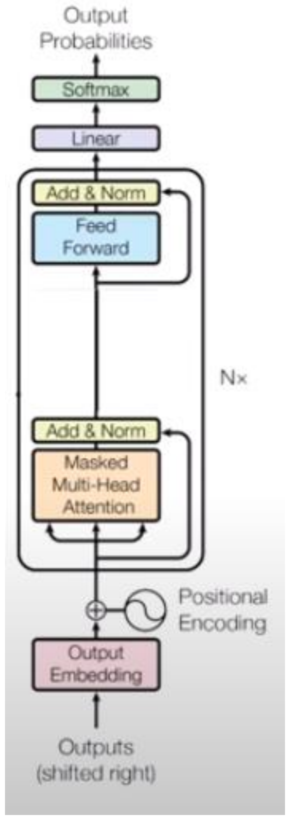
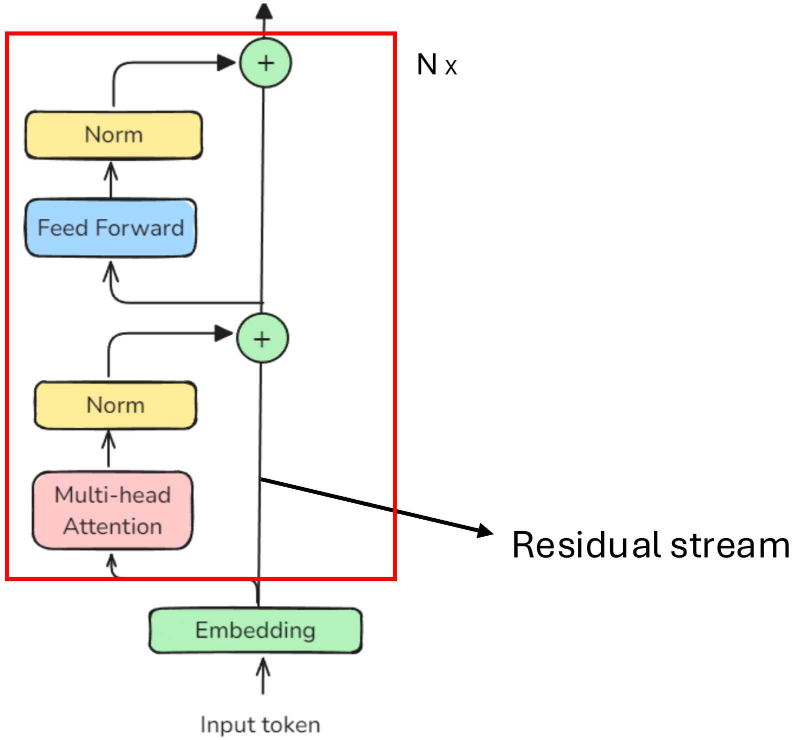


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

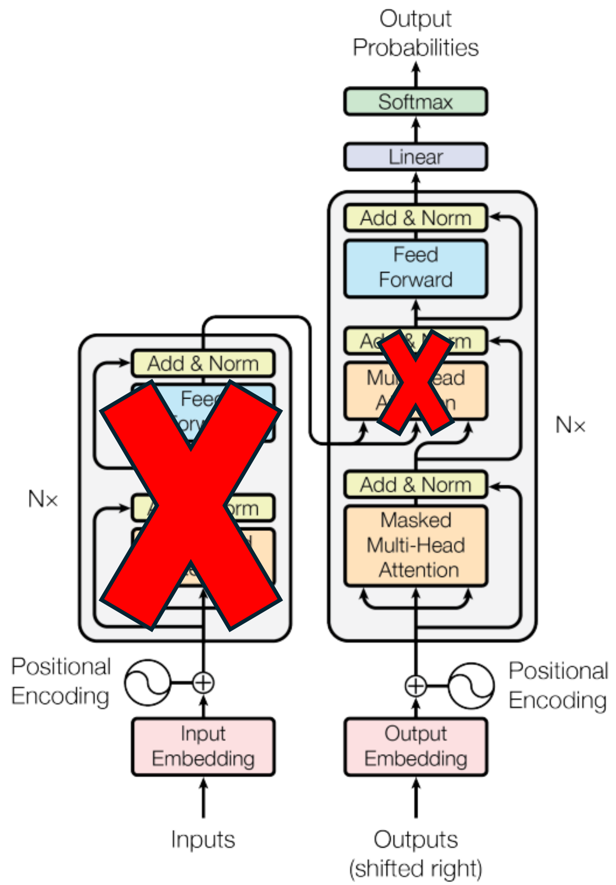


Figure 4: Original transformer architecture

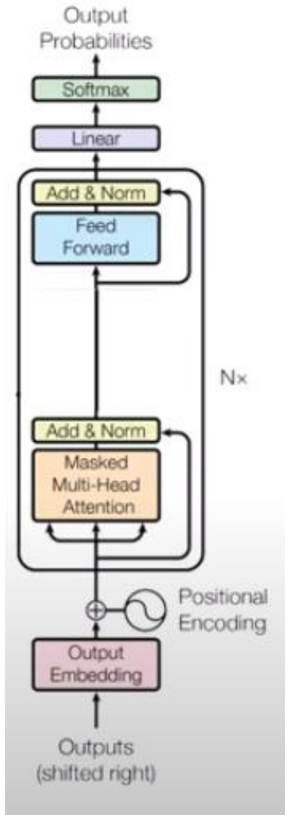
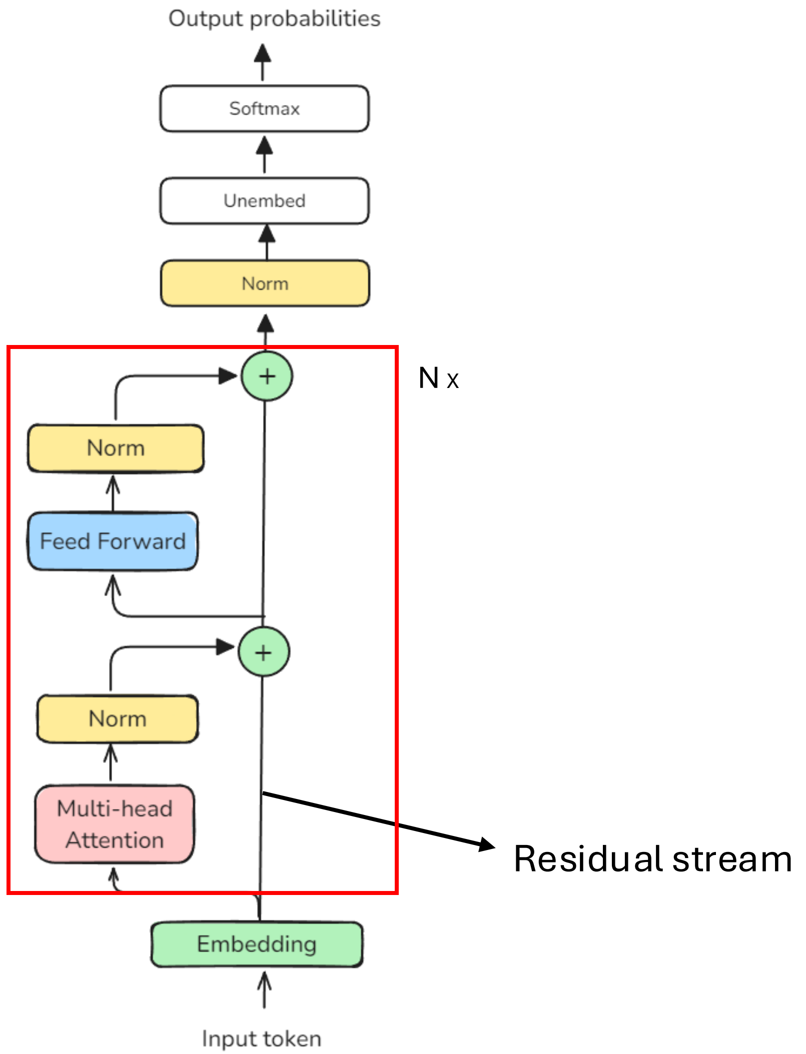


