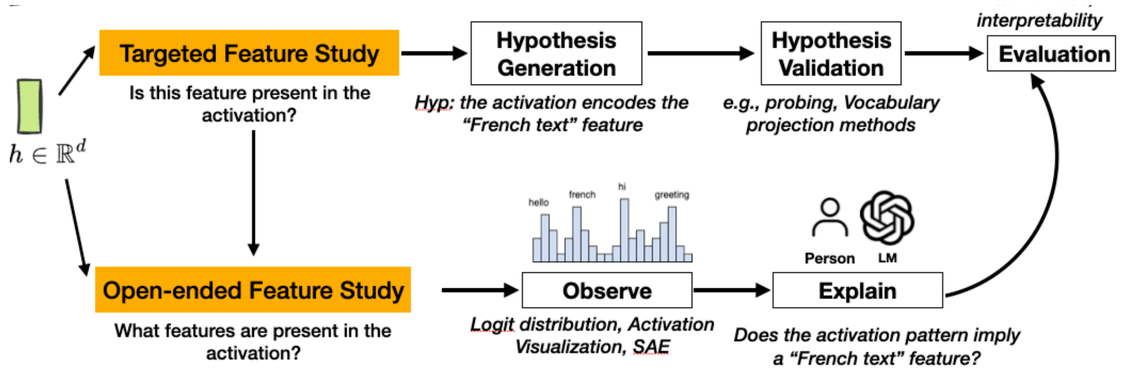
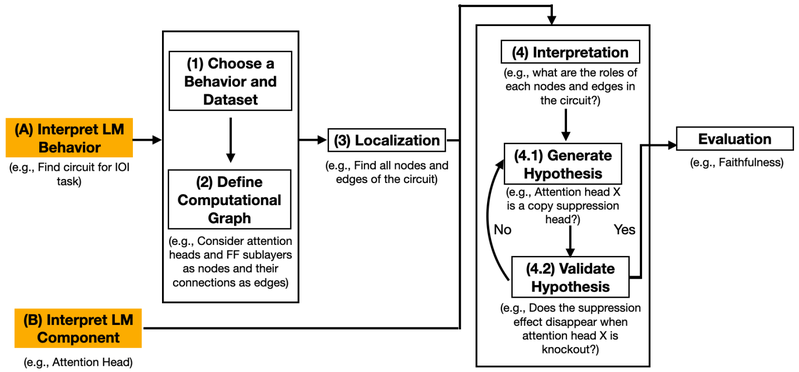
Part 2:
MI Practices and Techniques

Preview of Part 2

- Before diving deep into the technical details…
- How have people been applying MI to interpret LMs?
  - Introducing a **Task-centric Beginner's Roadmap to MI**





- Interleaving with the Roadmap presentation, we will look into **common methodologies** used in the specific MI workflows
  - What they are, and how people have been improving or applying them

Part 2.1 Feature Study

Overview of Workflows for Feature Study

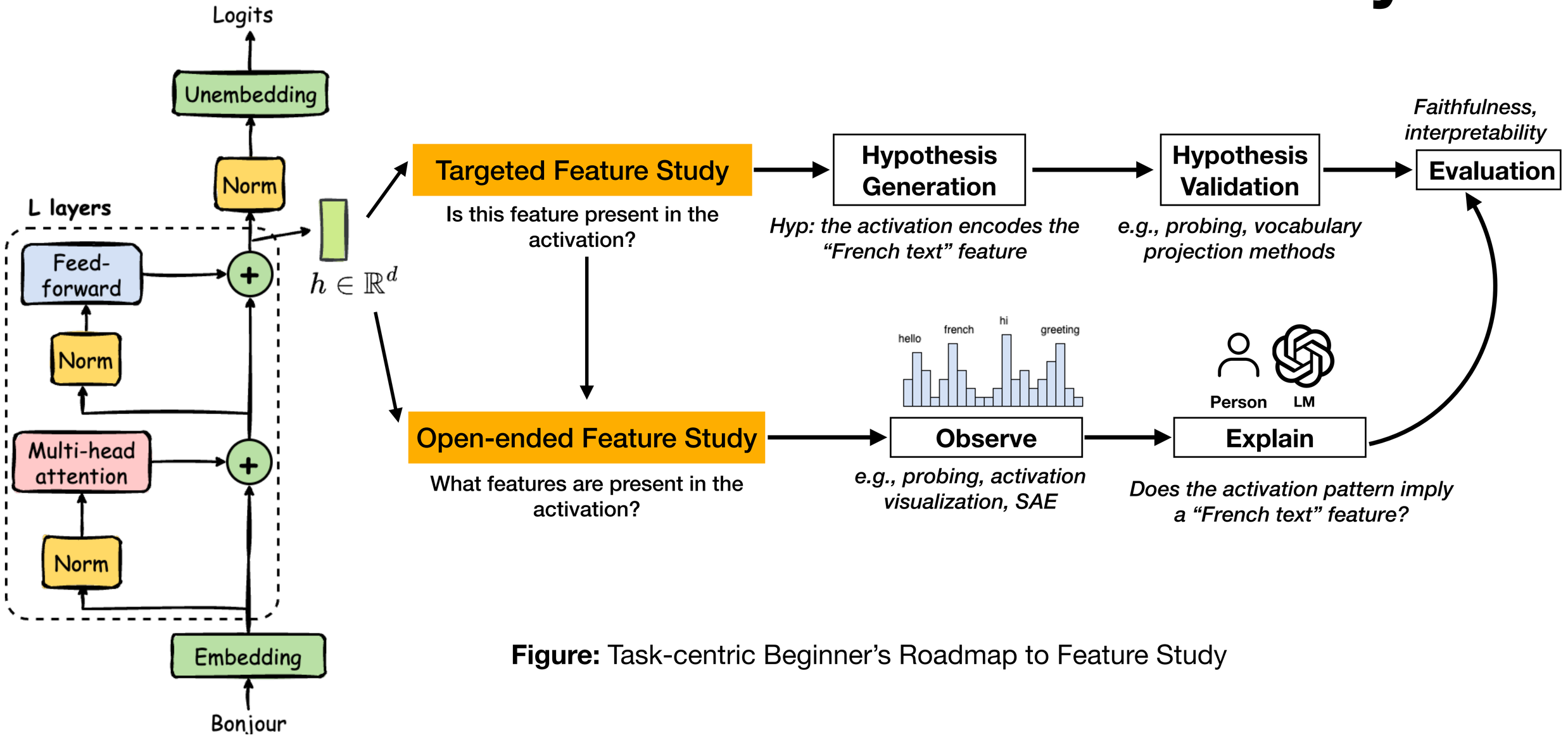
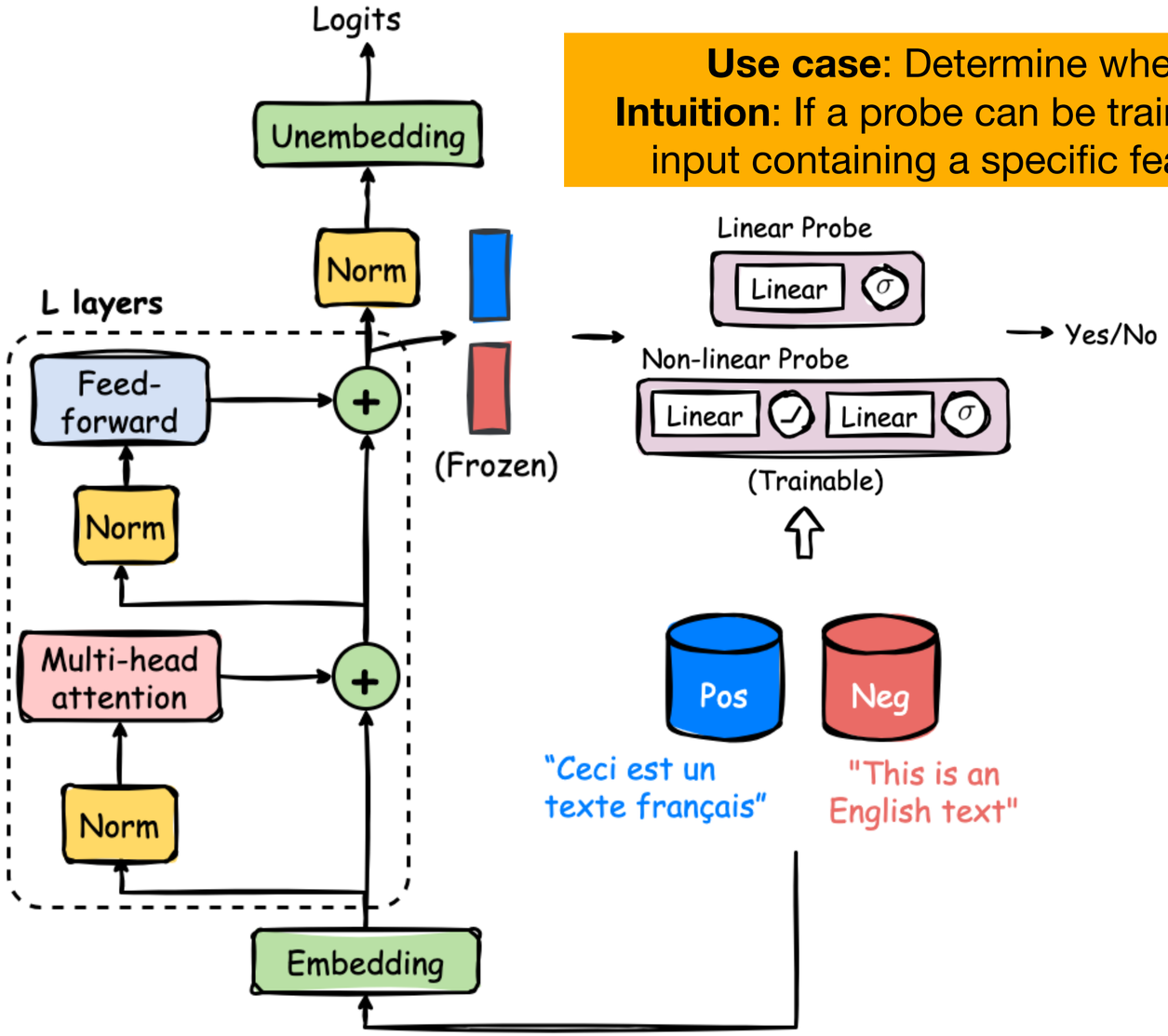


Figure: Task-centric Beginner's Roadmap to Feature Study

Techniques for Targeted Feature Study

Method 1: Probing



Use case: Determine whether a pre-defined feature is encoded in an activation
Intuition: If a probe can be trained to distinguish whether an activation originates from an input containing a specific feature, it suggest that the activation encodes that feature.

Principle: A **high probing accuracy** on a held-out test set indicates the presence of feature

Issue 1 (suspicious to "false positive"): A powerful probe may learn to encode the feature on its own instead of verifying presence of feature in the activation

Solution: Linear Probes + Control Baseline (Hewitt&Liang19; Belinkov+22)

Issue 2 (presence vs. usage): Probing only confirms the **presence** of the feature in the representation, but not its **usage** by the model (Belinkov, 2022).

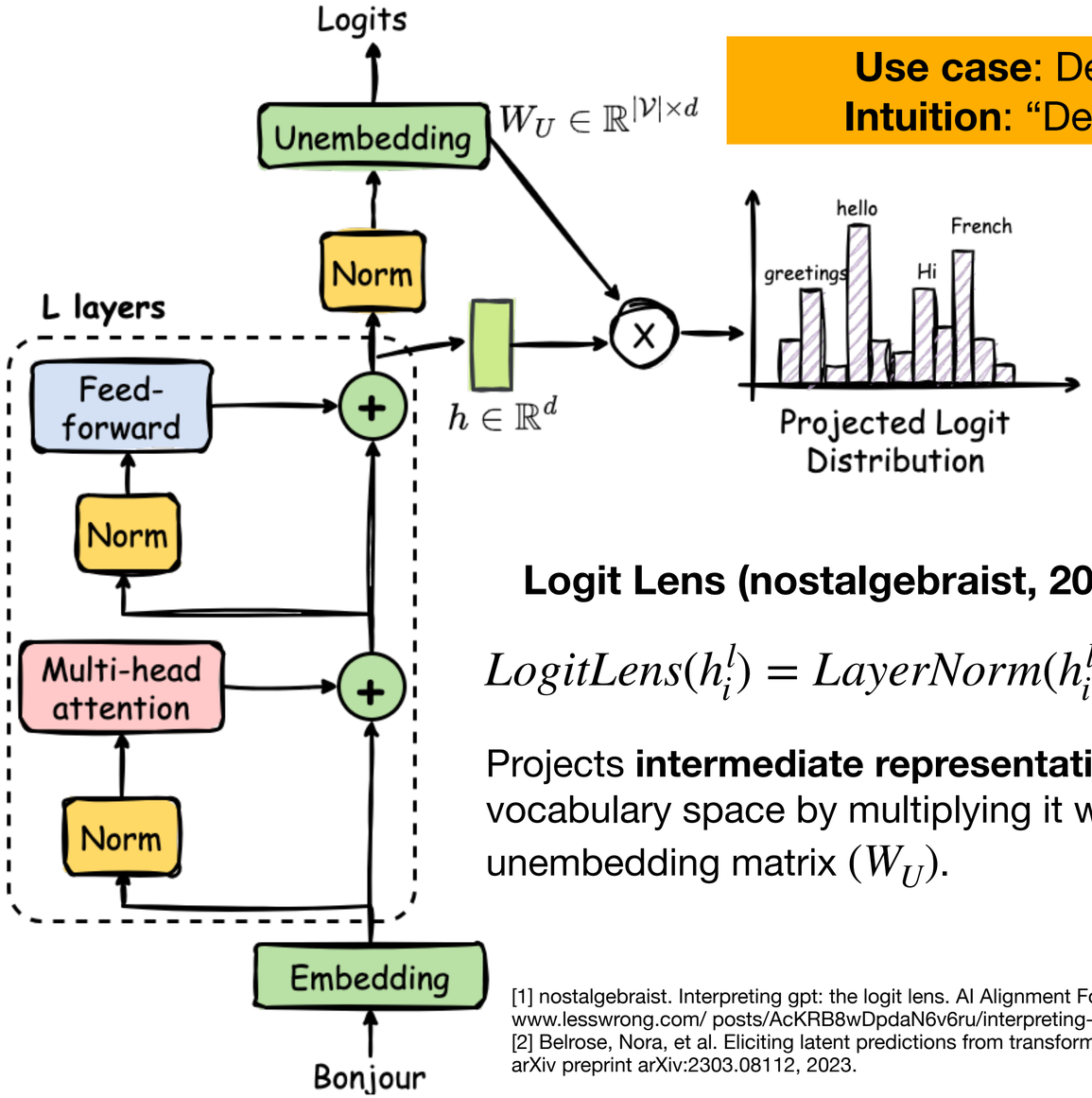
Solution: Intervention-based Techniques

Issue 3 (requiring additional resources): Need extra data generation and model training.

Solution: Patchscope (Ghandeharioun+24)

[1] Hewitt, John, and Percy Liang. "Designing and Interpreting Probes with Control Tasks." *EMNLP* 2019.
[2] Belinkov, Yonatan. "Probing classifiers: Promises, shortcomings, and advances." *Computational Linguistics* 48.1 (2022): 207-219.
[3] Ghandeharioun, Asma, et al. "Patchscopes: a unifying framework for inspecting hidden representations of language models." *ICML* 2024.

Method 2: Vocabulary Projection



Use case: Decode the next token predictions encoded in the activation.
Intuition: "Decode" an activation by projecting it to the vocabulary space

Explain the Pattern

The activation says about the translation of "Bonjour" in English

Logit Lens (nostalgebraist, 2020)

$$LogitLens(h_i^l) = LayerNorm(h_i^l)\, W_U^\top$$

Projects **intermediate representation** into vocabulary space by multiplying it with the unembedding matrix (W_U).