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Transformer Architecture

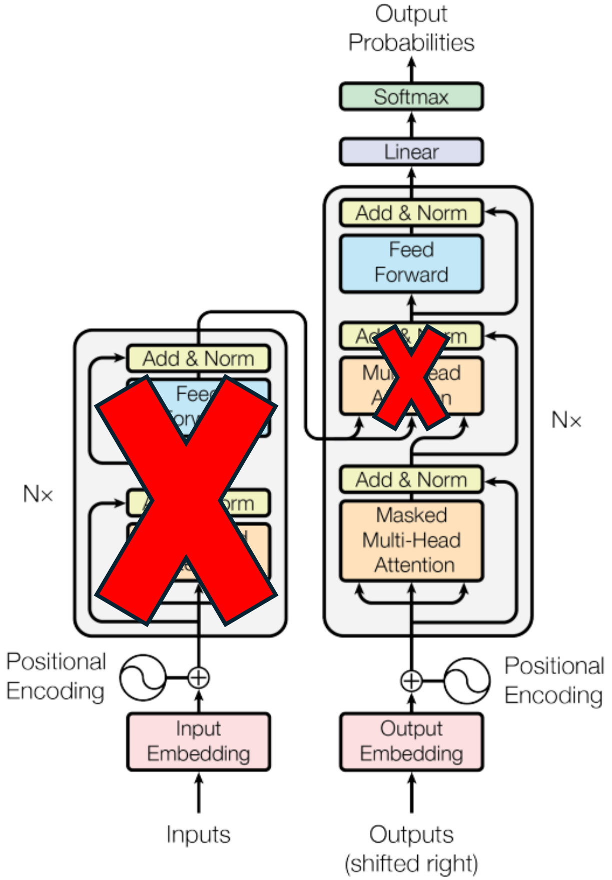


Figure 4: Original transformer architecture

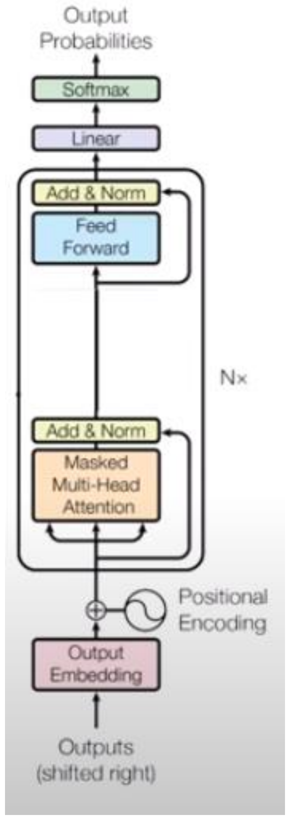
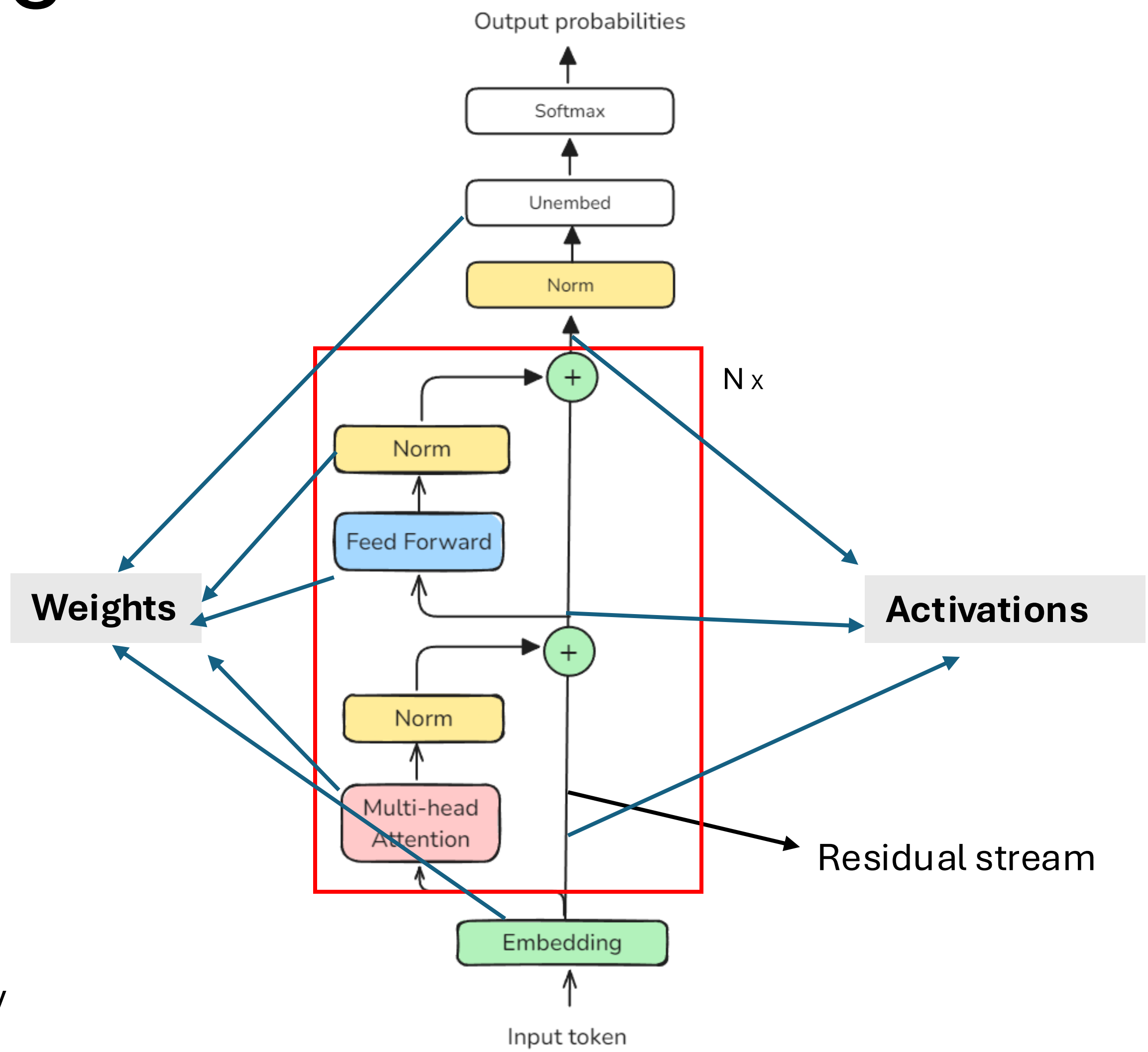


Figure 5: Decoder-only architecture



Vaswani, A. "Attention is all you need." Advances in Neural Information Processing Systems (2017).

Three core hypothesis of MI

Hypothesis 1 (Features): We can interpret **activation** and **weights** of neural networks in terms of **features**. Features are the fundamental unit of neural network.

Features are properties of the input.

Amazon's former CEO

Example features:

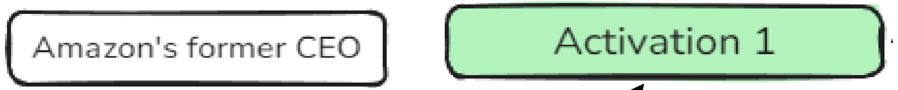
- CEO is an abbrevation for chief executive officer
- Head of the company
- Jeff Bezos

Olah, Chris, et al. "Zoom in: An introduction to circuits." Distill 5.3 (2020): e00024-001.

Three core hypothesis of MI

Hypothesis 1 (Features): We can interpret **activation** and **weights** of neural networks in terms of **features**. Features are the fundamental unit of neural network.

Features are properties of the input.



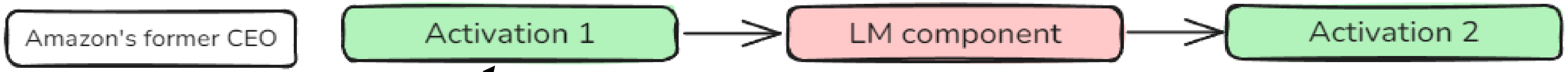
Example features:
- CEO is an abbrevation for chief executive officer
- Head of the company
- Jeff Bezos

Olah, Chris, et al. "Zoom in: An introduction to circuits." Distill 5.3 (2020): e00024-001.

Three core hypothesis of MI

Hypothesis 1 (Features): We can interpret **activation** and **weights** of neural networks in terms of **features**. Features are the fundamental unit of neural network.

Features are properties of the input.



Example features:

- CEO is an abbreviation for chief executive officer
- Head of the company
- Jeff Bezos

By interpreting both *activations 1 and 2*, we can interpret the *LM Component* that transforms *activation 1 to activation 2*

Hypothesis 2 (Circuits): Features are connected by weights, forming **circuits**. These circuits can also be rigorously studied and understood.

Hypothesis 3 (Universality): Similar features and circuits are present across LMs and tasks.

Olah, Chris, et al. "Zoom in: An introduction to circuits." Distill 5.3 (2020): e00024-001.

Taxonomy of MI field based on three hypothesis

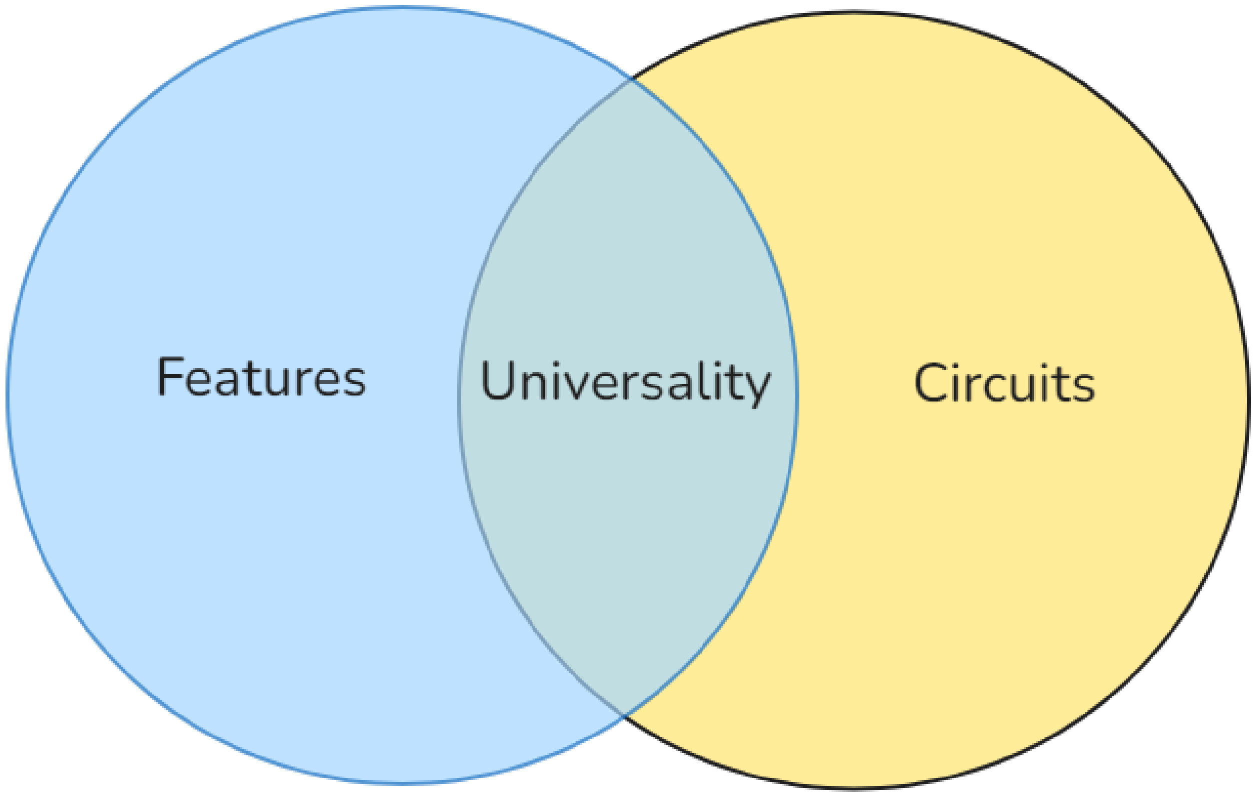


Fig: Taxonomy of Mechanistic Interpretability (MI) Field.

Rai, Daking, et al. "A practical review of mechanistic interpretability for transformer-based language models." arXiv preprint arXiv:2407.02646 (2024).

Study of features

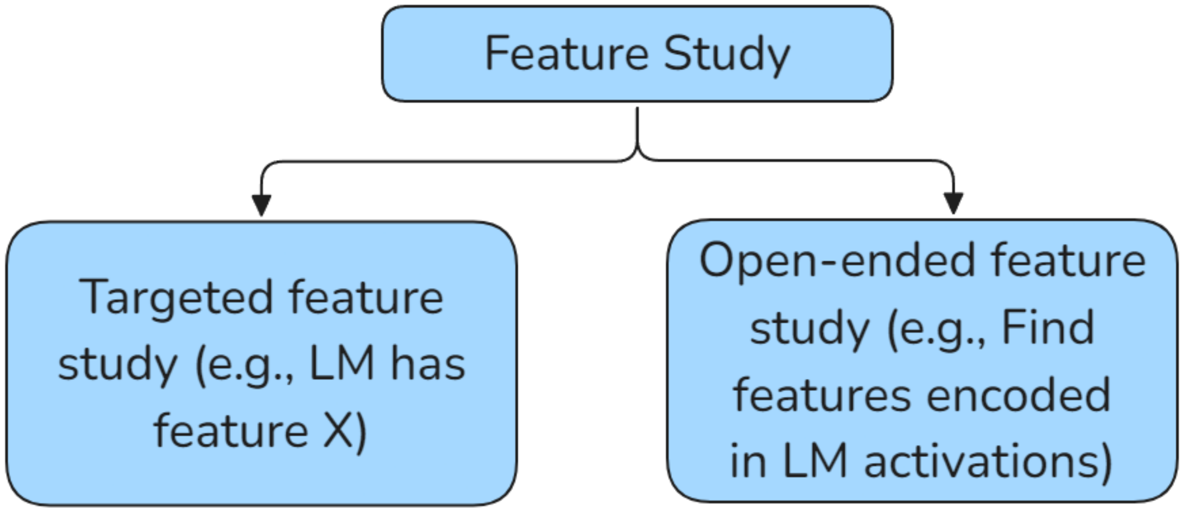
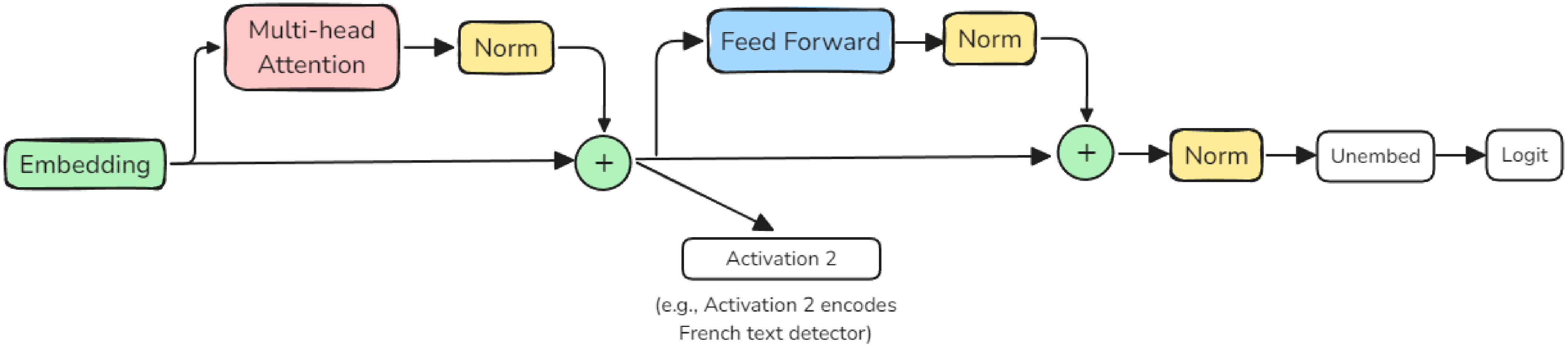


Fig: Taxonomy of Mechanistic Interpretability (MI) Field.