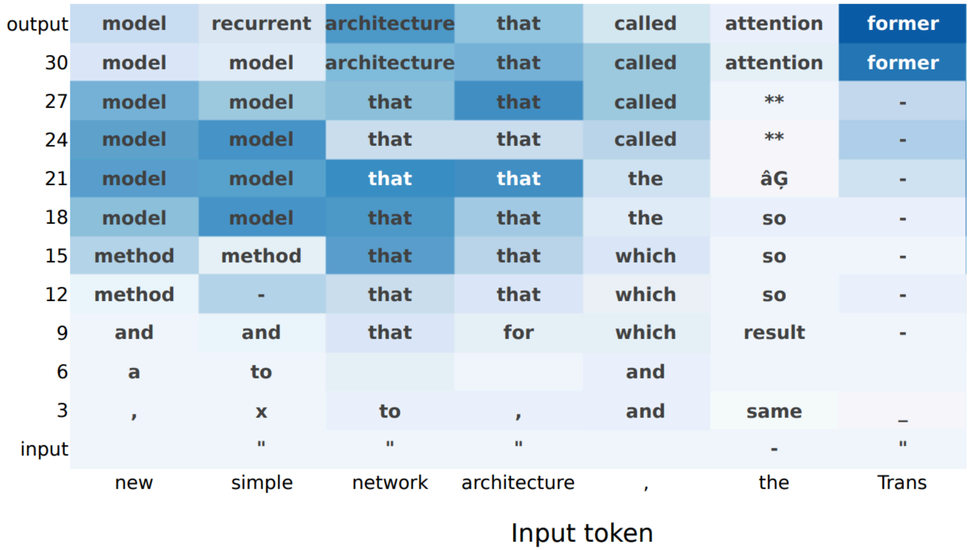
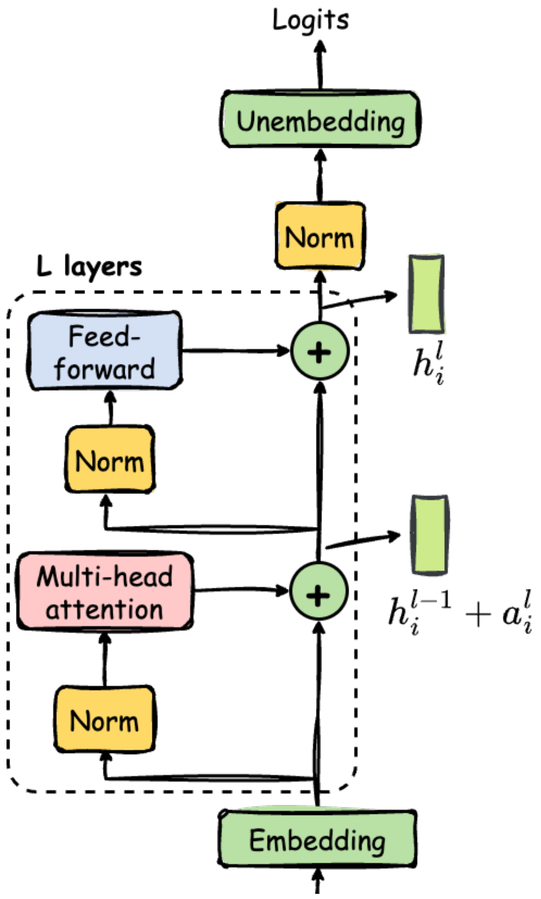


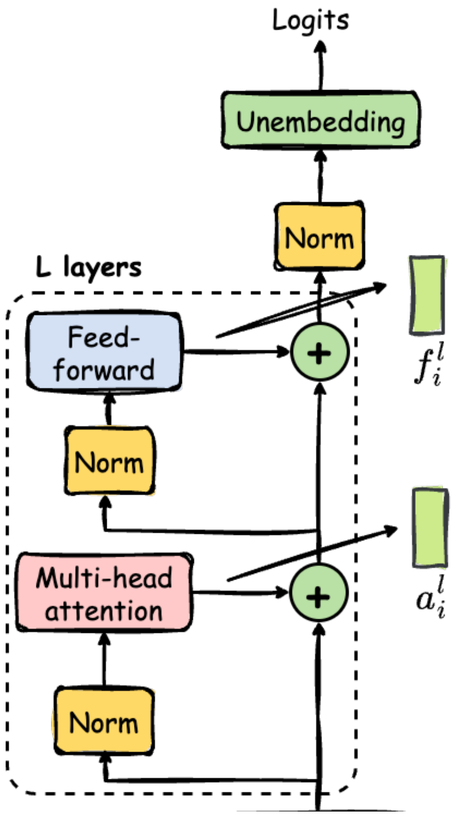
Figure: Logit lens reveal how LM refines prediction by layers (Belrose+23)

[1] nostalgebraist. Interpreting gpt: the logit lens. AI Alignment Forum, 2020. https://www.lesswrong.com/ posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.
[2] Belrose, Nora, et al. Eliciting latent predictions from transformers with the tuned lens. arXiv preprint arXiv:2303.08112, 2023.

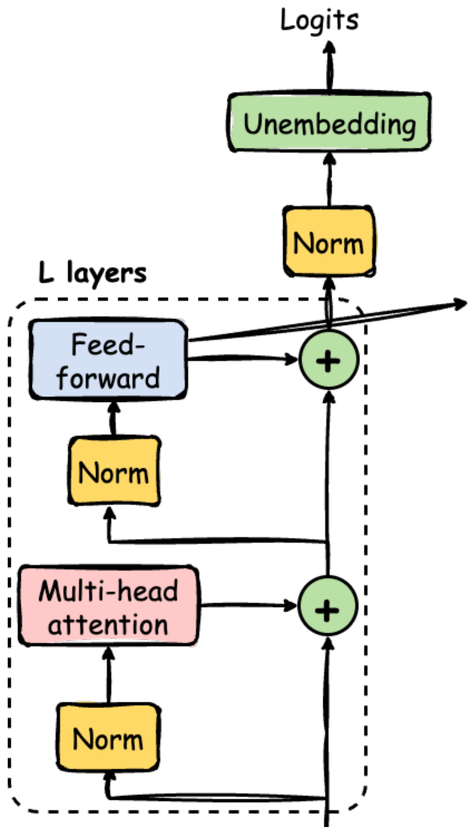
Method 2: Vocabulary Projection



Residual Stream

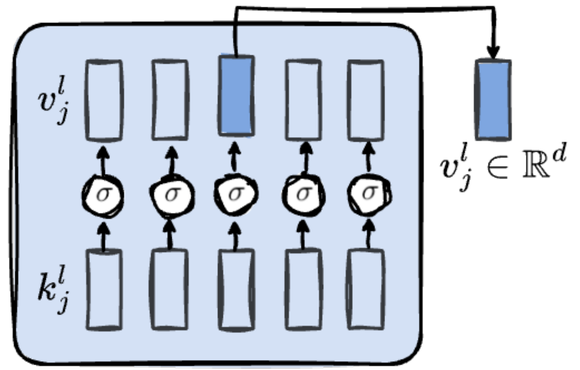


Output of MHA and FF



Feed-Forward (FF)

$$f_i^l = \underbrace{f\left((h_i^{l-1} + a_l^i) \cdot W_k^l\right)}_{\textit{Key}} \cdot \underbrace{W_v^l}_{\textit{Value}}$$

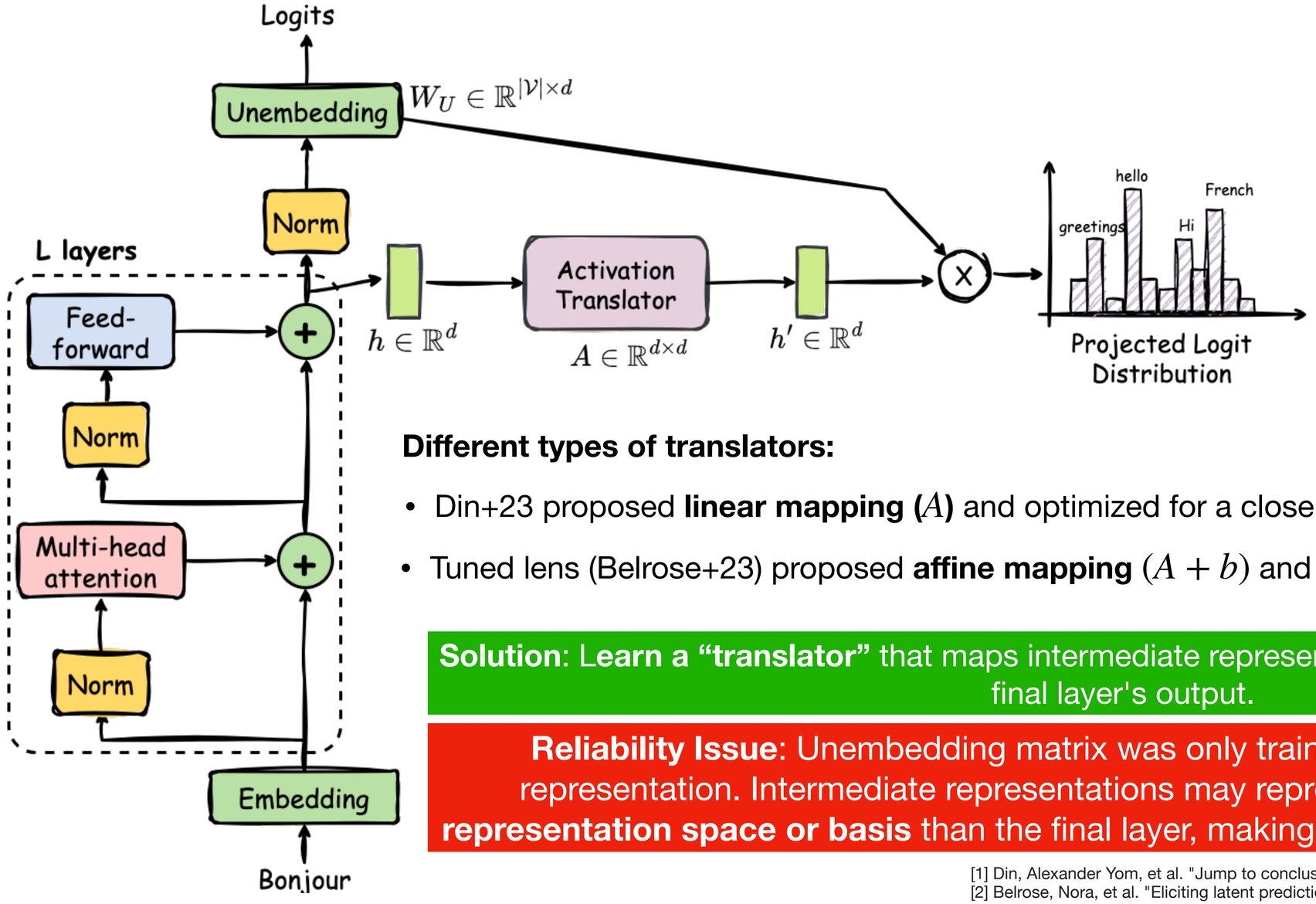


Weights of LM Components

[1] nostalgebraist. Interpreting gpt: the logit lens. AI Alignment Forum, 2020. https://www.lesswrong.com/ posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.
[2] Sakarvadia, Mansi, et al. "Attention lens: A tool for mechanistically interpreting the attention head information retrieval mechanism." arXiv 2023).
[2] Geva, Mor, et al. "Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space." *EMNLP* 2022.
[3] Dar, Guy, et al. "Analyzing Transformers in Embedding Space." *ACL* 2023.

Decoding Position: Vocabulary projection can be applied to different positions of the LM, e.g., activations (nostalgebraist20; Sakarvadia+23), and model weights (Geva+22; Dar+23)

Method 2: Vocabulary Projection



Loss Function

$$\sum_{x \in D} ||A \,.\, h^l_{i,x} - h^L_{i,x}||^2$$

Eq: Din+23

$$argmin \mathbb{E}_{x \in D} \Big[D_{KL}(f_{>l}(h^l_{i,x}) \,||\, TunedLens(h^l_{i,x}; A, b)) \Big]$$

Eq: Tuned lens (Belrose+23)

Different types of translators:

- Din+23 proposed **linear mapping (A)** and optimized for a closer translated vector space
- Tuned lens (Belrose+23) proposed **affine mapping** $(A + b)$ and optimized for a closer logit distribution

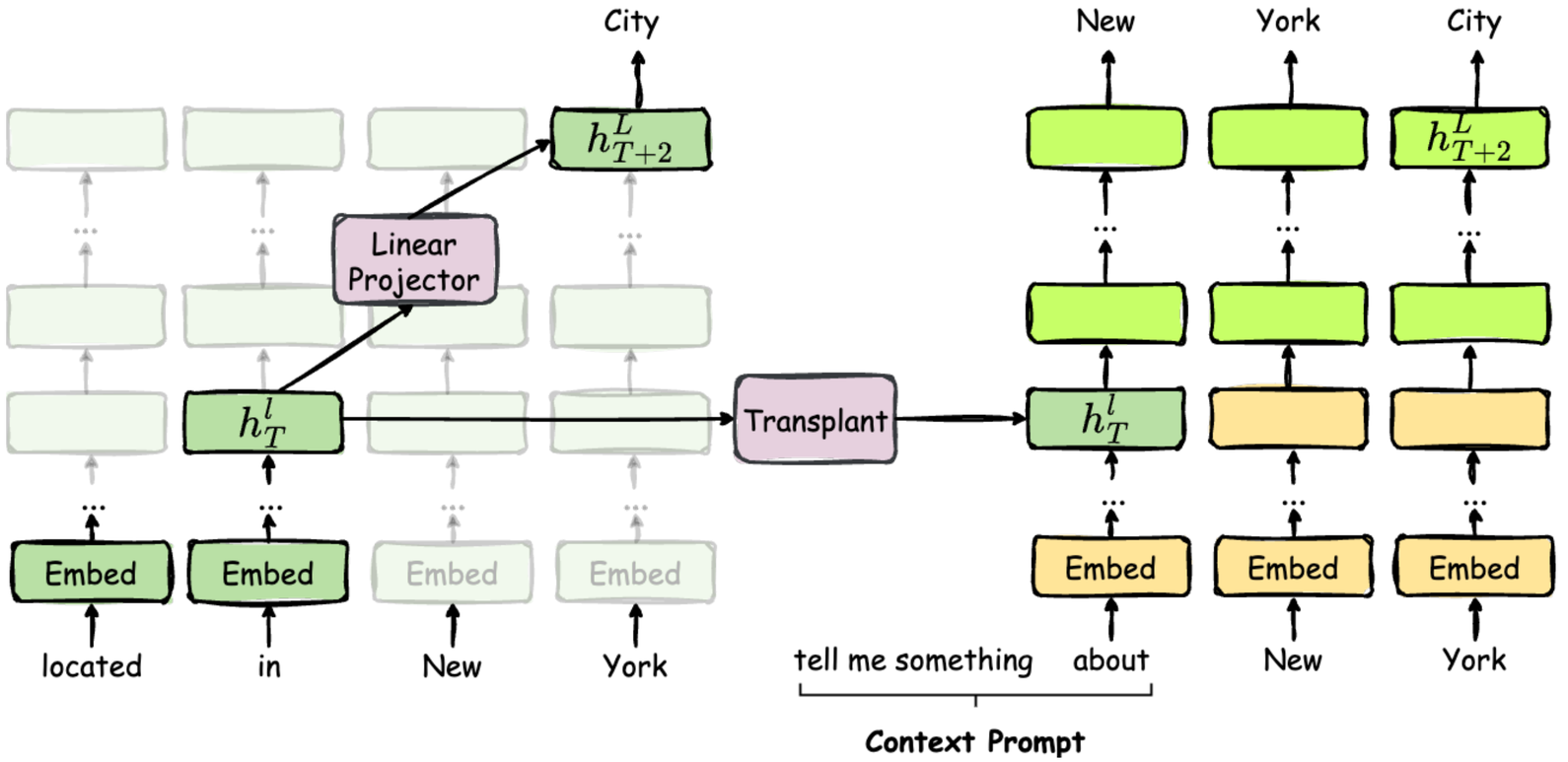
Solution: L**earn a "translator"** that maps intermediate representations into the same space as the final layer's output.

Reliability Issue: Unembedding matrix was only trained to project the final layer representation. Intermediate representations may represent features in a **different representation space or basis** than the final layer, making vocabulary projection unreliable.

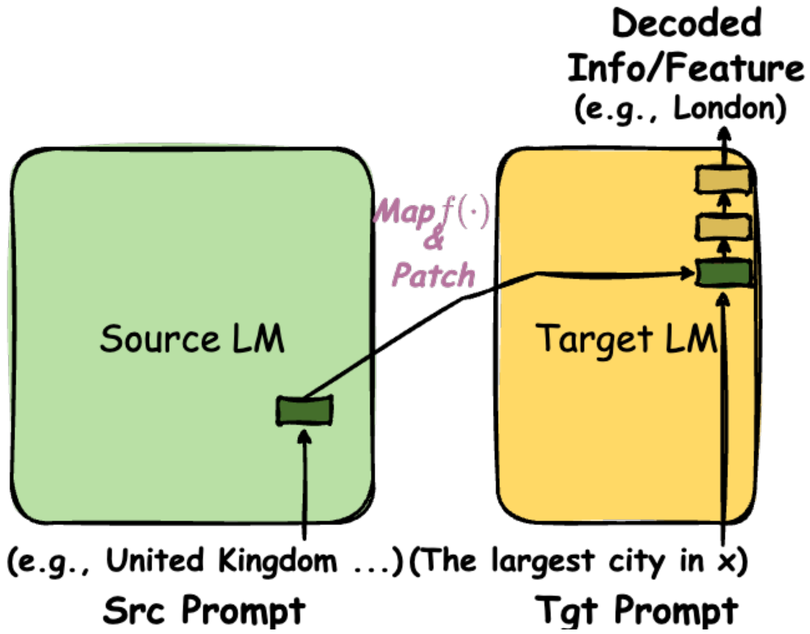
[1] Din, Alexander Yom, et al. "Jump to conclusions: Short-cutting transformers with linear transformations." *LREC-COLING* 2024.
[2] Belrose, Nora, et al. "Eliciting latent predictions from transformers with the tuned lens." arXiv 2023.

Method 2: Vocabulary Projection

Pal+23: Decode future output tokens from the last-token position (**Future Lens**)



Ghandeharioun+24: Generalize the transplanting idea into **Patchscope**



Decoding Expressivity: What other information can we decode from an activation, *beyond* the immediate next-token prediction?

Solution: Project or transplant the activation to an "inspection context" for more expressive decoding

[1] Pal, Koyena, et al. "Future Lens: Anticipating Subsequent Tokens from a Single Hidden State." *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*. 2023.
[2] Ghandeharioun, Asma, et al. "Patchscopes: a unifying framework for inspecting hidden representations of language models." *Proceedings of the 41st International Conference on Machine Learning*. 2024.

Techniques for Open-ended Feature Study

Method 3: Neuron Activation Visualization

Use case: Decode "**all the features**" encoded in the activation
Intuition: Decode features in the *neurons* of an activation via visualizing their patterns

$h \in \mathbb{R}^d$

Each element of activation is referred as a **neuron**

[1.0, 0.5, 2.2, -1.1, … , 3.2, 0.01]

What input tokens result in high neuron activation values?

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.

business in a small garage to become the world's largest manufacturer of "build-it-yourself" component car kits. They employ a full-time crew of about 40 people, and are located in Wareham, Massachusetts (about an hour south of Boston). They make their products right

Human Instruction: mark INTERPRETABLE if 80% or more of the strongest firings can be explained by a single rule or category, and NOT INTERPRETABLE otherwise (Elhage+22)

Neuron seems to encode *"numbers when and only when they refer to a number of people"*

Step 1: Pick a neuron of model activation for interpretation

Step 2: Display a series of text snippets that include tokens where the neuron activates heavily (i.e. showing high values).