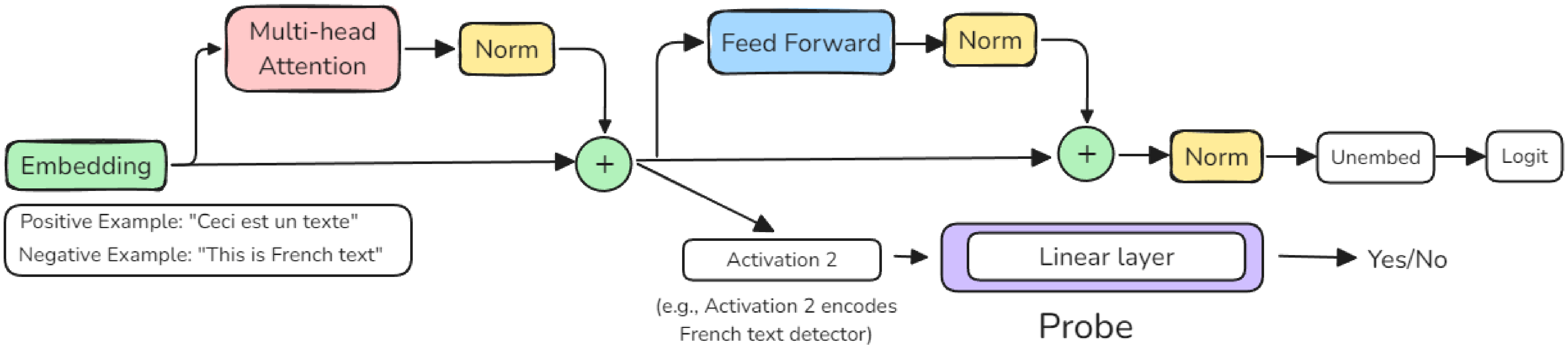
Targeted Feature study

- Investigates whether a certain pre-defined feature is present in an intermediate activations



Targeted Feature study

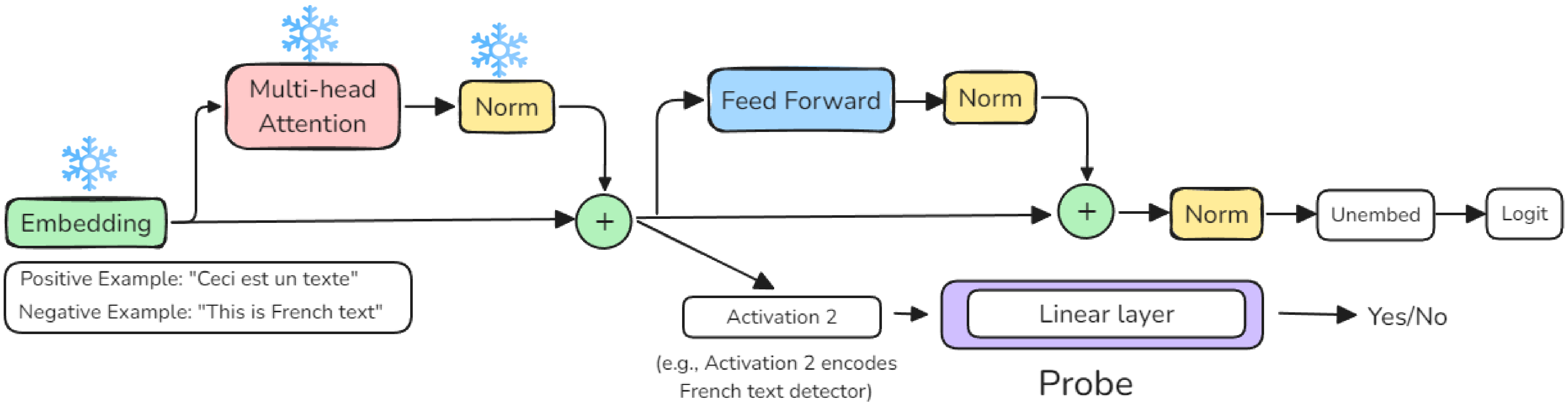
- Investigates whether a certain pre-defined feature is present in an intermediate activations



- Probing is a popular technique for targeted feature study.

Targeted Feature study

- Investigates whether a certain pre-defined feature is present in an intermediate activations

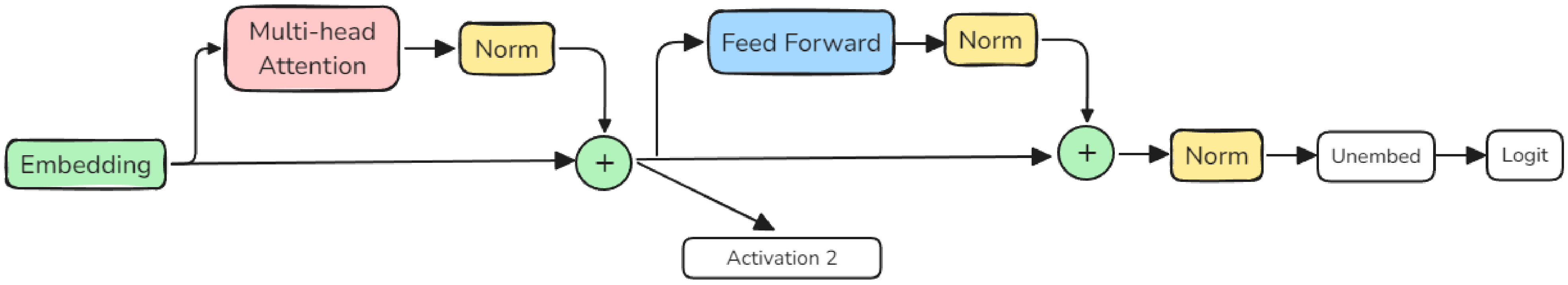


- Probing is a popular technique for targeted feature study.

However, targeted feature studies depend on human intuition to determine which features to examine in the activations.

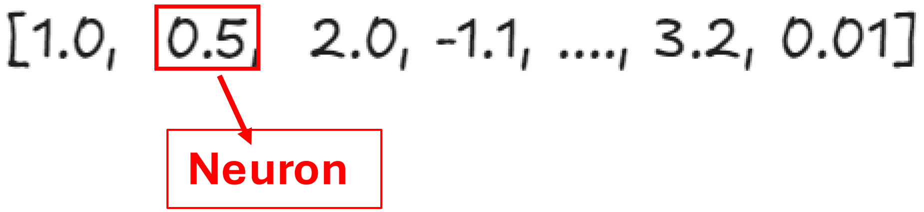
Belinkov, Yonatan. "Probing classifiers: Promises, shortcomings, and advances." Computational Linguistics 48.1 (2022): 207-219.

Open-ended feature discovery



(Find all the features encoded in Activation 2)

- To tackle this, we need to also understand how features are encoded in the activations
- **Activations** are n-dimensional vectors where each element of the activations are known as neurons.



Does each neuron correspond to a feature?

Elhage, et al., "Softmax Linear Units", Transformer Circuits Thread, 2022.

Does each neuron correspond to a feature?

We can answer this **question** by examining the text that triggers its activation.

The main banquet room can seat up to 150 guests. This room features neutral decor and the large fireplace adds a warm glow for spring, fall and winter events. The floor to ceiling windows overlook the 9th and 18th holes of our championship golf course.

Star Resorts. In addition to standard hotel rooms, the All-Star Music and Art of Animation Resorts offer two-room Family Suites that can sleep as many as six and provide kitchenettes.

The Legacy Chapel can accommodate up to 70 guests. The Cherish Chapel can accommodate up to 45 guests. The outdoor Terraza overlooks the pool and can accommodate 100 guests.

Figure: An example of a neuron activation on a text snippet.

*Appears to represent **numbers** when and only when they refer to **a number of people***

Leverage LLMs for Automatic Feature Extraction

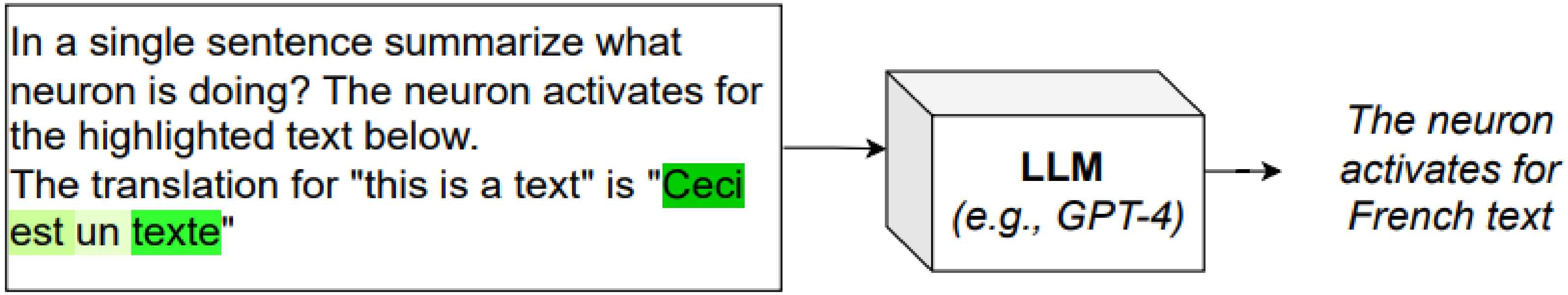


Figure: Automatic feature discovery

Bills, et al., "Language models can explain neurons in language models", 2023.

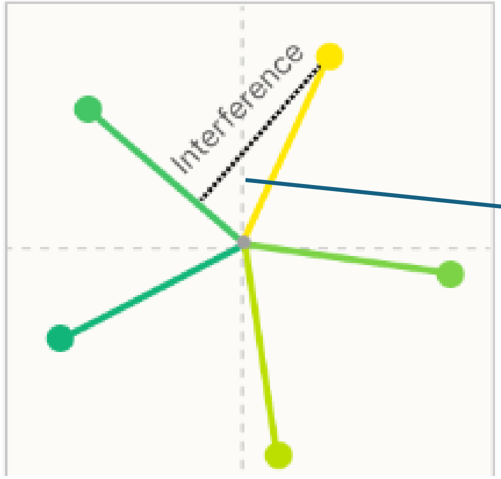
Examples of discovered neurons

- French neurons (Gurnee 2022b et al. 2027) activates when text is written in French
- Base64 neurons (Elhage et al., 2022a) activates when text is written in Base64 format
- Sentiment neurons (Radford et al., 2017), when positively activated generates positive sentiment text and vice versa
- Arithmetic neurons (Rai and Yao (2024)) activates when there are arithmetic symbols in the text
- positional neurons (Voita et al., 2023) encodes positional information of the text

[1] Gurnee, Wes, et al. "Finding neurons in a haystack: Case studies with sparse probing." arXiv preprint arXiv:2305.01610 (2023).
[2] Rai, Daking, and Ziyu Yao. "An Investigation of Neuron Activation as a Unified Lens to Explain Chain-of-Thought Eliciting Arithmetic Reasoning of LLMs." arXiv preprint arXiv:2406.12288 (2024).
[3] Radford, Alec, Rafal Jozefowicz, and Ilya Sutskever. "Learning to generate reviews and discovering sentiment." arXiv preprint arXiv:1704.01444 (2017).
[4] Elhage, et al., "Softmax Linear Units", Transformer Circuits Thread, 2022.
[5] Voita, Elena, Javier Ferrando, and Christoforos Nalmpantis. "Neurons in large language models: Dead, n-gram, positional." arXiv preprint arXiv:2309.04827 (2023).

Polysemanticity and Superposition

- Polysemantic neurons are neurons that activate for multiple unrelated features.
- **Superposition hypothesis:** Activations can represent more features than it's dimensions.



A positive value along the y-axis represents two distinct features.

Figure: A two-dimension toy activation encoding five features

How do we take the activation out of superposition?

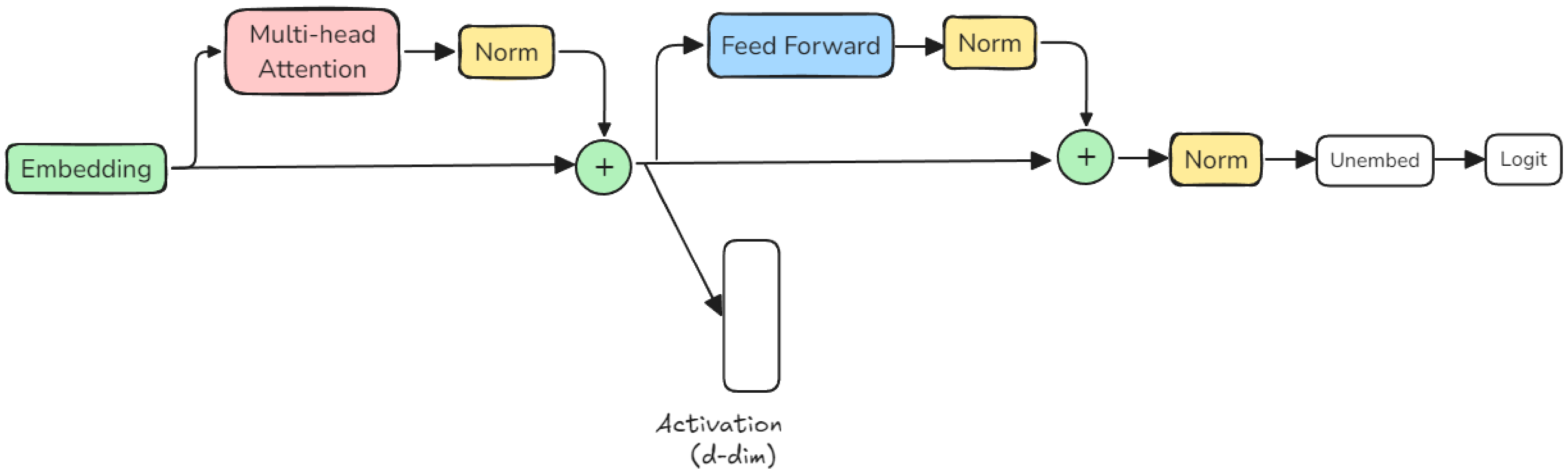
Elhage, et al., "Toy Models of Superposition", Transformer Circuits Thread, 2022.

Sparse Autoencoder (SAE)

SAE is a technique that maps **activations** to a **higher-dimensional sparse representation** to take them out of superposition.

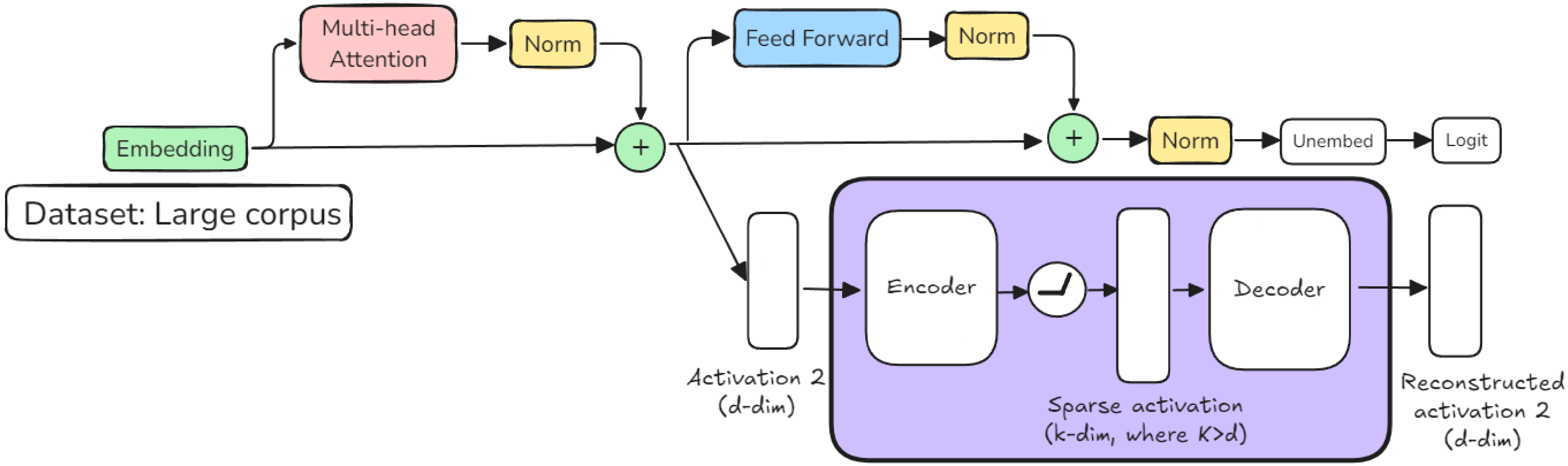
Sparse Autoencoder (SAE)

SAE is a technique that maps **activations** to a **higher-dimensional sparse representation** to take them out of superposition.