Step 3: Human/Machine explains the neuron based on the tokens where the neuron fires heavily (step 2)

Issue: Most neurons activate in response to multiple unrelated features, hindering interpretability.

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

Superposition

Why does a neuron activate for multiple unrelated features?

Too many features, not enough neurons → model uses **linear combinations** of neurons for encoding a feature (**superposition**)

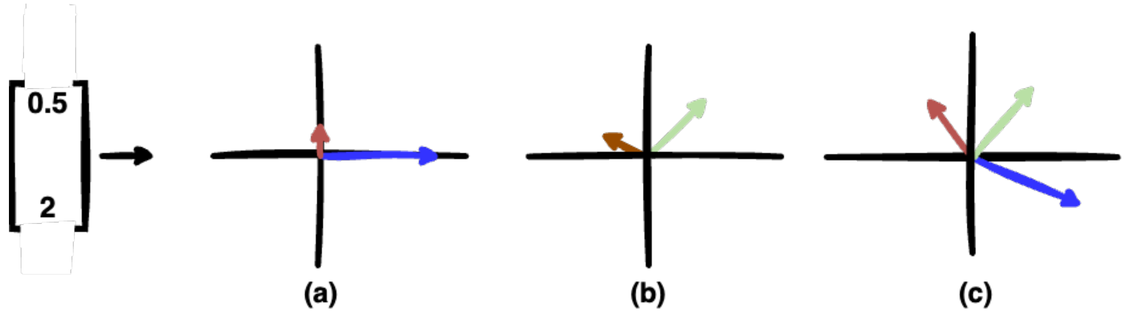


Figure: (a) ***Monosemantic neurons*** *encoding 2 features; (b)* ***Polysemantic neurons*** *encoding 2 features; (c)* ***Polysemantic neurons*** *encoding 3 features (superposition).*

Tradeoff: Superposition enables representation to encode more features, but it also makes interpretation harder (Elhage+22)

Why do we want monosemantic neurons?

Feature Entanglement: Multiple distinct features are entangled, preventing us from isolating them and understanding their individual influence on model behavior.

Partial Understanding: Finding one feature doesn't rule out others — interpretation remains uncertain.

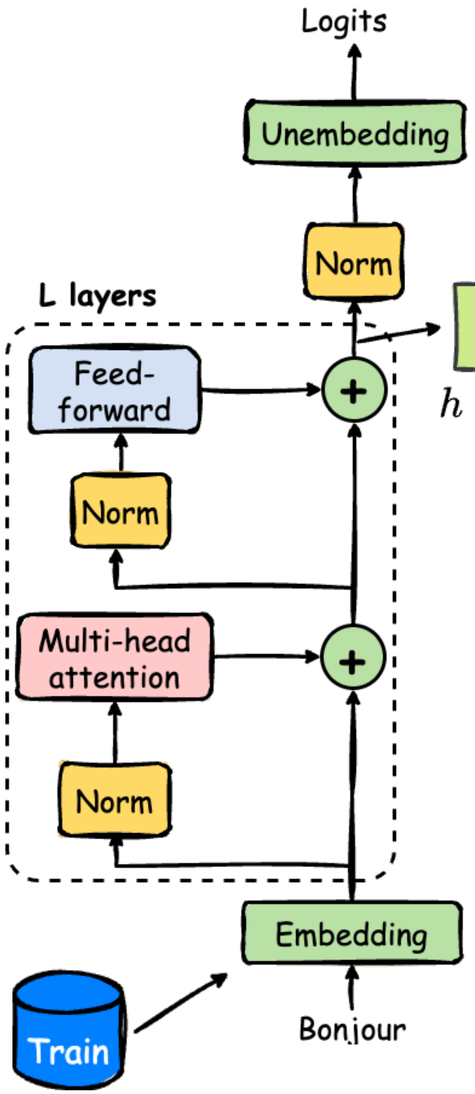
Interpretability Challenge: The neuron's behavior cannot be explained by a single, coherent rule, making interpretation of neuron difficult for both humans and models.

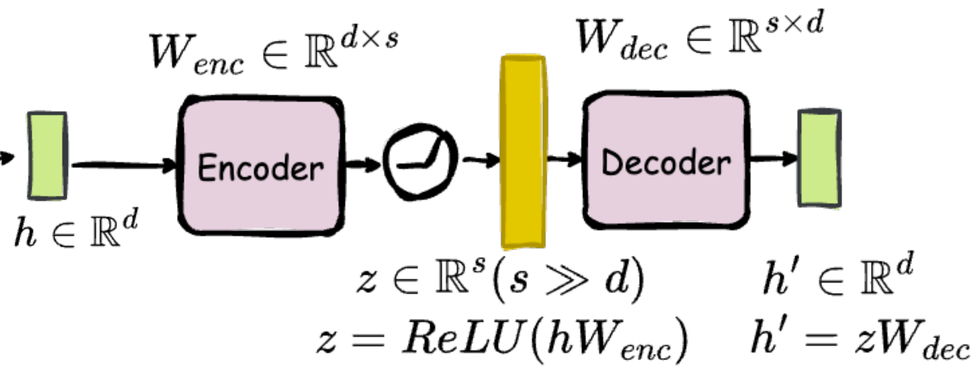
Solution: Can we project superposed representation to more interpretable representations?

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

Method 4: Sparse Autoencoder (SAE)

Use case: Disentangle superposed features to obtain monosemantic, interpretable activations.
Intuition: Map superposed activations to a higher-dimensional sparse activation with monosemantic neurons—each *(ideally)* encoding just one feature.





Loss Function

$$L(h, h') = \underbrace{||h - h'||_2^2}_{\text{Reconstruction Loss}} + \underbrace{\lambda ||z||_1^1}_{\text{Sparsity Loss}}$$

SAE Architecture

- **Encoder**: Projects model activations ($h \in \mathbb{R}^d$) to a higher-dimensional sparse activation ($z \in \mathbb{R}^s, s \gg d$).
  - Aim for z with **monosemantic neurons** and is **sparsely activated**
- **Decoder:** Attempts to reconstructs h from z , yielding h'.
  - Each **row vector** of W_{dec} represents a **feature direction** in the model activation space.

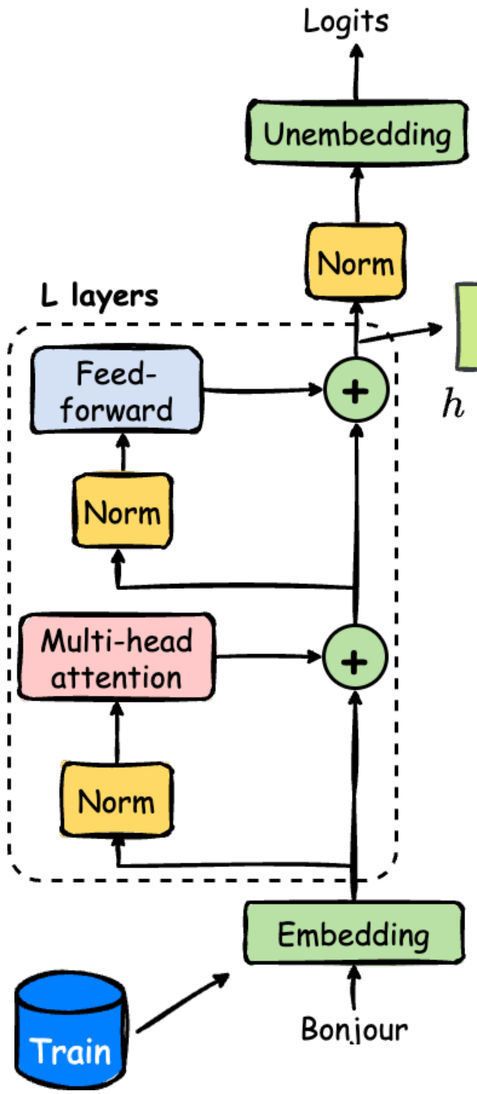
Bricken, et al., "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning", Transformer Circuits Thread, 2023.

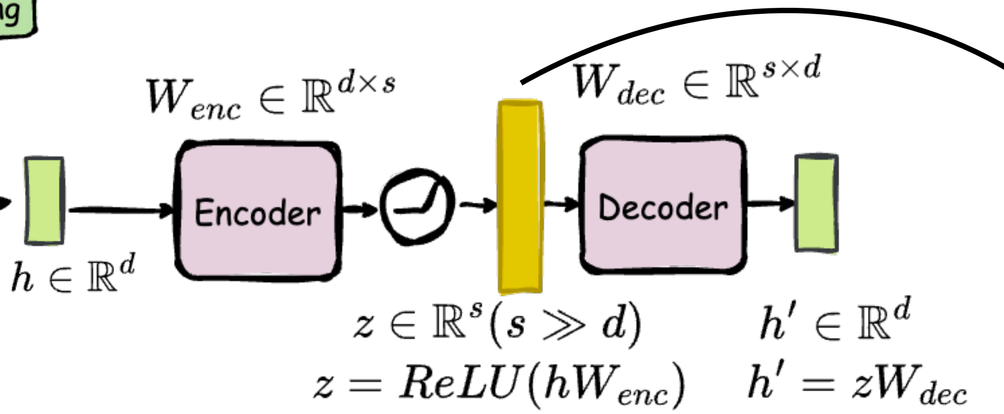
Method 4: Sparse Autoe

Check out our survey for a more comprehensive view!

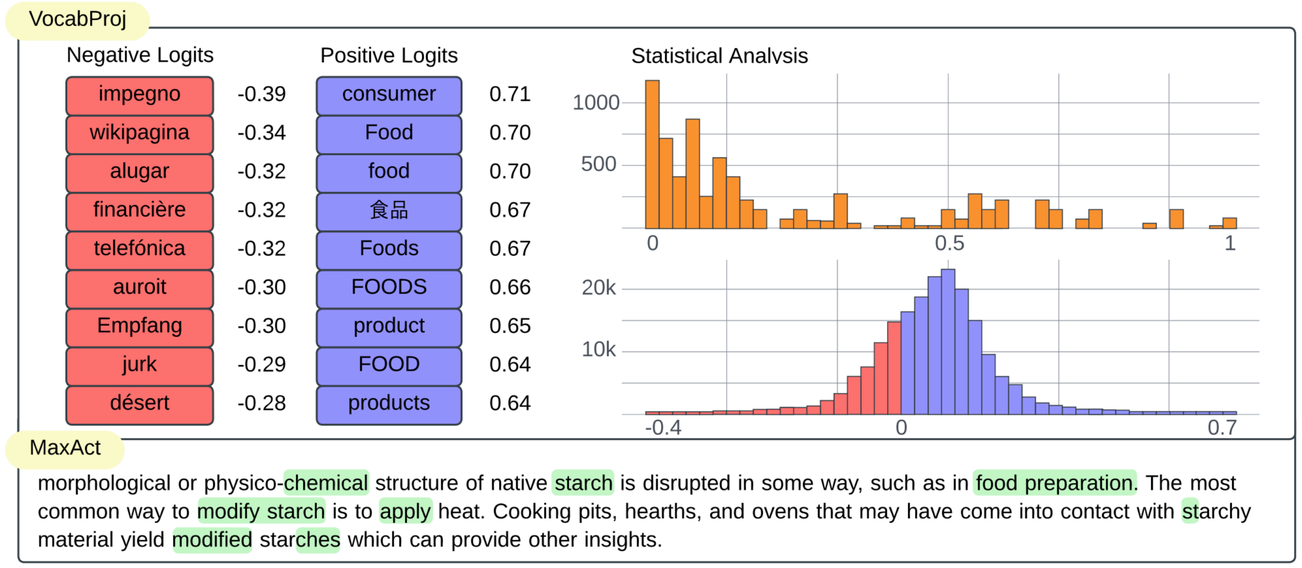
A Survey on Sparse Autoencoders: Interpreting the Internal Mechanisms of Large Language Models

Dong Shu[1,†], Xuansheng Wu[2,†], Haiyan Zhao[3,†], Daking Rai[4], Ziyu Yao[4], Ninghao Liu[2], Mengnan Du[3]
[1]Northwestern University [2]University of Georgia
[3]New Jersey Institute of Technology [4]George Mason University





Explain the sparsely activated feature $W_{dec}[j]$ (row vector of W_{dec}), if $\mathbf{z}[j]$ is high



Vocabulary Projection:
Project $W_{dec}[j]$ to the vocabulary space

Neuron Activation Visualization:
Visualize the activation patterns of neurons of $W_{dec}[j]$

Method 4: Sparse Autoencoder (SAE)

$$z = ReLU(hW_{enc})$$

$$L(h, h') = \underbrace{||h - h'||_2^2}_{\text{Reconstruct}} + \underbrace{\lambda||z||_1^1}_{\text{Sparsity}}$$

Issue 1: Trade-off between Reconstruction and Sparsity Loss

Greater sparsity could reduce the reconstruction effect.

- L1 sparsity loss not only penalizes **the number** of active features but also **the strength** of their activation (Wright&Sharkey24).
- ReLU has implicit **"activation threshold" of zero** which can lead to **false positive** feature activation.

Architectural Solutions:

- TopK SAE (Gao+24): Removes sparsity loss and instead keeps only top-k strongest features.
- JumpReLU SAE (Rajamanoharan+24a): Replaces ReLU with JumpReLU (learned threshold) + L0 loss.
- Gated SAE (Rajamanoharan+24b): Decouples feature activation (via gating) and strength.

Issue 2: Discovering functionally important features

- Standard SAE-discovered features ≠ functionally meaningful features

Example Solution: End-to-End SAE (Braun+24), which replaced the reconstruction loss with minimizing the KL divergence that preserve the *task prediction distributions*

Issue 3: Concern about practical usefulness of SAE features vs. baselines (To discuss in Findings & Applications)

[1] Benjamin Wright and Lee Sharkey. Addressing feature suppression in saes. In AI Alignment Forum, pp. 16, 2024.
[2] Leo Gao, etc.. Scaling and evaluating sparse autoencoders. *ICLR* 2025.
[3] Rajamanoharan, Senthooran, et al. "Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders." CoRR (2024a).
[4] Rajamanoharan, Senthooran, et al. "Improving Dictionary Learning with Gated Sparse Autoencoders." CoRR (2024b).
[5] Braun, Dan, et al. "Identifying functionally important features with end-to-end sparse dictionary learning." *NeurIPS* 2024.

Methods for Feature Study Evaluation

Methods for Feature Study Evaluation

Faithfulness

*Measures whether the feature **truly exists** in the LM's internal representation.*

Interpretability

Measures how well a feature maps to a simple coherent human- or LM-generated explanation.

Technique-specific Evaluation

Causal-Intervention Evaluation

Automatic LM Evaluation

Each technique consists of its own evaluation measures.

- **Probing** - > Accuracy/F1-score on held-out test set
- **Vocabulary projection method** - > Recall of the target "answer token" within the top-k logits on the test set.
- **SAE** - > Reconstruction loss

Methods for Feature Study Evaluation

Faithfulness

*Measures whether the feature **truly exists** in the LM's internal representation.*

Technique-specific Evaluation

Causal-Intervention Evaluation

Automatic LM Evaluation

Interpretability

Measures how well a feature maps to a simple coherent human- or LM-generated explanation.

How does the feature activation affect the model output?

SAE finds the "French" feature

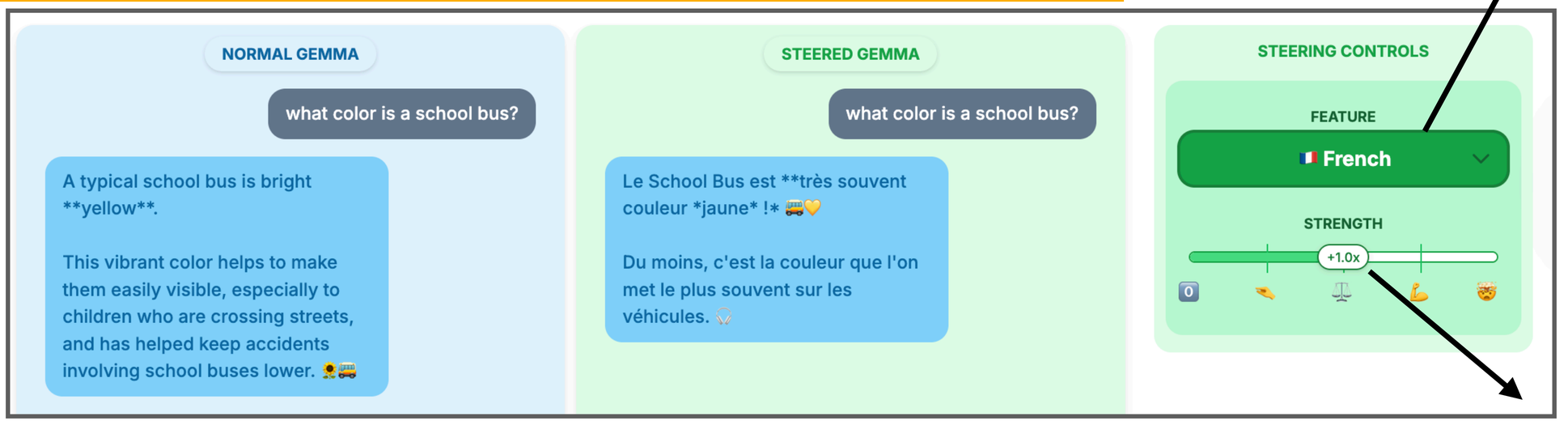


Figure: The figure references the Neuronpedia website (Lin+23)

What happens if we manually increase the activation of the "French" feature?

[1] Johnny Lin. 2023. Neuronpedia: Interactive reference and tooling for analyzing neural networks. Software available from neuronpedia.org.

Methods for Feature Study Evaluation

Faithfulness

*Measures whether the feature **truly exists** in the LM's internal representation.*

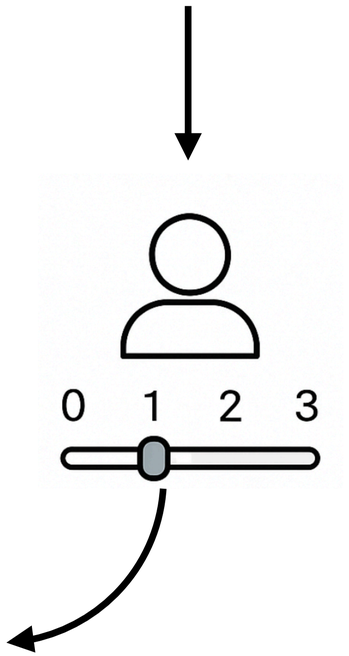
Technique-specific Evaluation

Causal-Intervention Evaluation

Automatic LM Evaluation

Interpretability

Measures how well a feature maps to a simple coherent human- or LM-generated explanation.



Rubric for scoring how understandable features are.
Example scoring instructions (Bricken+23):

1. Form an interpretation of the feature based on the positive logits and feature activations across all intervals.
2. On a scale of 0-3, rate your confidence in this interpretation.
3. On a scale of 0–5, rate how consistent the high-activation (bolded) tokens are with your interpretation from (1).
4. On a scale of 0–3, rate how consistent the positive logit effects are with your interpretation from (1).
5. If some of the positive logit effects were inconsistent with your interpretation, was there a separation in effect size between the consistent and inconsistent ones? If so, score as 1, otherwise or if not applicable, score as 0.
6. On a scale of 0–3, how specific is your interpretation of this feature?

[1] Bricken, et al., "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning", Transformer Circuits Thread, 2023.

Methods for Feature Study Evaluation

Faithfulness

*Measures whether the feature **truly exists** in the LM's internal representation.*

Technique-specific Evaluation

Causal-Intervention Evaluation

Interpretability

Measures how well a feature maps to a simple coherent human- or LM-generated explanation.

Automatic LM Evaluation

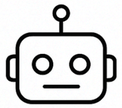
Can we use LMs to perform Faithfulness evaluation? (Bill+23)

Prompt to LM: *The neuron activates on "references to movies, characters, and entertainment." Based on this explanation of what the neuron does, and try to predict its activations on a particular token.*

Step 1: **Simulate** activations using GPT-4, conditioning on the explanation

Simulated activations:

: Age of Ultron and it sounds like his role is going to play a bigger part in the Marvel cinematic universe than some of you originally thought. Marvel has a new press release that offers up some information on the characters in the film. Everything included in it is pretty standard stuff, but then there was this new



Explanation Score: 0.337

Step 2: **Score** the explanation by comparing the simulated and real activations

Real activations:

: Age of Ultron and it sounds like his role is going to play a bigger part in the Marvel cinematic universe than some of you originally thought. Marvel has a new press release that offers up some information on the characters in the film. Everything included in it is pretty standard stuff, but then there was this new

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

Examples for Feature Study

Example of Targeted Feature Study (Gurnee+23)

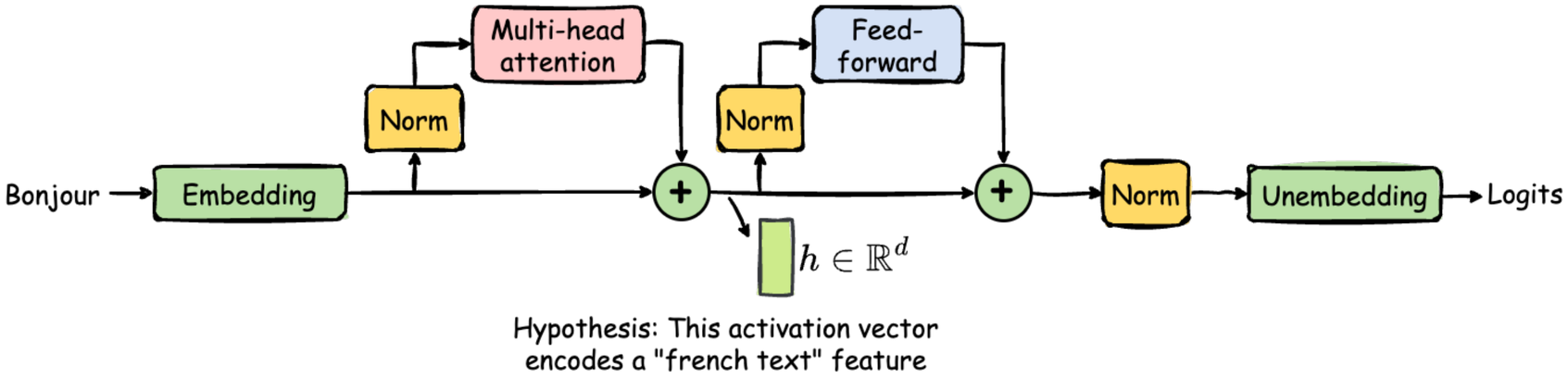
Step 1

Hypothesis Generation → **Hypothesis Validation** → **Evaluation**

Hypothesizing the present of feature $feat$ in activation h
***Techniques**: human intuition*

Validating the hypothesis.
***Techniques**: Probing*

Measure faithfulness



Categories of features

- **Part-of-speech:** AUX, ADP, ..
- **Morphology:** eos_True, Person_2, Tense_Past, ..
- **Latex:** is_superscript, is_title, is_reference, ..
- **Gender:** male, female
- **Occupation:** is_athelete, is_journalist, is_singer, ..
- …

Investigates 100 pre-defined features across 7 LMs of varying sizes, aiming to localize each feature to **specific neuron(s) within activations**.

[1] Wes Gurnee, et al., Finding neurons in a haystack: Case studies with sparse probing. Transactions on Machine Learning Research, 2023.

Example of Targeted Feature Study (Gurnee+23)

Step 2

Hypothesis Generation

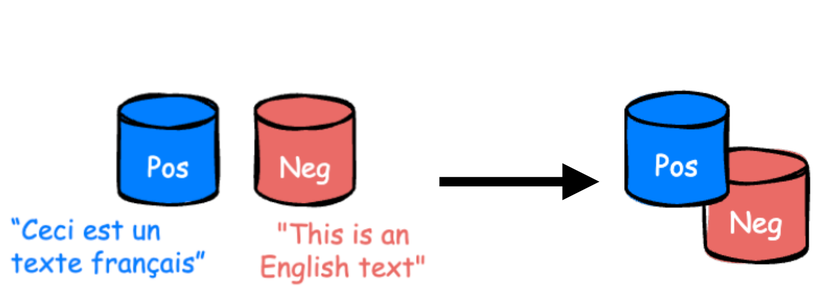
Hypothesizing the present of feature $feat$ in activation h.
***Techniques**: human intuition*

Hypothesis Validation

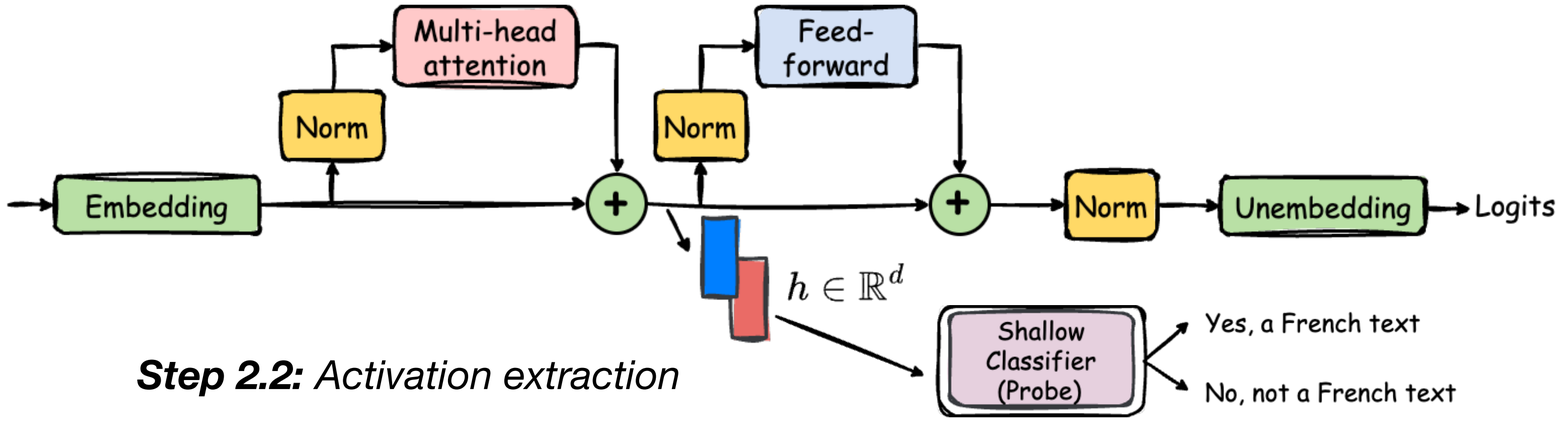
Validating the hypothesis.
***Techniques**: Probing*

Evaluation

Measure faithfulness



***Step 2.1:** Dataset preparation*



***Step 2.2:** Activation extraction*

***Step 2.3:** Train a sparse probe*

*(Probe trained using only the **top K most informative neurons**.)*

How to select subset of neurons?

e.g., Adaptive thresholding:

- Iteratively retraining probes: At each steps, it selects top-k neurons from previous k-sparse probe
- Decrease *k* iteratively and retrain probe

[1] Wes Gurnee, et al., Finding neurons in a haystack: Case studies with sparse probing. Transactions on Machine Learning Research, 2023.

Example of Targeted Feature Study (Gurnee+23)

Hypothesis Generation

Hypothesizing the present of feature $feat$ in activation h.
***Techniques**: human intuition*

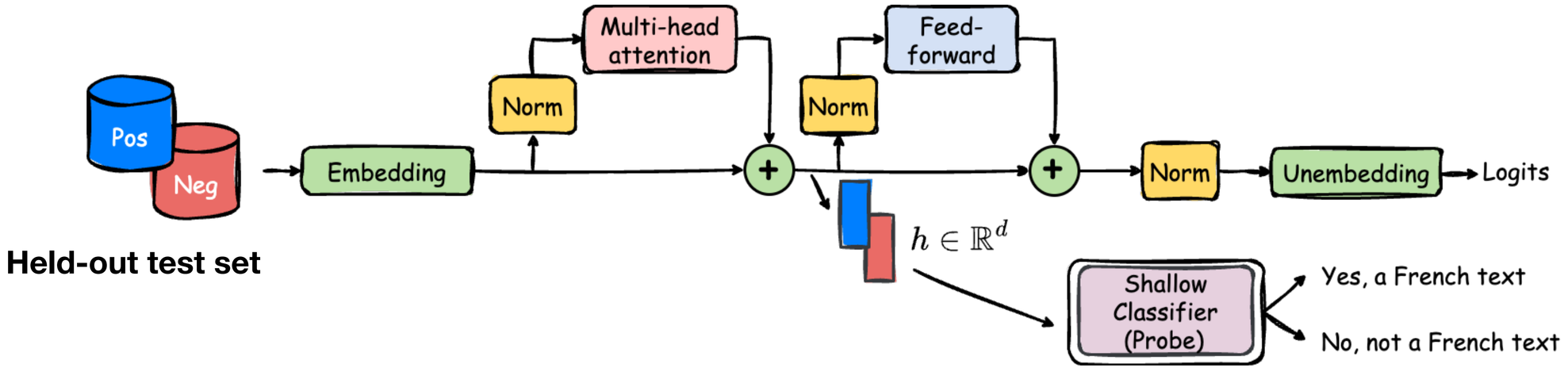
Hypothesis Validation

Validating the hypothesis.
***Techniques**: Probing*

Step 3

Evaluation

Measure faithfulness and human-understandability



- High F1 suggests selected neurons encode the target feature (e.g., *is_french_text*)

Evaluate F1 score of the learned probe on **test set**

[1] Wes Gurnee, et al., Finding neurons in a haystack: Case studies with sparse probing. Transactions on Machine Learning Research, 2023.

Example of Open-ended Feature Study (Bricken+23)

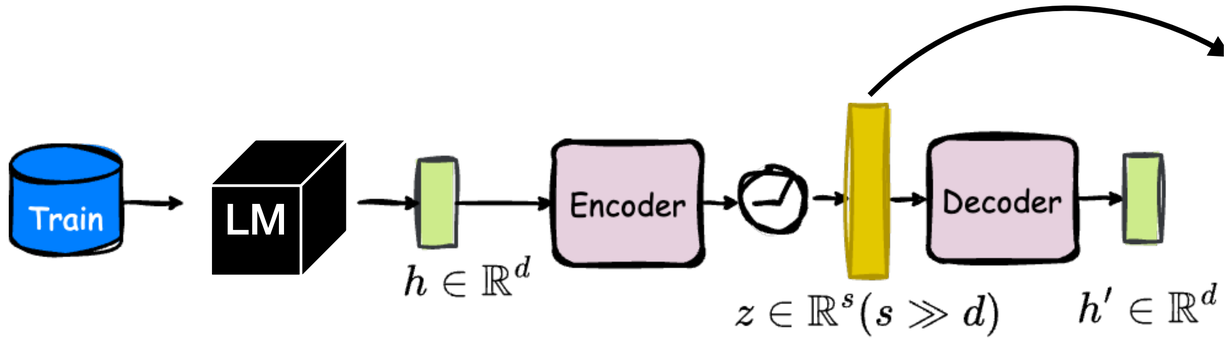
Step 1

Observe → Explain → Evaluation

Observing activation patterns.
***Techniques**: Vocab Projection, Sparse Autoencoder, and Activation Visualization*

Does the observed activation pattern encode feature $feat$?
***Techniques**: by human or machine.*

Measure faithfulness and human-understandability

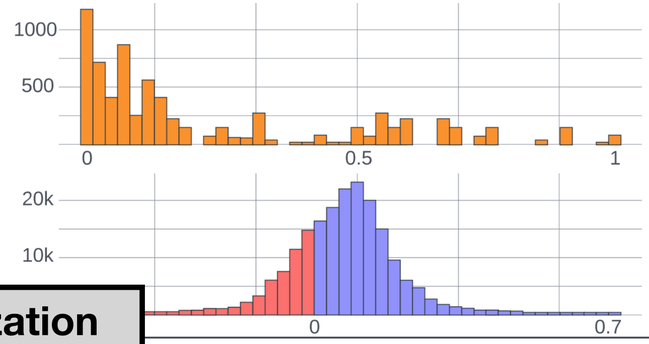


Sparse Autoencoder (SAE)

Vocabulary Projection

Negative Logits		Positive Logits	
impegno	-0.39	consumer	0.71
wikipagina	-0.34	Food	0.70
alugar	-0.32	food	0.70
financière	-0.32	食品	0.67
telefónica	-0.32	Foods	0.67
auroit	-0.30	FOODS	0.66
Empfang	-0.30	product	0.65
jurk	-0.29	FOOD	0.64

Statistical Analysis



Neuron Activation Visualization

morphological or physico-chemical structure of native starch is disrupted in some way, such as in food preparation. The most common way to modify starch is to apply heat. Cooking pits, hearths, and ovens that may have come into contact with starchy material yield modified starches which can provide other insights.

Bricken+23 conducted an open-ended feature discovery on a one-layer toy LM.

{1] Trenton Bricken, et al., Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Example of Open-ended Feature Study (Bricken+23)

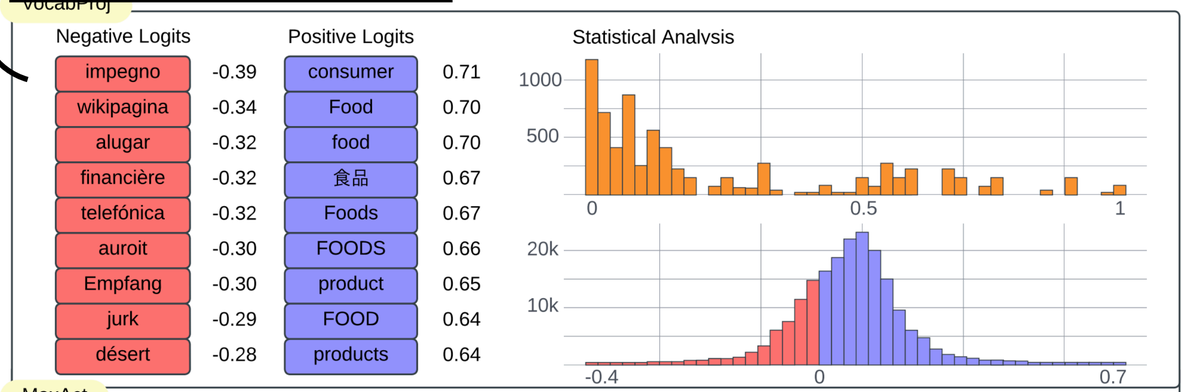
Observe → **Step 2** **Explain** → Evaluation

Observing activation patterns.
***Techniques**: Vocab Projection, Patchscope, Sparse Autoencoder, and Activation Visualization*

Does the observed activation pattern encode feature $feat$?
***Techniques**: by human or machine.*

Measure faithfulness and human-understandability

Vocabulary Projection



Neuron Activation Visualization

Explain the Observation

The SAE feature seems to mainly activate for text that relates to “food” and “food product”

{1] Trenton Bricken, et al., Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Example of Open-ended Feature Study (Bricken+23)

Observe → Explain → Step 3 **Evaluation**

Observing activation patterns.
***Techniques**: Vocab Projection, Sparse Autoencoder, and Activation Visualization*

Does the observed activation pattern encode feature $feat$?
***Techniques**: by human or machine.*

Measure faithfulness and human-understandability

The SAE feature seems to mainly activate for text that relates to “food” and “food product”

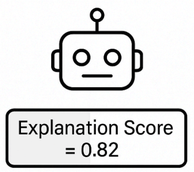
Faithfulness

Measures whether the feature truly exists in the LM’s internal representation.