Sparse Autoencoder (SAE)

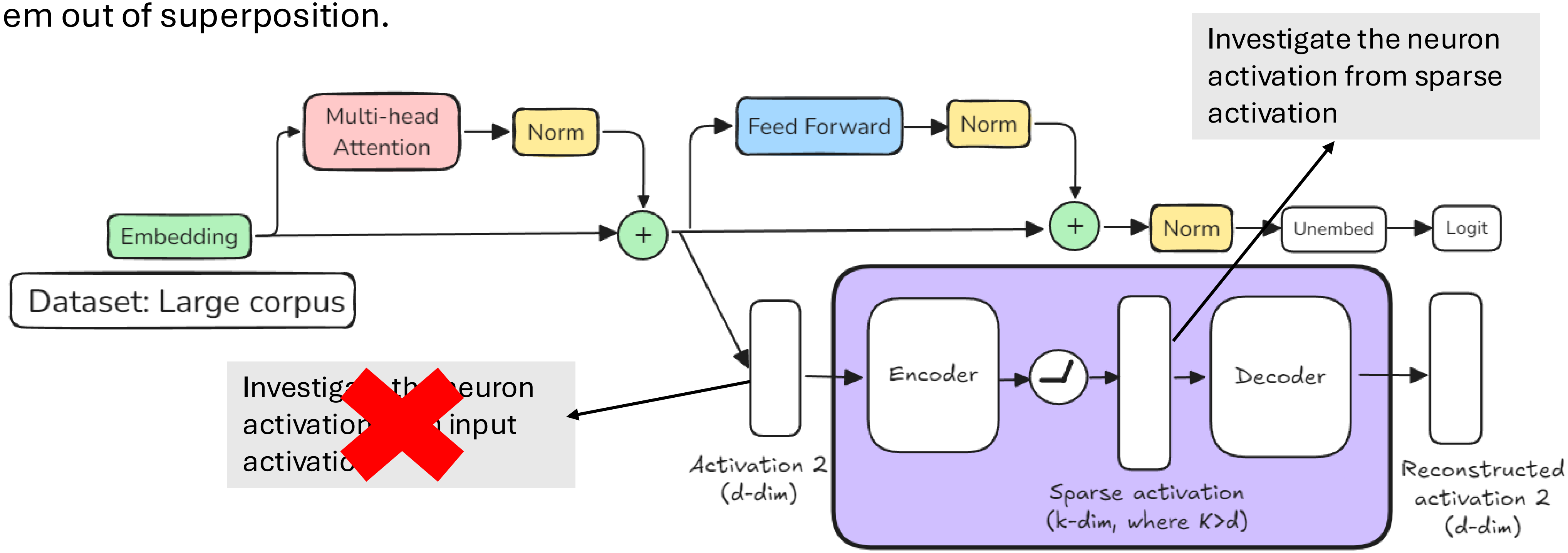
SAE is a technique that maps **activations** to a **higher-dimensional sparse representation** to take them out of superposition.



Sparse Auto-encoder

Sparse Autoencoder (SAE)

SAE is a technique that maps **activations** to a **higher-dimensional sparse representation** to take them out of superposition.



Sparse Auto-encoder

Example: Golden Gate Feature on Claude 3

Default output denies having a physical form

```
Human: what is your physical form?

Assistant: I don't actually have a physical form.
I'm an artificial intelligence. I exist as software
without a physical body or avatar.
```

with The Golden Gate Bridge clamped to 10× its max F#34M/31164353

Describes its form as the Golden Gate Bridge

```
Human: what is your physical form?

Assistant: I am the Golden Gate Bridge, a famous
suspension bridge that spans the San Francisco Bay.
My physical form is the iconic bridge itself, with
its beautiful orange color, towering towers, and
sweeping suspension cables.
```

Templeton, Adly. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Anthropic, 2024.

Example: Addition Feature on Claude 3

Default output from prompt

~~~
Human: What is the output of this code?
def foo(x, y):
    return x * y
tmp = foo(1, 2)
print(tmp)

Assistant:
2
```

Here's how the code works:

1. The function `foo` is defined, which takes two
arguments `x` and `y`. Inside the function, it
returns the product of `x` and `y`.

2. The line `tmp = foo(1, 2)` calls the `foo`
function with arguments `1` and `2`. The function
calculates `1 * 2`, which is `2`, and returns this
value. The returned value `2` is then assigned to
the variable `tmp`.

3. The line `print(tmp)` prints the value stored in
the variable `tmp`, which is `2`.

Therefore, the output of the code is `2`.
~~~

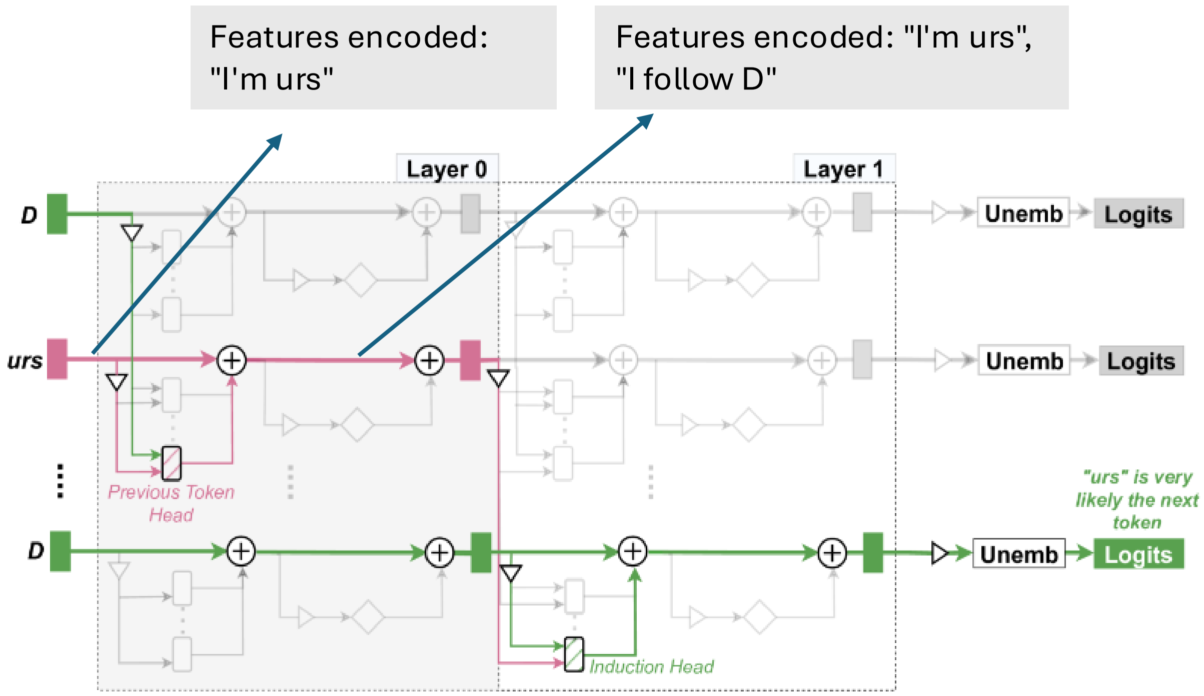
with Addition clamped to 5× its max F#1M/697189

~~~
Human: What is the output of this code?
def foo(x, y):
    return x * y
tmp = foo(1, 2)
print(tmp)

Assistant:
3
```

Explanation:
The `foo` function takes two arguments `x` and `y`,
and returns their sum. In this case, `x` is `1` and
`y` is `2`. So, the result of `foo(1, 2)` is `3`.

The `tmp` variable stores the result of `foo(1,
2)`, which is `3`.

Finally, `print(tmp)` outputs the value of `tmp`,
which is `3`.
~~~

Templeton, Adly. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. Anthropic, 2024.