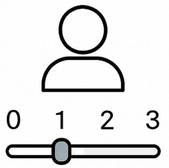
SAE’s Reconstruction Loss: How well does the autoencoder reconstruct the activations?

Interpretability

Measures how well a feature maps to a simple coherent human- or LM-generated explanation.



Auto Eval: How well can LM simulate the neuron activation based on the generated explanation



Human Eval: On a scale of 0-3, rate your confidence in this interpretation…

{1] Trenton Bricken, et al., Towards monosemanticity: Decomposing language models with dictionary learning. Transformer Circuits Thread, 2023.

Part 2.2: Circuit Study

Overview of Workflows for Circuit Study

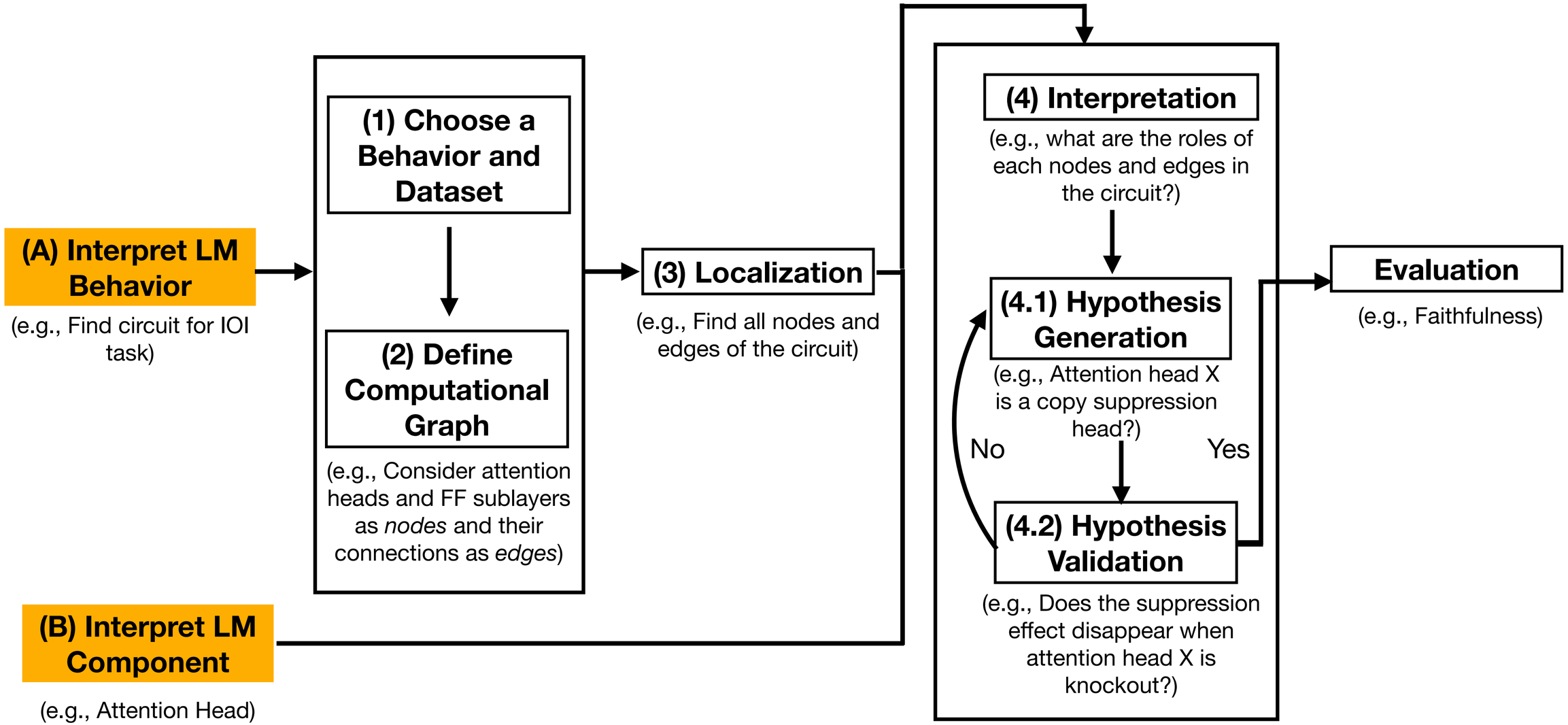
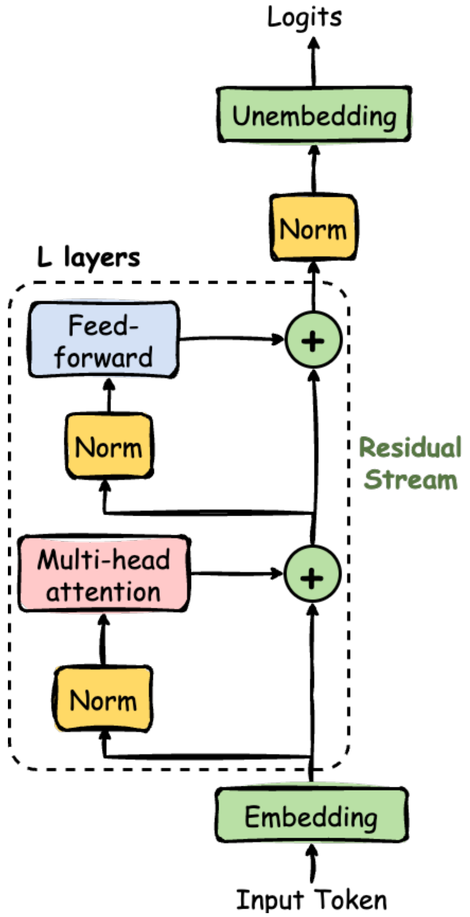
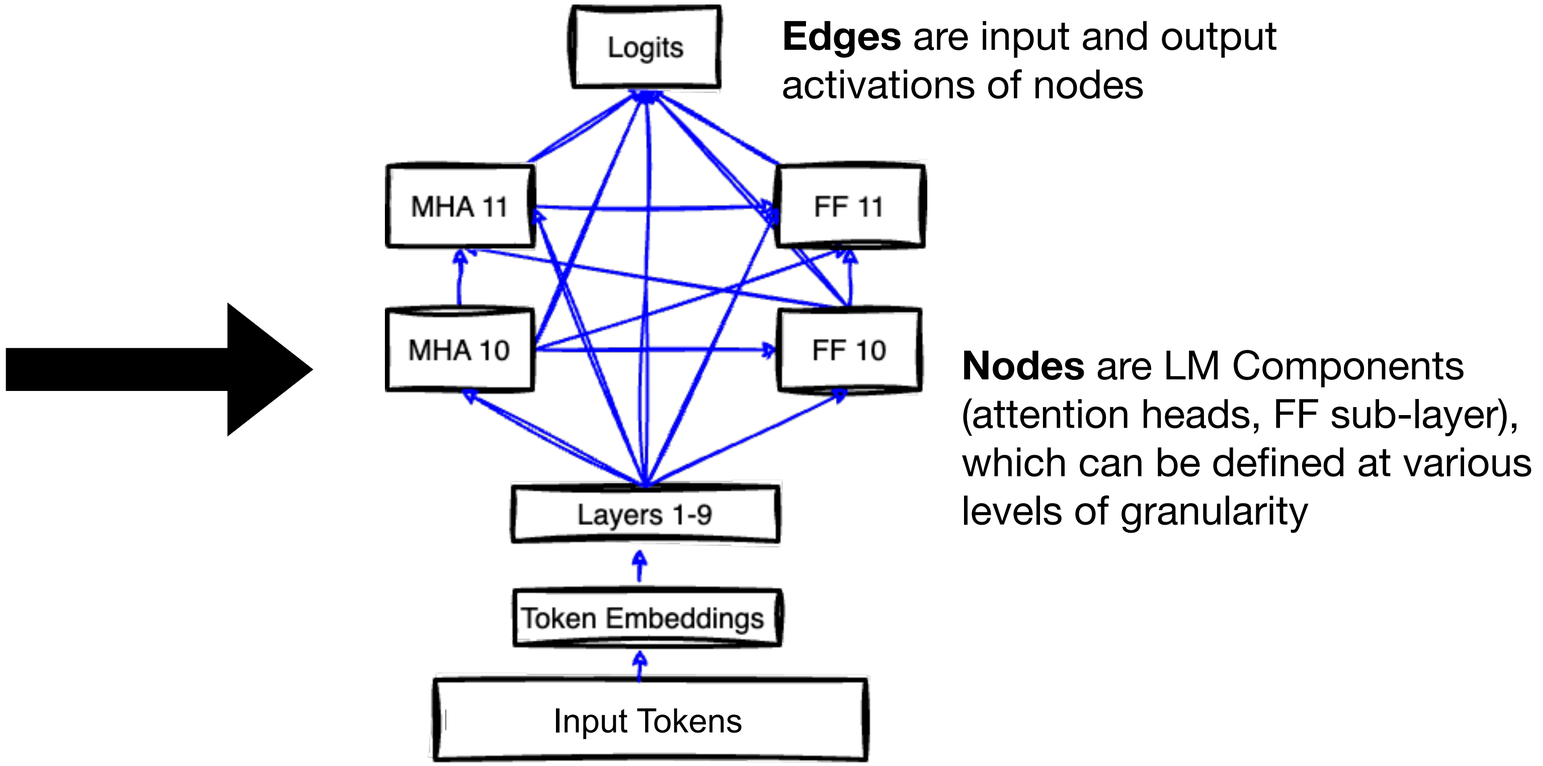


Figure: Task-centric Beginner's Roadmap to Circuit Study

Example Graph

Components across non-adjacent layers are connected through Residual Stream (Hanna+23)





Edges are input and output activations of nodes

Nodes are LM Components (attention heads, FF sub-layer), which can be defined at various levels of granularity

Figure: Transformer represented as Directed Acyclic Graph (DAG) / Computational graph.

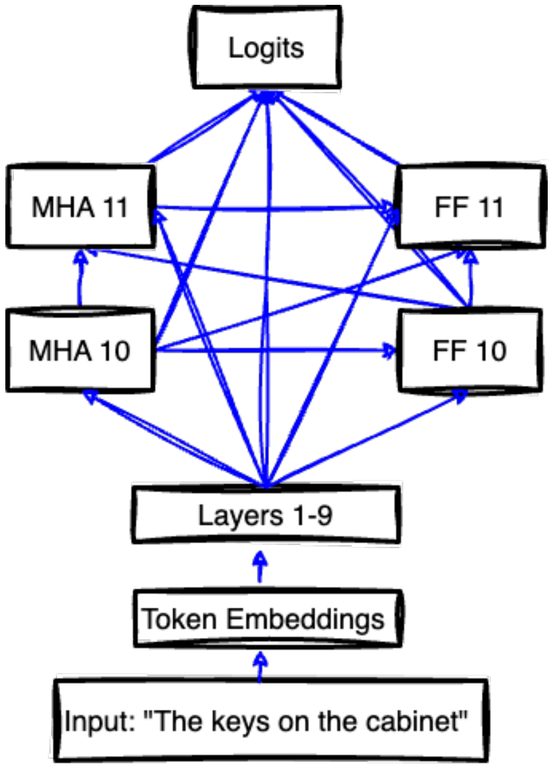
[1] Hanna, Michael, et al. “How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model.” NeurIPS 2023.

Techniques for Circuit Study

Method 5: Intervention

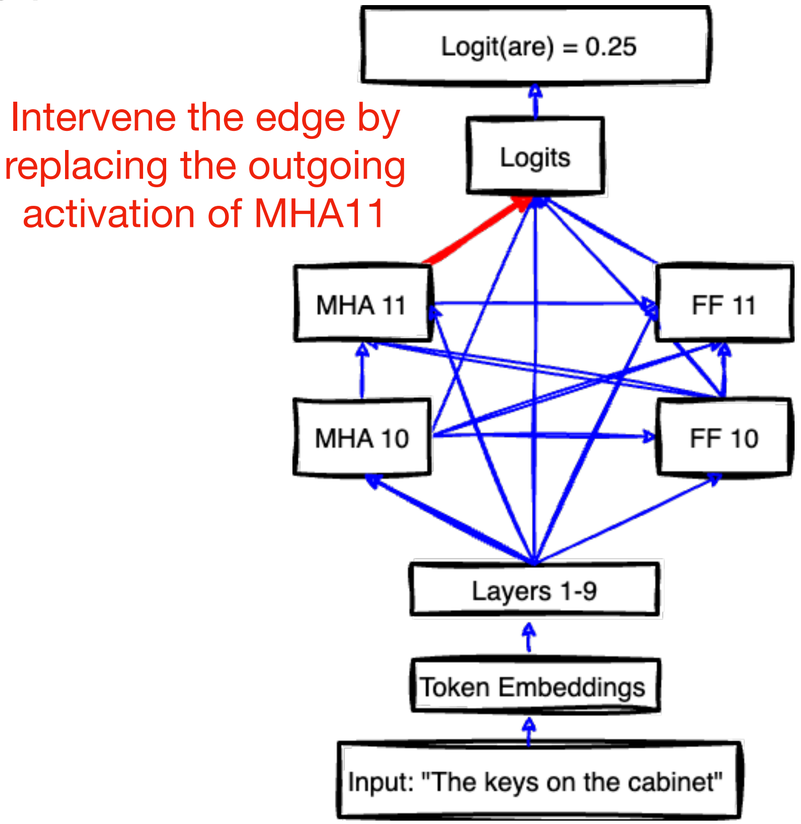
Use case: Determine if a component or edge is important for specific model behavior
Intuition: Intervene on the component or edge activation during the forward pass and observe changes in the output.

Example: How does the LM know to predict “are” instead of “is” following the input *“The keys on the cabinet”*?



Step 1: Represent LM as a computational graph

Intervene the edge by replacing the outgoing activation of MHA11



Step 2: Perform Intervention

Step 3: Observe the change on the model output

If the correct token (“are”) logit is changed significantly, then this edge is important for subject-verb agreement

Metrics:

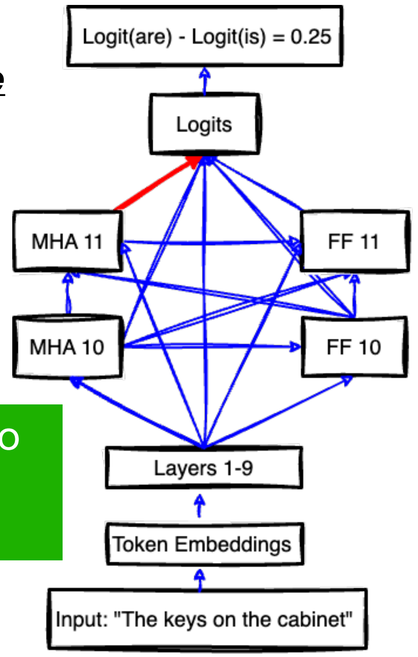
1. **Logit:** Logit(“are”)
2. **Probability:** Sofmax(“are”)
3. **Logit Difference:** Logit(“are”) - Logit(“is”)
4. **KL-Divergence:** KL divergence between the output probability distributions before and after intervention

Method 5: Intervention

Noising vs. Denoising Intervention Techniques

Noising intervention **removes** the component from the graph to analyze it's importance.

*Intervention should **decrease** the logit for "**are**".*



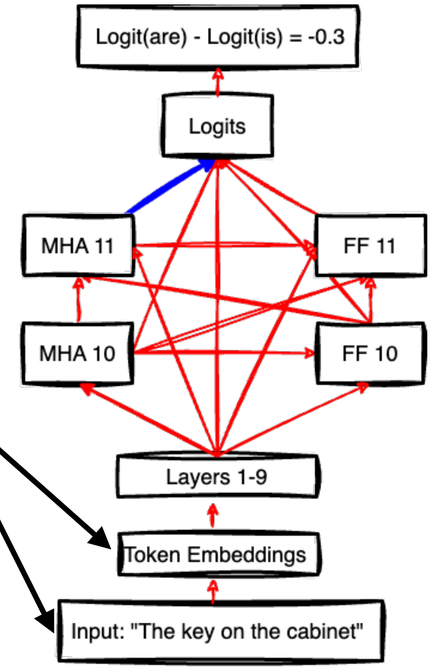
Noising tests **necessity**; also referred as "Ablation" or "Knockout".

Noising Intervention

Denoising intervention evaluates a component's importance by considering only its contribution to the model's output.

*Intervention should **increase** the logit for "**are**".*

Corrupt embedding by adding noise or use counterfactual input.



Denoising tests **sufficiency**; also referred as "Causal Tracing" or "Causal Mediation Analysis".

Denoising Intervention

Method 5: Intervention

Noising Intervention

Different noising methods differ in how they remove component contributions.

Zero Ablation (Unrealistic, often OOD)

- Replace with zero activation

Random Ablation (Unrealistic, often OOD)

- Replace with random vector

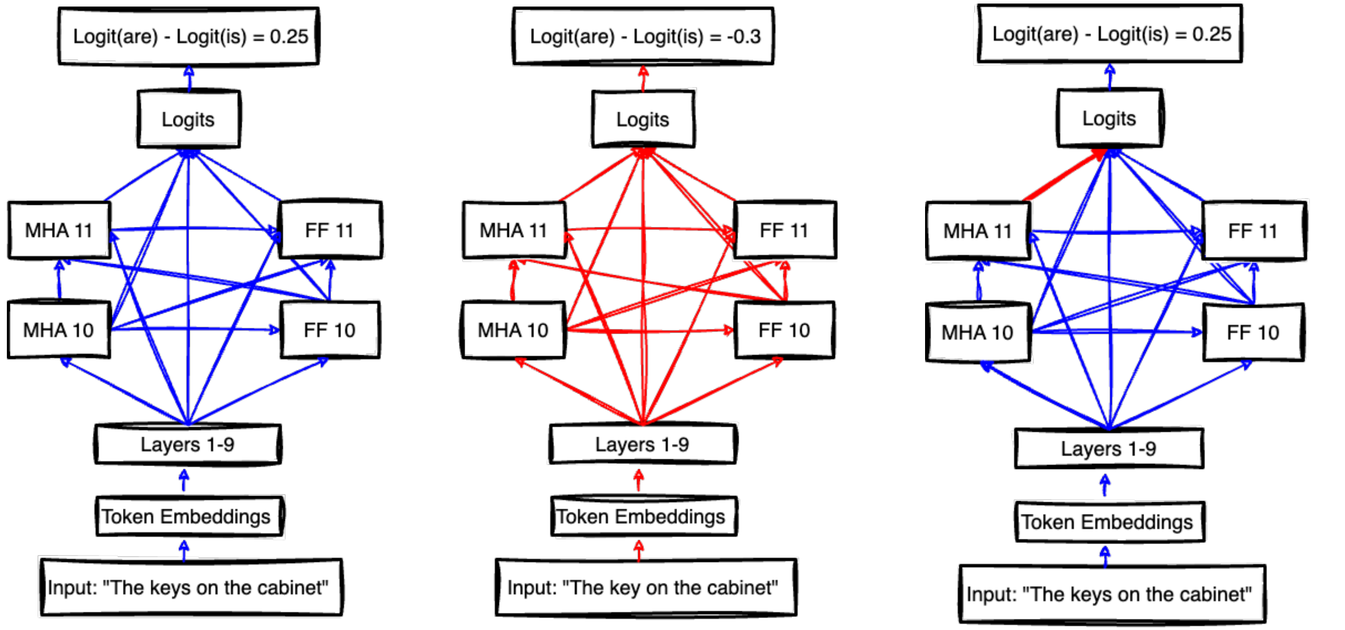
Resampling Ablation (Best mitigates OOD risk)

- Replace with activation from a real, counterfactual in-distribution input.

Mean Ablation (More realistic than Zero/Random)

- Replace with average activation over in-distribution samples
- Still problematic if the true distribution is **non-linear**

Issue: The activations used to replace the original activation can push the model out of distribution (OOD)



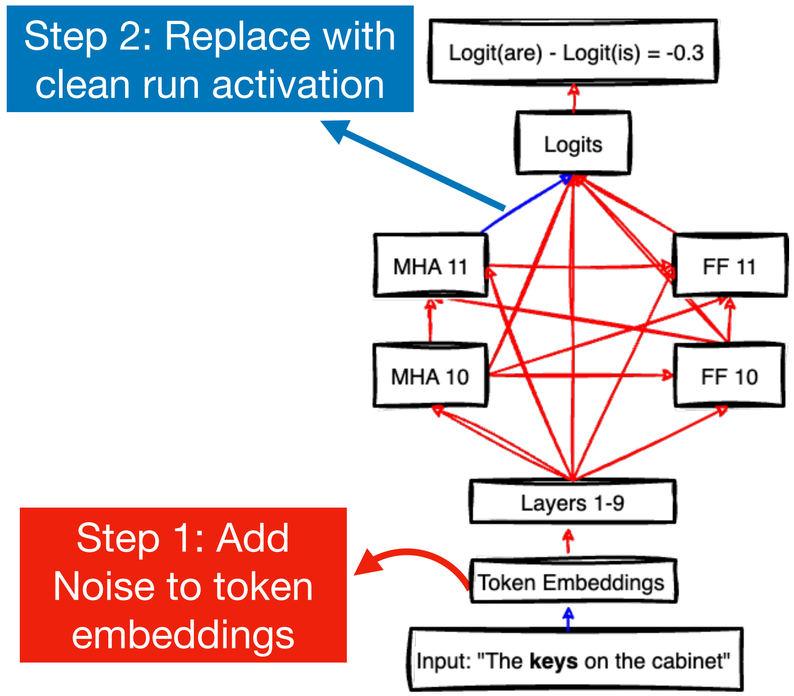
Clean Run **Corrupt Run** **Resampling Intervention**

Method 5: Intervention

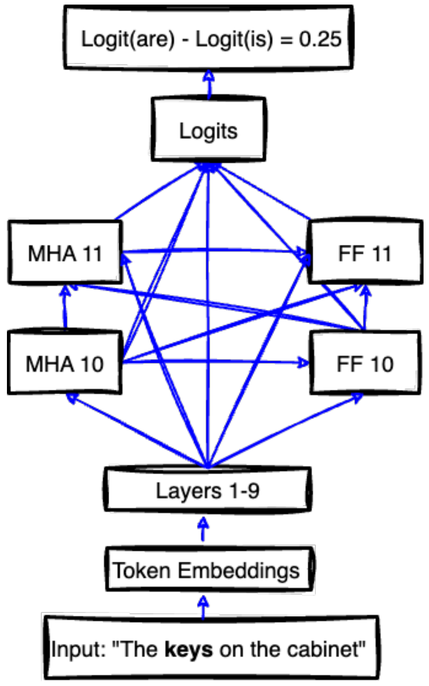
Denoising Intervention

1. Random-noise Denoising Intervention

Perform corrupt run by adding noise to the embedding of the clean run input.



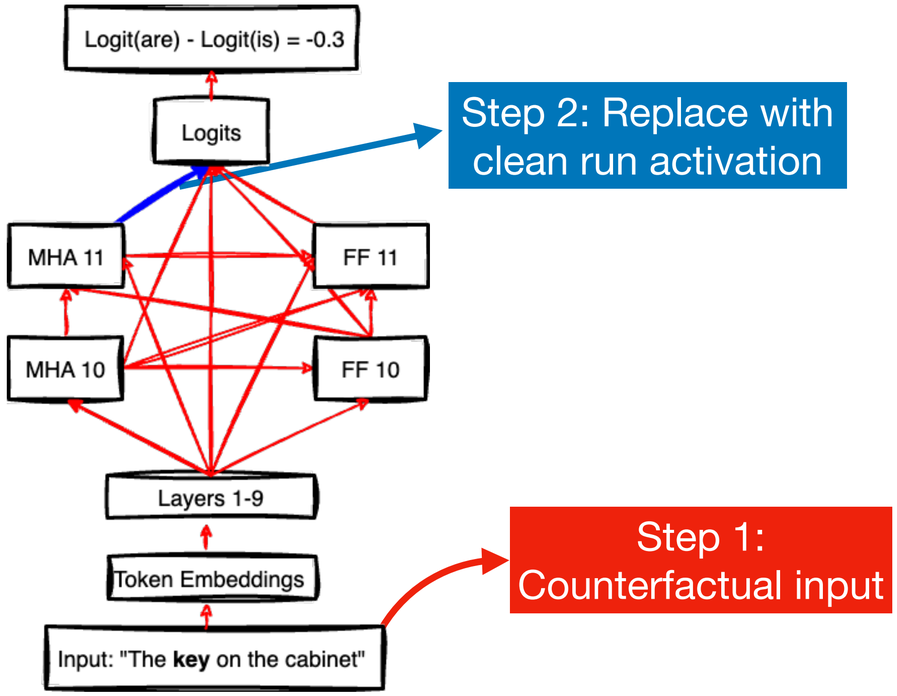
Random-noise



Clean Run

2. Resampling Denoising Intervention

Perform corrupt by running the counterfactual input.



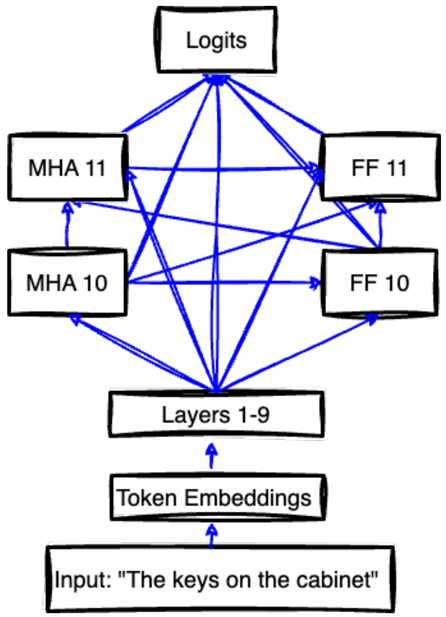
Resampling Denoising Intervention

Issue: The corruption of token embedding by adding random noise can push the model OOD.

Method 5: Intervention for Localization

Use case: Find all the important nodes and edges that implements specific LM behavior

Clean Run (Edge) | Corrupt Run (Edge) | Indirectly Affected Edge



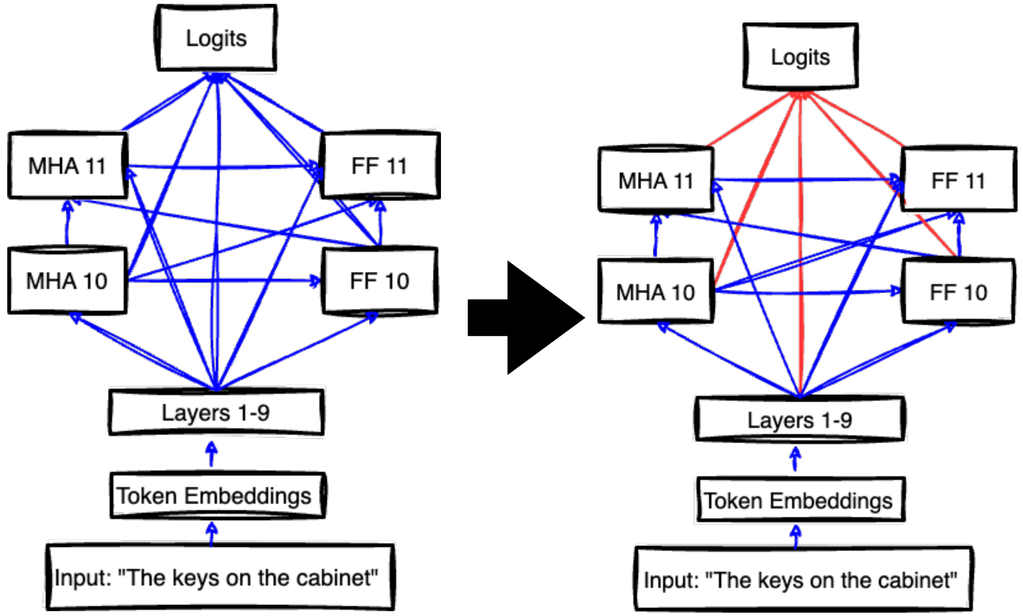
Step 1: Define computational graph.

[1] Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." *NeurIPS* 2023.
[2] Syed, Aaquib, et al. "Attribution Patching Outperforms Automated Circuit Discovery." *BlackboxNLP Workshop* 2024.

Method 5: Intervention for Localization

Use case: Find all the important nodes and edges that implements specific LM behavior

Clean Run (Edge) | Corrupt Run (Edge) | Indirectly Affected Edge



Step 1: Define computational graph.

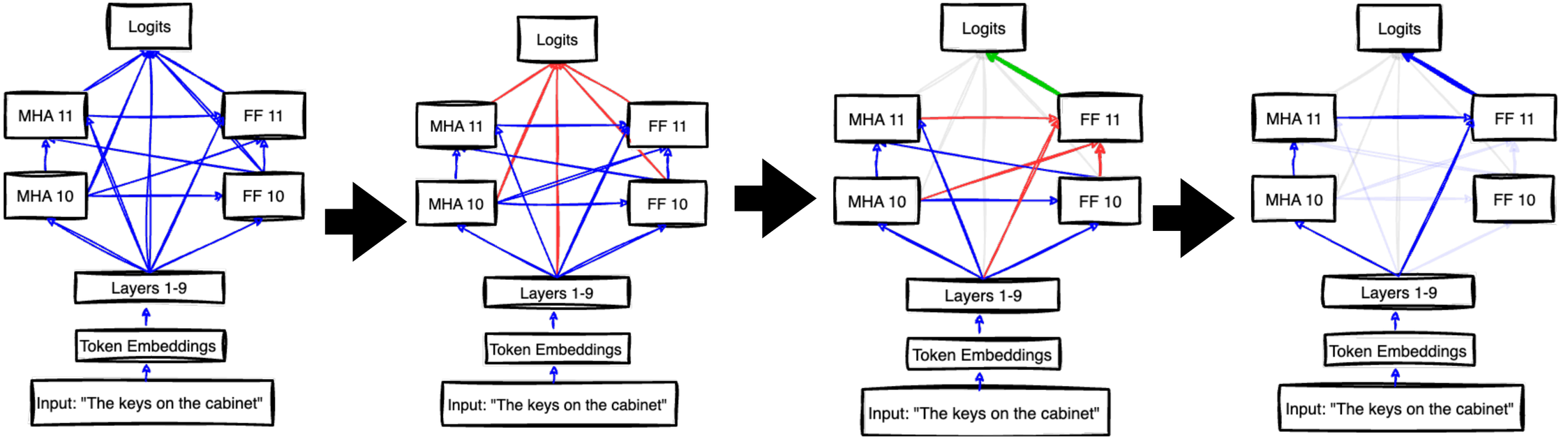
Step 2: Find nodes that directly impact the logits node.

[1] Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." *NeurIPS* 2023.
[2] Syed, Aaquib, et al. "Attribution Patching Outperforms Automated Circuit Discovery." *BlackboxNLP Workshop* 2024.

Method 5: Intervention for Localization

Use case: Find all the important nodes and edges that implements specific LM behavior

Clean Run (Edge) Corrupt Run (Edge) Indirectly Affected Edge



Step 1: Define computational graph.

Step 2: Find nodes that directly impact the logits node.

Step 3: Trace upstream nodes influencing Step 2 results, iterating back to the token embedding layer.

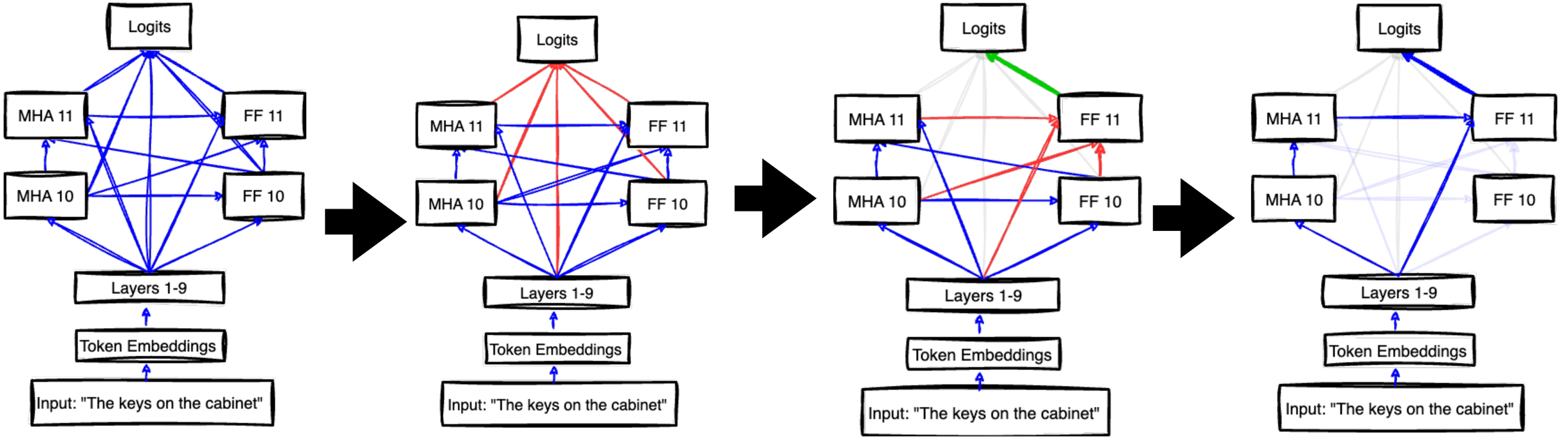
Step 4: At the end, you have a circuit with important nodes and edges.

[1] Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." *NeurIPS* 2023.
[2] Syed, Aaquib, et al. "Attribution Patching Outperforms Automated Circuit Discovery." *BlackboxNLP Workshop* 2024.

Method 5: Intervention for Localization

Use case: Find all the important nodes and edges that implements specific LM behavior

Clean Run (Edge) | Corrupt Run (Edge) | Indirectly Affected Edge



Step 1: Define computational graph.

Step 2: Find nodes that directly impact the logits node.

Step 3: Trace upstream nodes influencing Step 2 results, iterating back to the token embedding layer.

Step 4: At the end, you have a circuit with important nodes and edges.

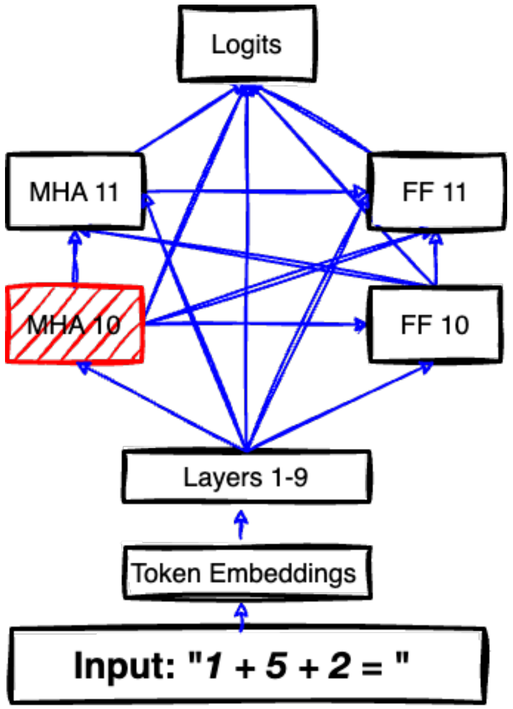
Issue: Manual intervention experiment requires lot of human effort and computationally expensive.

Solution: Automated techniques such as ACDC (Conmy+23), EAP (Syed+24), etc.

[1] Conmy, Arthur, et al. "Towards automated circuit discovery for mechanistic interpretability." *NeurIPS* 2023.
[2] Syed, Aaquib, et al. "Attribution Patching Outperforms Automated Circuit Discovery." *BlackboxNLP Workshop* 2024.