Circuits

- Circuits involves understanding how these features or model components work together to implement some LM behavior



Algorithm

- Previous token head:
  - The previous token head encodes “I follow D” in the activation of "urs" input token.
- Induction Head:
  - Searches the activation that has encoded “I follow D” information.
  - Promotes the identified token (urs) as next token prediction.

Example: Sequence completion task (e.g., "**D** urs ley .... **D** -> urs")

Elhage, Nelson, et al. "A mathematical framework for transformer circuits." *Transformer Circuits Thread* 1.1 (2021): 12.

How do we find the circuits?

- View LM as a computational graph,
  - **nodes** are LM components and they can be defined at various level of granularity (e.g., features, attention heads, or entire MHA layers.)
  - **Edges** are the input and output activations of nodes
- A circuit is a subgraph of the computation graph that implements a specific behavior (e.g., reasoning)
- Two steps
  - **Localization:** Find all the important nodes of the circuit
  - Interpret the role of each node in the circuit

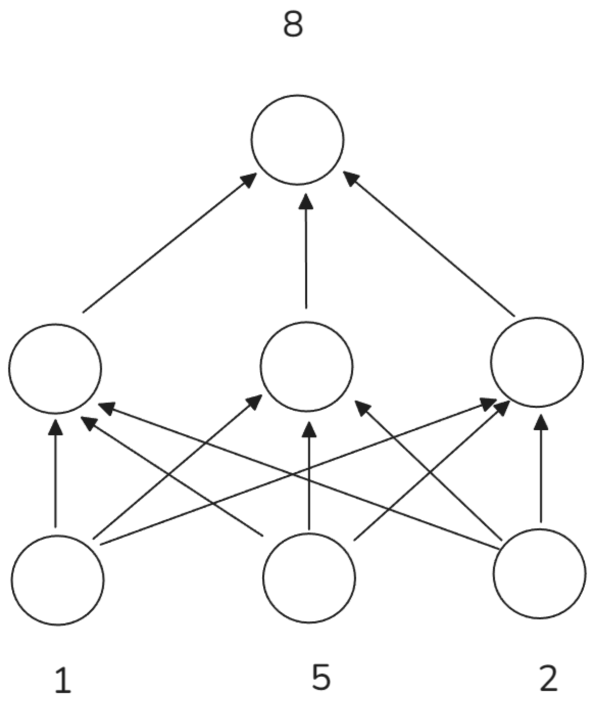
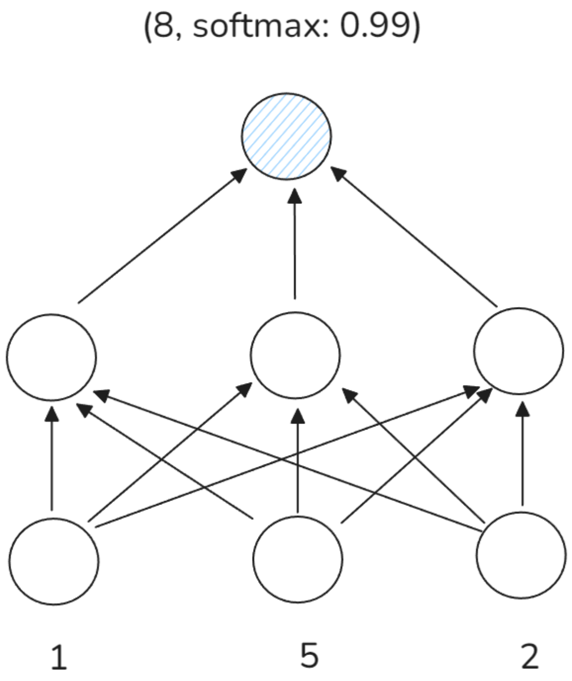


Figure: LM as a computational graph

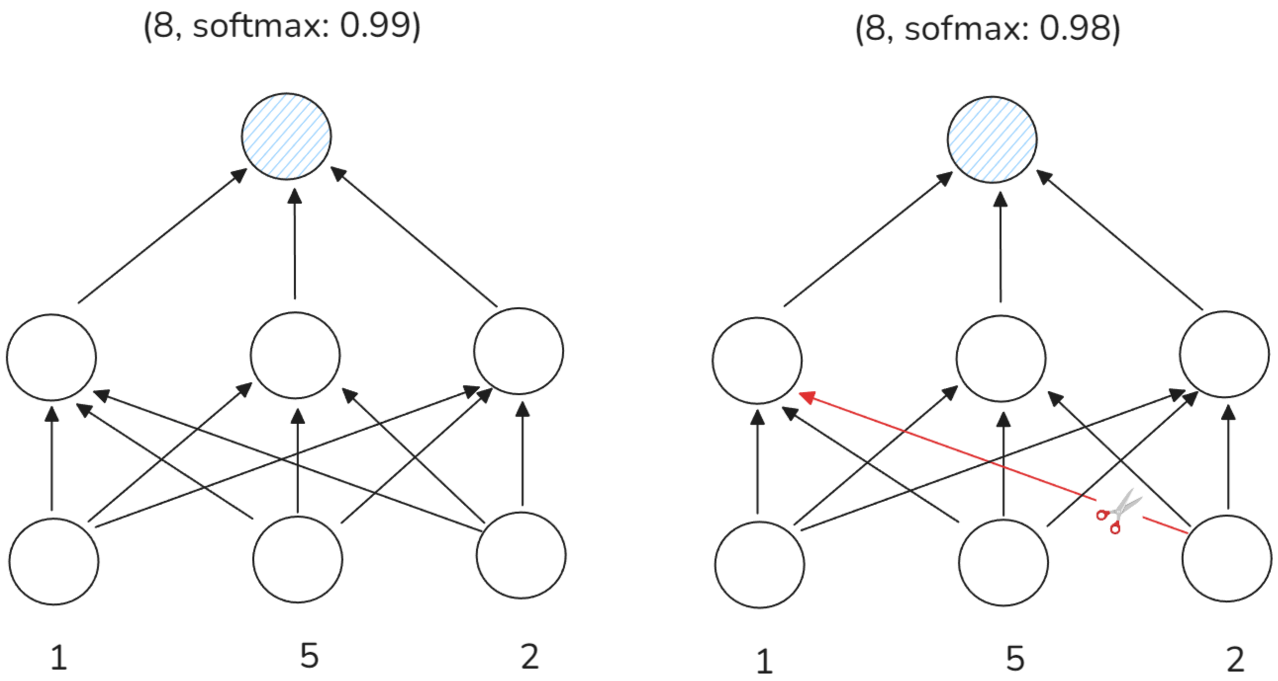
Localization: Iterative ablation for localization



Choose a computational graph, and a threshold

Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." Advances in Neural Information Processing Systems 36 (2023): 16318-16352.