Method 5: Intervention for Validating Hypotheses

Use case: Validating a hypothesize computation via intervention. Example: Causal Scrubbing (Chen+22)

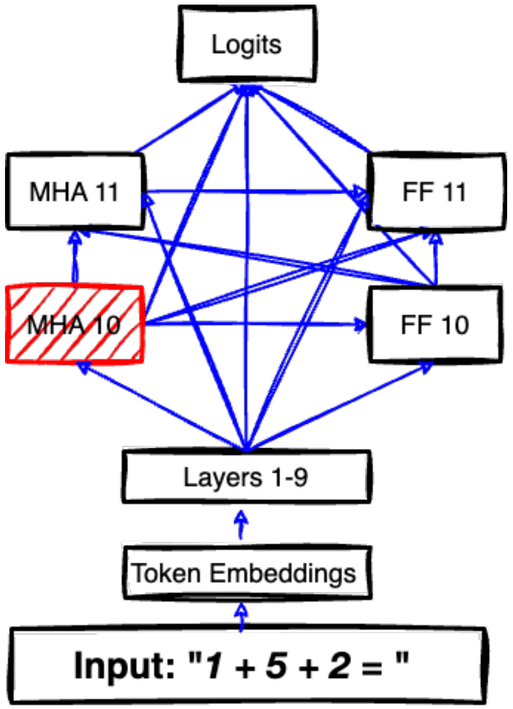


Step 1: Generate Hypothesis (e.g., MHA10 calculates the sum of first and second operand).

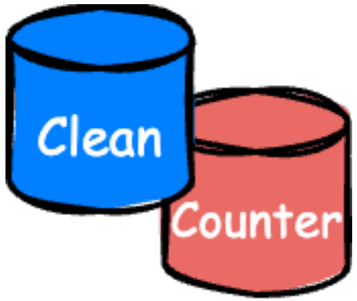
[1] Chan, et al., "Causal Scrubbing: a method for rigorously testing interpretability hypotheses", AI Alignment Forum, 2022.

Method 5: Intervention for Validating Hypotheses

Use case: Validating a hypothesize computation via intervention. Example: Causal Scrubbing (Chen+22)



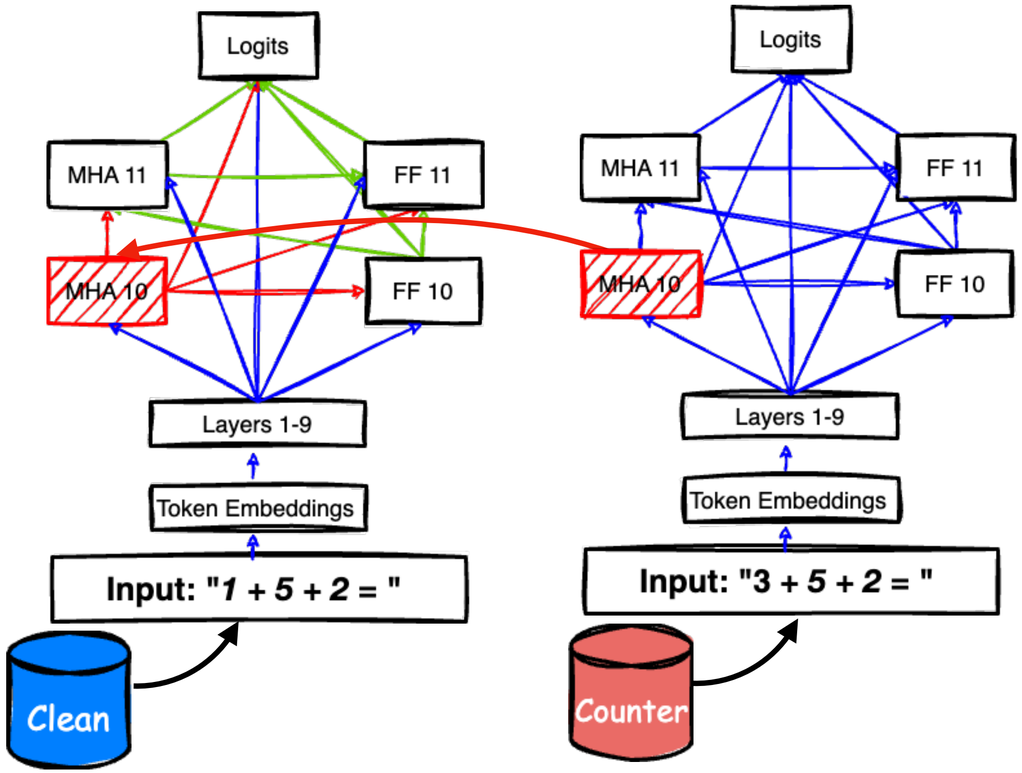
Step 1: Generate Hypothesis (e.g., MHA10 calculates the sum of first and second operand).



Step 2: Dataset with pair of clean and counterfactual examples.

Step 4: Logit difference between correct and counterfactual tokens before and after ablation

If the hypothesis is true, the logit for token "10" will increase after ablation while decrease for token "8"

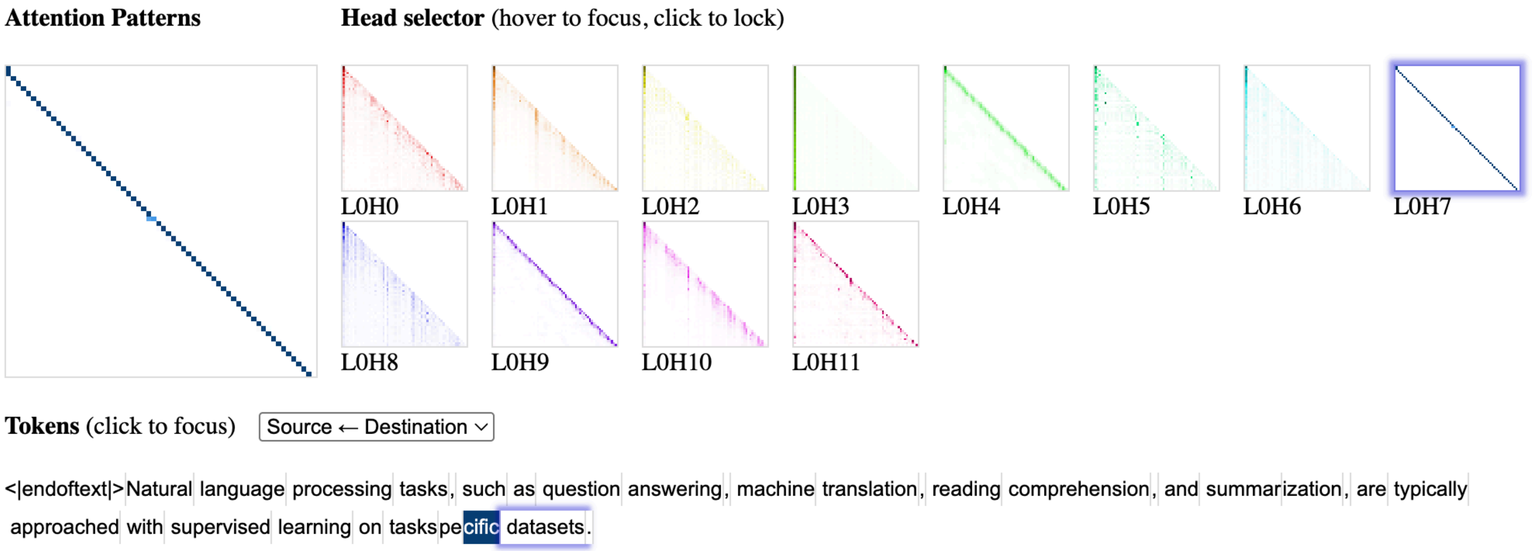


Step 3: Perform resampling ablation.

[1] Chan, et al., "Causal Scrubbing: a method for rigorously testing interpretability hypotheses", AI Alignment Forum, 2022.

Method 6: Attention Visualization

Use case: Understand the role of an attention head and facilitate hypothesis generation
Intuition: Visualizing the attention distribution of a head over texts provides observations for further human explanations



Explain attention visualization

Observation: L0H7 seems to attend previous token position

Hypothesis: L0H7 transfers information about the token on previous token position

Figure: Attention visualization, created using the tool of Cooney & Nanda+23.

Heads up: The role of an attention head should be further validated by other approaches such as intervention and vocabulary projection.

[1] Alan Cooney and Neel Nanda. Circuitsvis. https://github.com/TransformerLensOrg/CircuitsVis, 2023.

Methods for Circuit Study Evaluation

Methods for Circuit Study Evaluation

Circuit study is commonly evaluated by three metrics—faithfulness, minimality, and completeness.

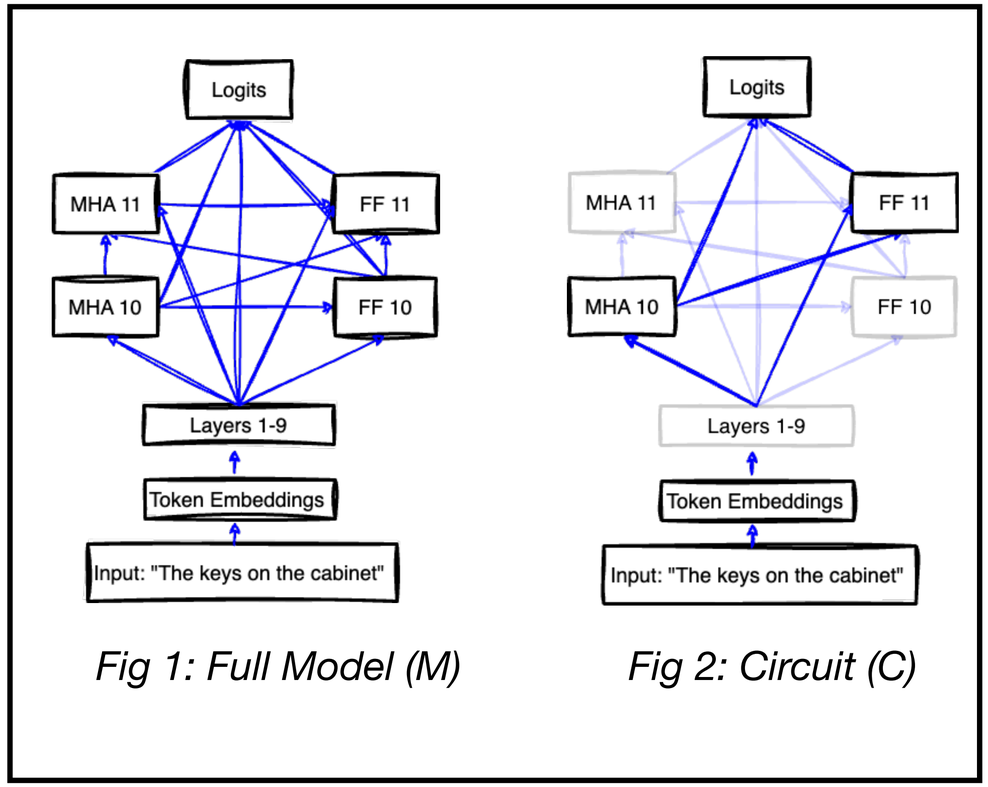


Fig 1: Full Model (M)

Fig 2: Circuit (C)

Faithfulness: The circuit can reproduce the performance of the full model

- Measure the performance gap between full model and the circuit on a test set. **The smaller, the better.**

$$Faithfulness = |F(M) - F(C)|$$

$F(M) =$ Performance of full model M

$F(C) =$ Performance of Circuit C

[1] Kevin Ro Wang, etc., Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *ICLR* 2022.
[2] Hanna, Michael, et al. “How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model.” NeurIPS 2023.

Methods for Circuit Study Evaluation

Circuit study is commonly evaluated by three metrics—faithfulness, minimality, and completeness.

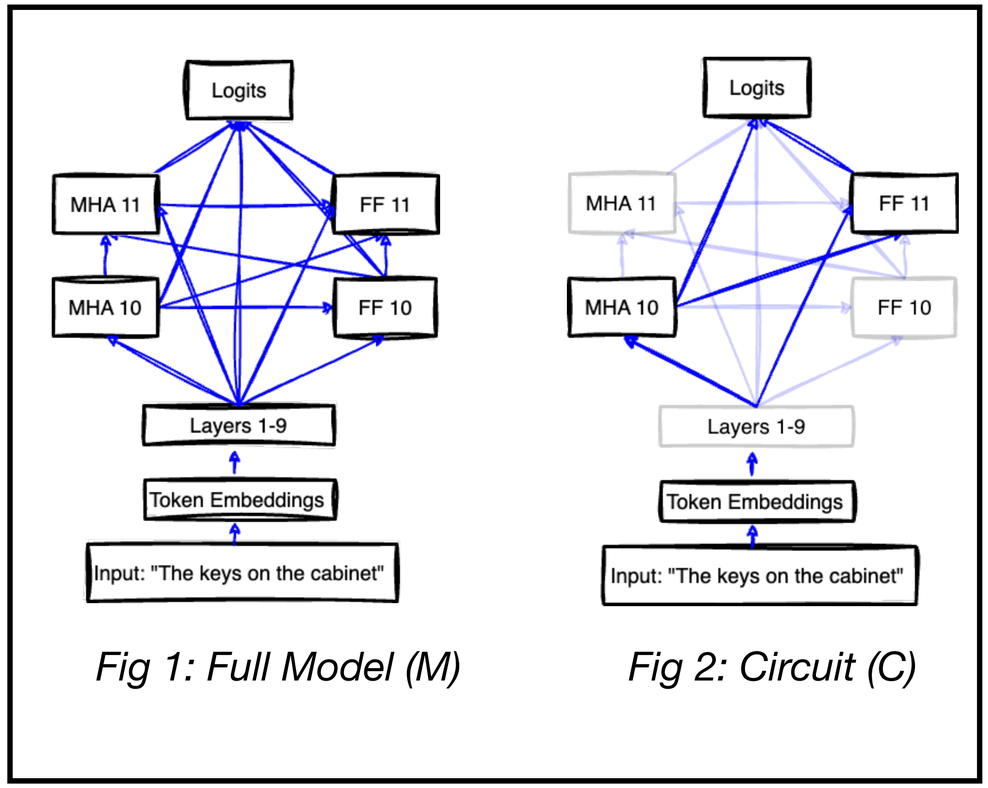


Fig 1: Full Model (M)

Fig 2: Circuit (C)

Minimality: The circuit does not contain unnecessary components or edges

- For each node $v \in C$, check if there exist a subset $K \subseteq C \backslash \{v\}$ that has a **high minimality score**:

$$Minimality(v) = |F(C \backslash (K \cup \{v\})) - F(C \backslash K)|$$

$F(C \backslash (K \cup \{v\}))$ = Performance of circuit C without K and node v

$F(C \backslash K)$ = Performance of circuit without K

[1] Kevin Ro Wang, etc., Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *ICLR* 2022.
[2] Hanna, Michael, et al. “How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model.” NeurIPS 2023.

Methods for Circuit Study Evaluation

Circuit study is commonly evaluated by three metrics—faithfulness, minimality, and completeness.

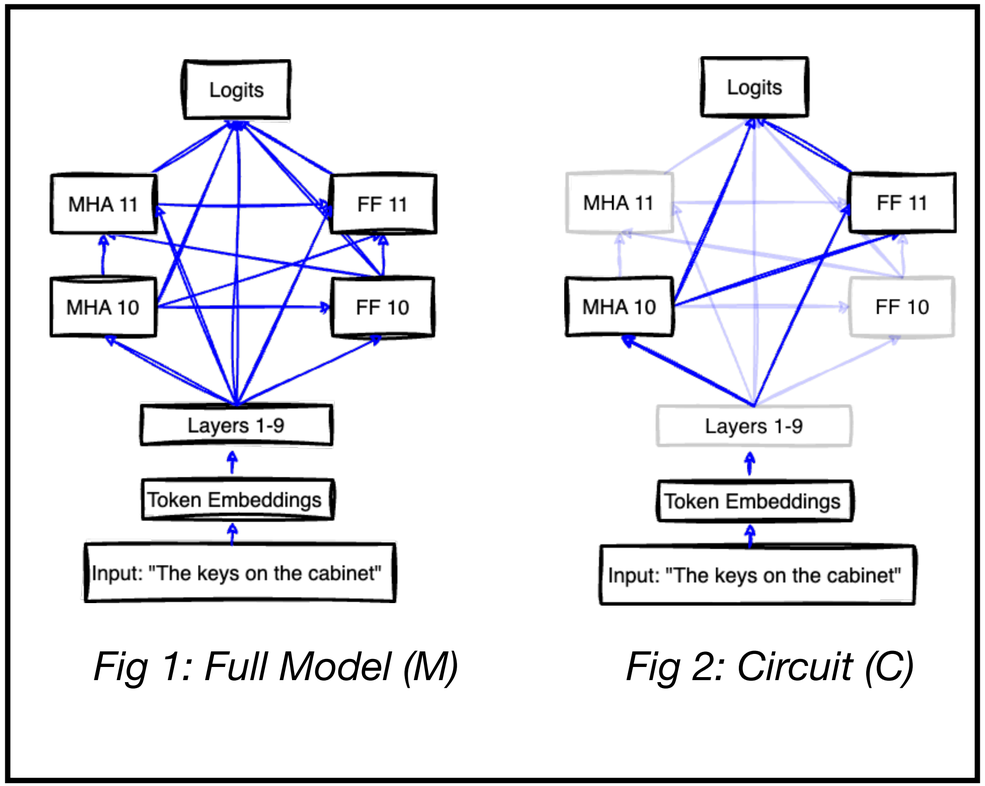


Fig 1: Full Model (M) *Fig 2: Circuit (C)*

Completeness: The circuit contains all the nodes used to perform the task.