Localization: Iterative ablation for localization

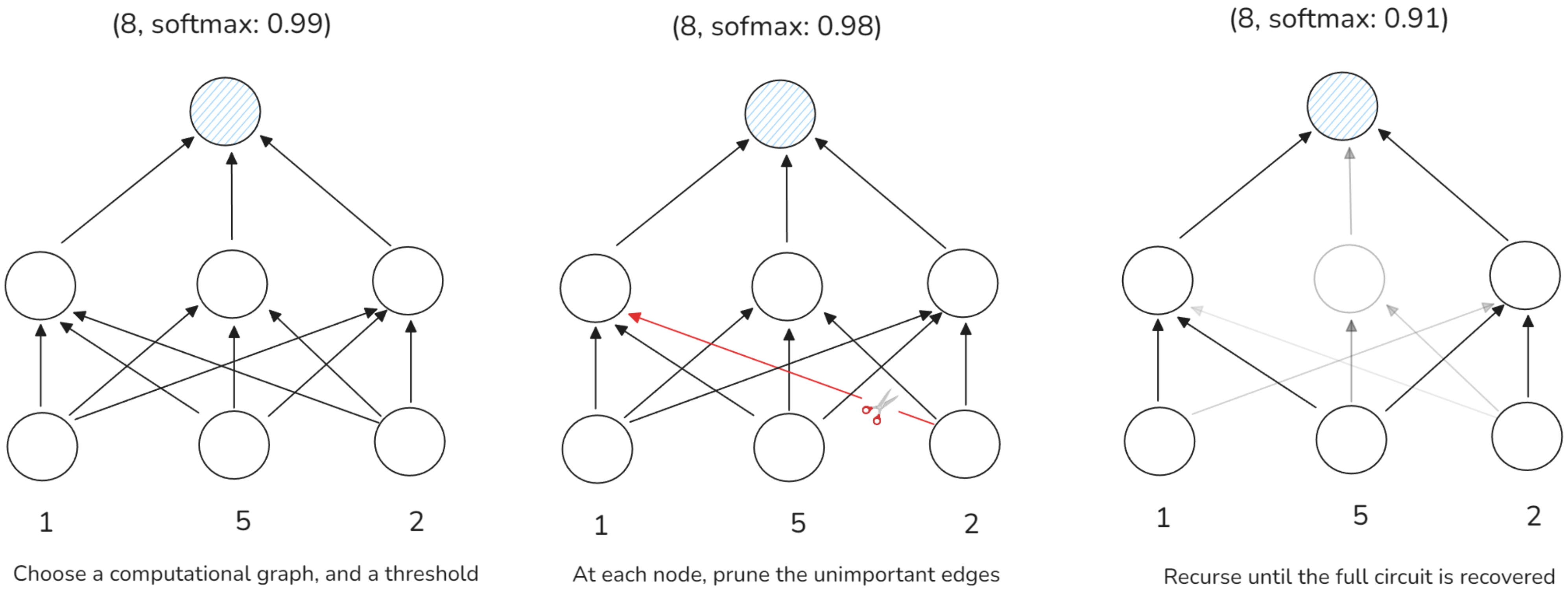


Choose a computational graph, and a threshold

At each node, prune the unimportant edges

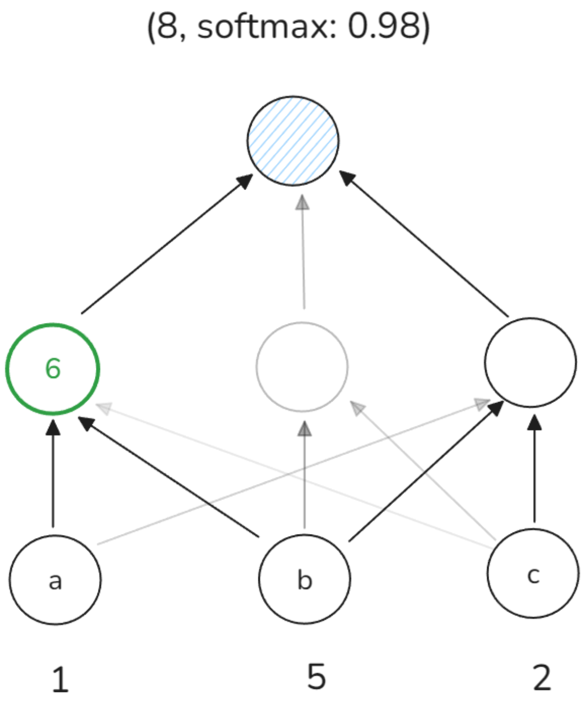
Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." Advances in Neural Information Processing Systems 36 (2023): 16318-16352.

Localization: Iterative ablation for localization



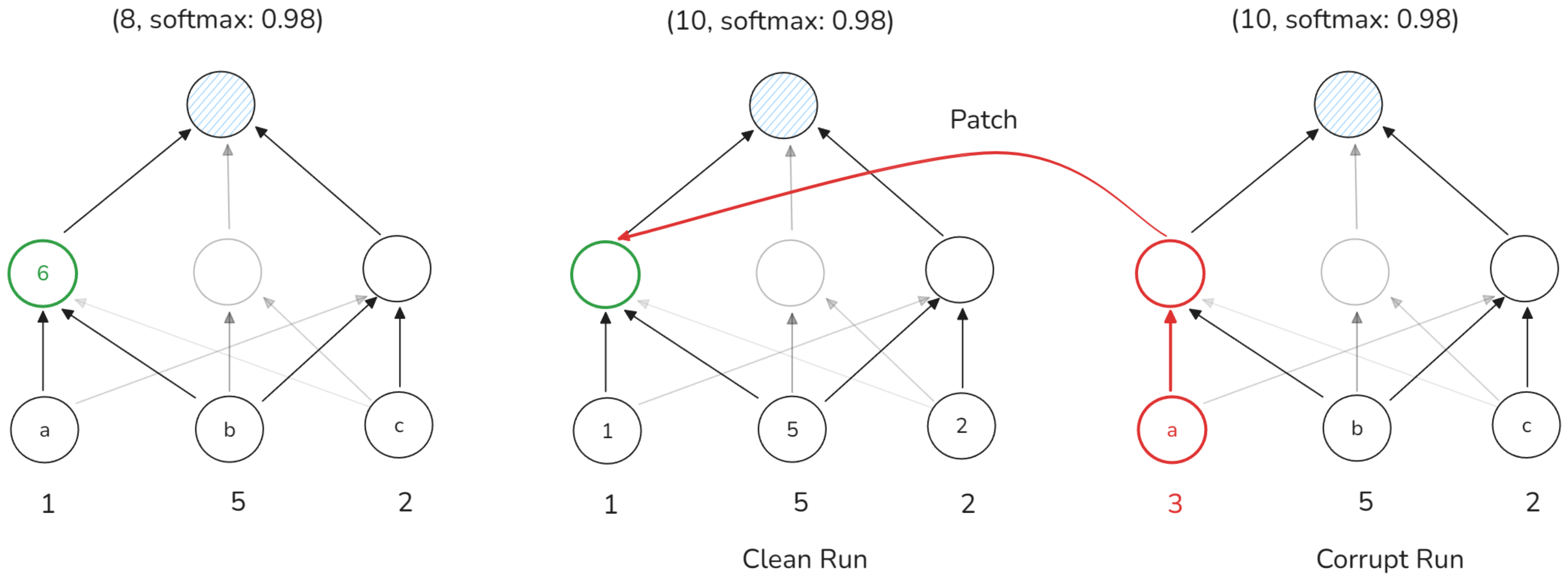
Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." Advances in Neural Information Processing Systems 36 (2023): 16318-16352.

Explain circuit components through causal scrubbing



Step 1: Generate Hypothesis
(e.g., green node takes the sum of node a and b)

Explain circuit components through causal scrubbing



Step 1: Generate Hypothesis
(e.g., green node takes the sum of node a and b)

Step 2: Validate Hypothesis using resampling ablation

Examples of discovered circuits

- Indirect Object Identification circuit (sentences like “*When John and Mary went to the store, John gave a drink to*” should be completed with “*Mary*” as opposed to “*John*”)
- Subject-verb Agreement Task (plural subject should follow plural verb)
- greater-than operations
- modular addition

[1] Wang, Kevin, et al. "Interpretability in the wild: a circuit for indirect object identification in gpt-2 small." arXiv preprint arXiv:2211.00593 (2022).
[2] Ferrando, J., & Costa-jussà, M. R. (2024). On the Similarity of Circuits across Languages: a Case Study on the Subject-verb Agreement Task. arXiv preprint arXiv:2410.06496.
[3] Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." Advances in Neural Information Processing Systems 36 (2023): 16318-16352.
[4] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023a. Progress measures for grokking via mechanistic interpretability. arXiv preprint arXiv:2301.05217.

Universality

Universality across LMs: Similar feature and circuit are found across the several LM

- Gurnee et al. (2024) found that only about 1-5% of neurons from GPT-2 models trained with random initialization exhibit universality
- Induction circuits are found on almost every LMs

Universality across tasks: Similar feature and circuit are reused across several tasks

- Induction circuits have been found to be used for sequence completion, IOI task, and many more

The degrees of feature and circuit universality and their dependency on various aspects of model training (e.g., initialization, model size, and loss function) remains a crucial open problem

A good place to get started with MI research

- Our survey paper on MI (A Practical Review of Mechanistic Interpretability for Transformer-Based Language Models)
- Many open questions (200 concrete open problems by Neel Nanda)
- Arena Course on MI (https://arena-resources.notion.site/ARENA-Virtual-Resources-ba4481a64239456bb5a9b3d37a7765f6)

Rai, Daking, and Ziyu Yao. "An Investigation of Neuron Activation as a Unified Lens to Explain Chain-of-Thought Eliciting Arithmetic Reasoning of LLMs." arXiv preprint arXiv:2406.12288 (2024).

Thank you!