- For every subset $K \subseteq C$, the performance difference between $C\backslash K$ and $M\backslash K$ (i.e., "incompleteness score") should be **small**.

$$Incompleteness(K) = |F(C\backslash K) - F(M\backslash K)|$$

$F(C\backslash K)$ = Performance of circuit C without K

$F(M\backslash K)$ = Performance of model M without K

[1] Kevin Ro Wang, etc., Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *ICLR* 2022.
[2] Hanna, Michael, et al. "How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model." NeurIPS 2023.

Examples for Circuit Study

Example of Interpreting an LM behavior (Wang+22)

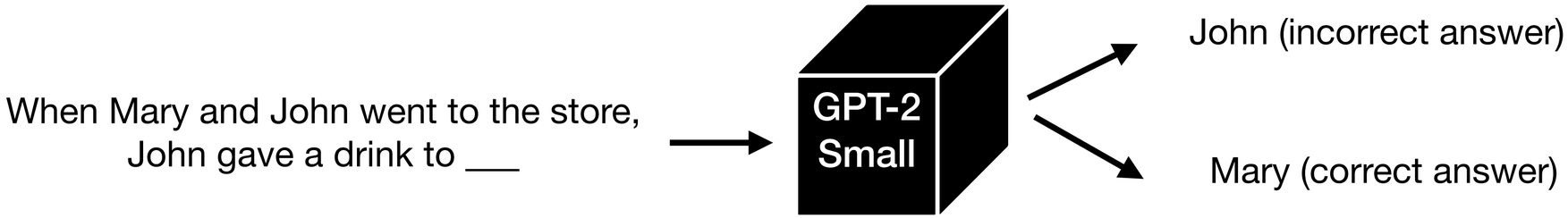
Defining Dataset & Computational Graph → Localization → Interpretation → Evaluation

Create a dataset and clarify the definition of computational graph to study the behavior

Find all important nodes/edges

Understand circuit nodes/edges

e.g., Faithfulness



Wang+22 conducted circuit analysis of the Indirect Object Identification (IOI) task in GPT-2 Small. They assumed that GPT-2 Small consists of a circuit, a sub-network that implements IOI task

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

Step 1&2

Defining Dataset & Computational Graph → Localization → Interpretation → Evaluation

Create a dataset and clarify the definition of computational graph to study the behavior

Find all important nodes/edges

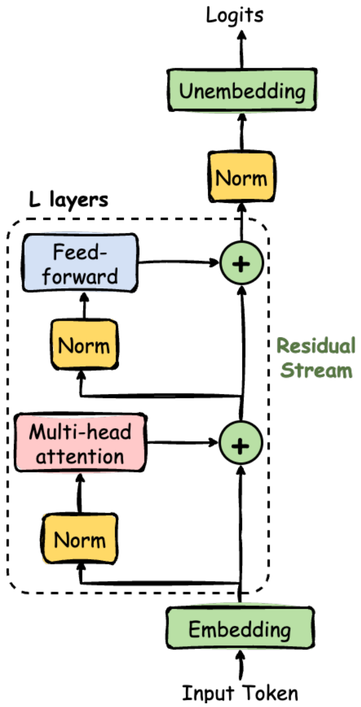
Understand circuit nodes/edges

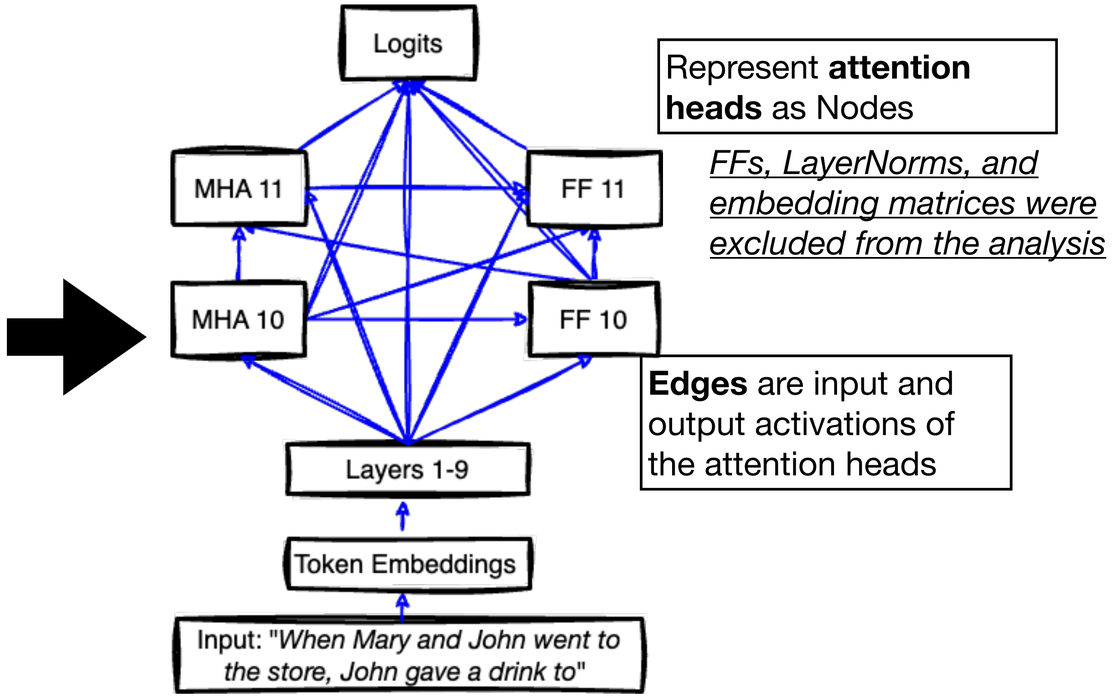
e.g., Faithfulness

Dataset Construction

- 15 sentence templates
- Randomly filled with single-token names, places, and items

Example: *When Mary and John went to the store, John gave a drink to -> Mary*



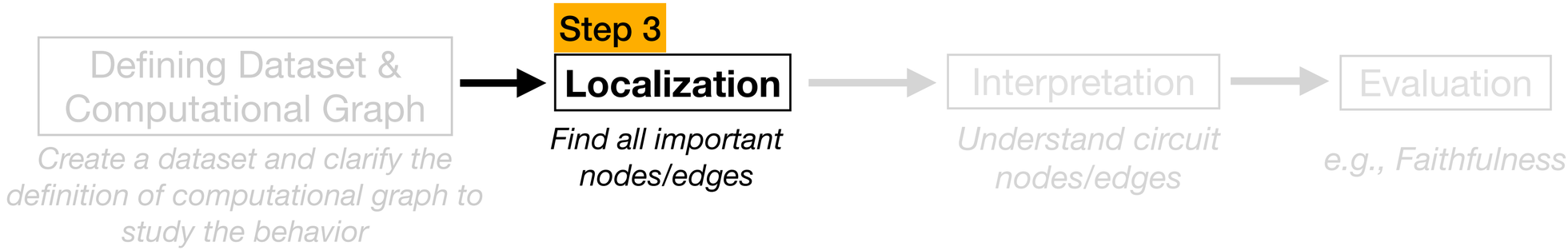


Step 1: synthesize an IOI dataset

Step2: define a computational graph to include position-dependent LM components

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)



Logit-level tracing

Find heads with direct influence on the logits

5 heads

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

Defining Dataset & Computational Graph

Create a dataset and clarify the definition of computational graph to study the behavior

Step 3

Localization

Find all important nodes/edges

Interpretation

Understand circuit nodes/edges

Evaluation

e.g., Faithfulness

Logit-level tracing

Find heads with direct influence on the logits

5 heads

Recursive Tracing

Trace upstream heads that influence the identified logit-affecting components via Edge Patching

13 heads

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

Defining Dataset & Computational Graph

Create a dataset and clarify the definition of computational graph to study the behavior

→

Step 3

Localization

Find all important nodes/edges

→

Interpretation

Understand circuit nodes/edges

→

Evaluation

e.g., Faithfulness

Logit-level tracing

Find heads with direct influence on the logits

5 heads

→

Recursive Tracing

Trace upstream heads that influence the identified logit-affecting components via Edge Patching

13 heads

→

Redundancy Check

Knock out identified components and rerun tracing to reveal backup mechanisms

8 heads

26 attention heads in GPT-2 Small were identified as key contributors to the IOI circuit.

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

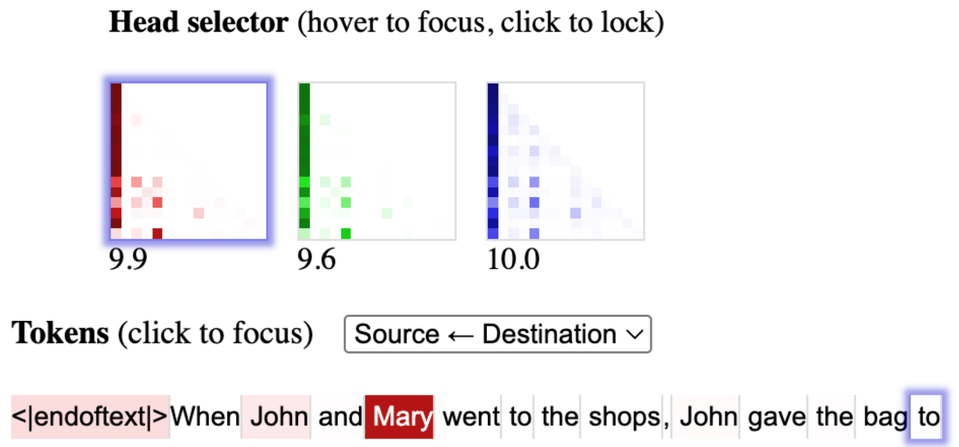
Defining Dataset & Computational Graph
Create a dataset and clarify the definition of computational graph to study the behavior

→ Localization
Find all important nodes/edges

→ **Step 4**
Interpretation
Understand circuit nodes/edges

→ Evaluation
e.g., Faithfulness

Example: Interpreting attention head 9 at layer 9 (L9H9) discovered to be important in the circuit



Observe the attention Visualization

Observation: Attention head L9H9 attends to the indirect object token

Hypothesis: L9H9 is a ***name mover head*** that **attends** to the indirect object token and **copies** whatever it attends to

Step 4.1: Generating a hypothesis

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

Defining Dataset & Computational Graph → Localization → **Step 4** **Interpretation** → Evaluation

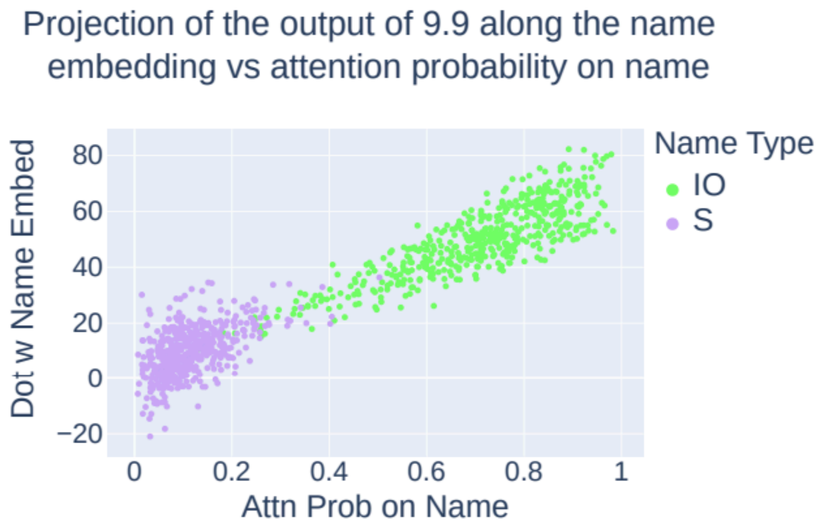
Create a dataset and clarify the definition of computational graph to study the behavior

Find all important nodes/edges

Understand circuit nodes/edges

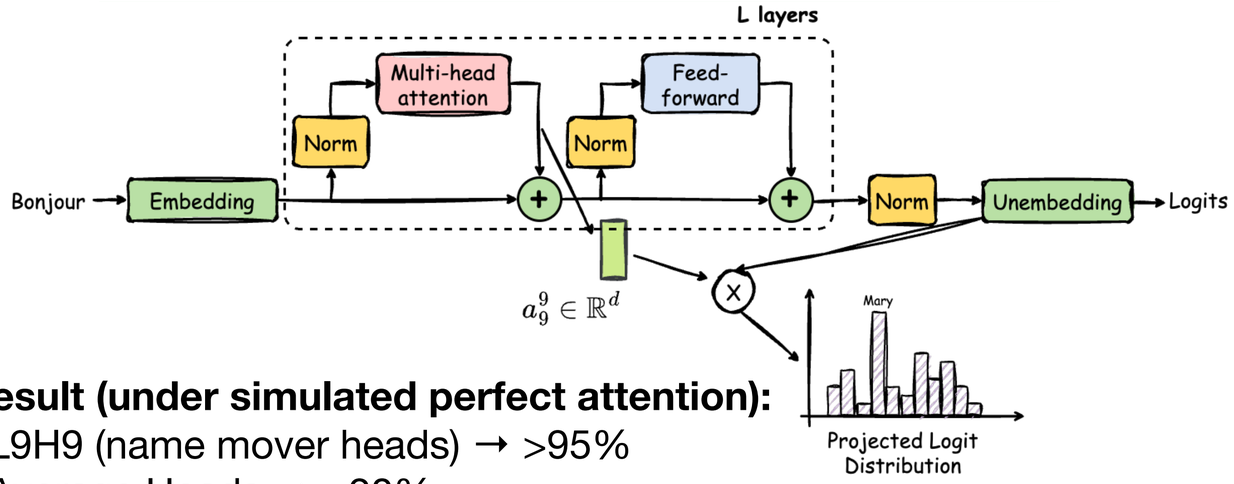
e.g., Faithfulness

Vocabulary Projection, calculating *Logit(IO)*



Validation 1: L9H9 attends to the indirect object (IO) token

Validation 2: L9H9 copies the token it attends to



Result (under simulated perfect attention):

- L9H9 (name mover heads) → >95%
- Average Heads → <20%

Step 4.2: Validating the hypothesis

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Example of Interpreting an LM behavior (Wang+22)

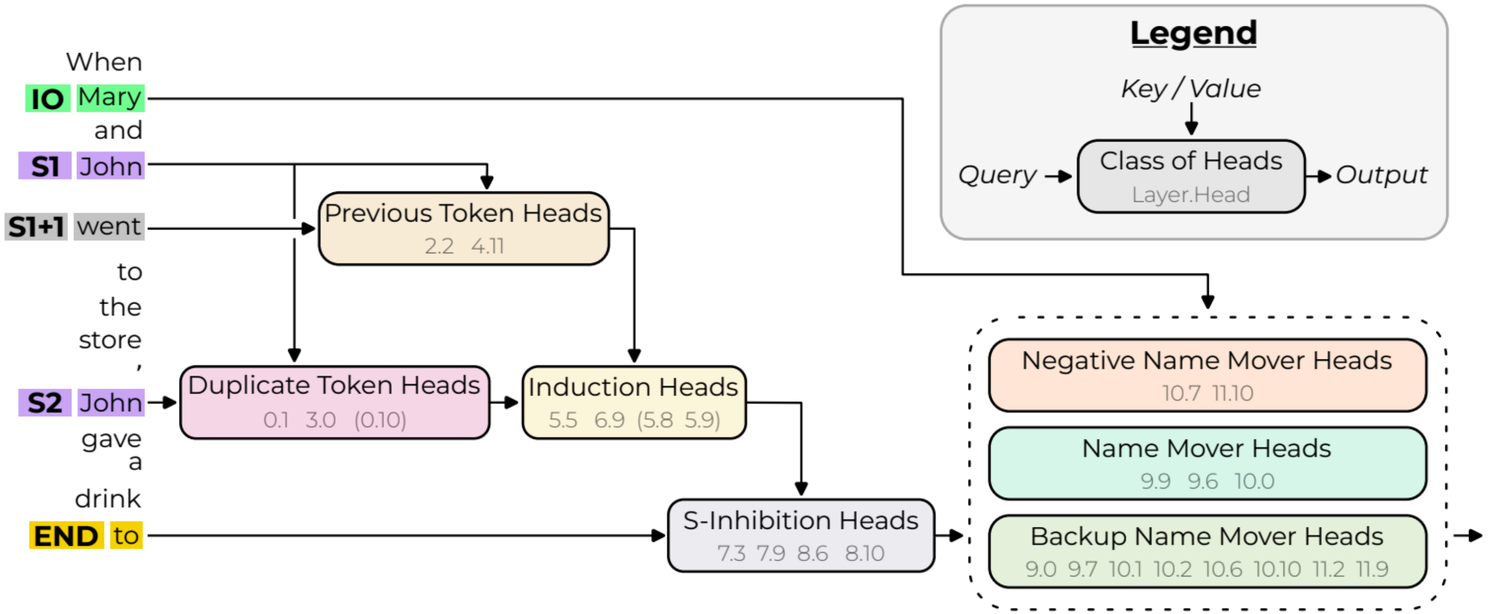
Defining Dataset & Computational Graph → Localization → Interpretation → **Step 5** **Evaluation**

Create a dataset and clarify the definition of computational graph to study the behavior

Find all important nodes/edges

Understand circuit nodes/edges

Faithfulness, Minimality, Completeness



Performance metric $F = Logit(\text{IO}) - Logit(\text{Subj})$

$$Faithfulness = |F(M) - F(C)|$$

Faithfulness: The circuit retains 87% of the full model performance on IOI task

$$Minimality(v) = |F(C \backslash (K \cup \{v\})) - F(C \backslash K)|$$

Minimality: Nodes all have nontrivial impact

$$Incompleteness(K) = |F(C \backslash K) - F(M \backslash K)|$$

Completeness: Low incompleteness score with random sampled K but a high score with greedy sample

[1] Wang, Kevin, et al. “Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.” ICLR 2022.

Part 2.3 Universality Study

Overview of Workflows for Universality Study

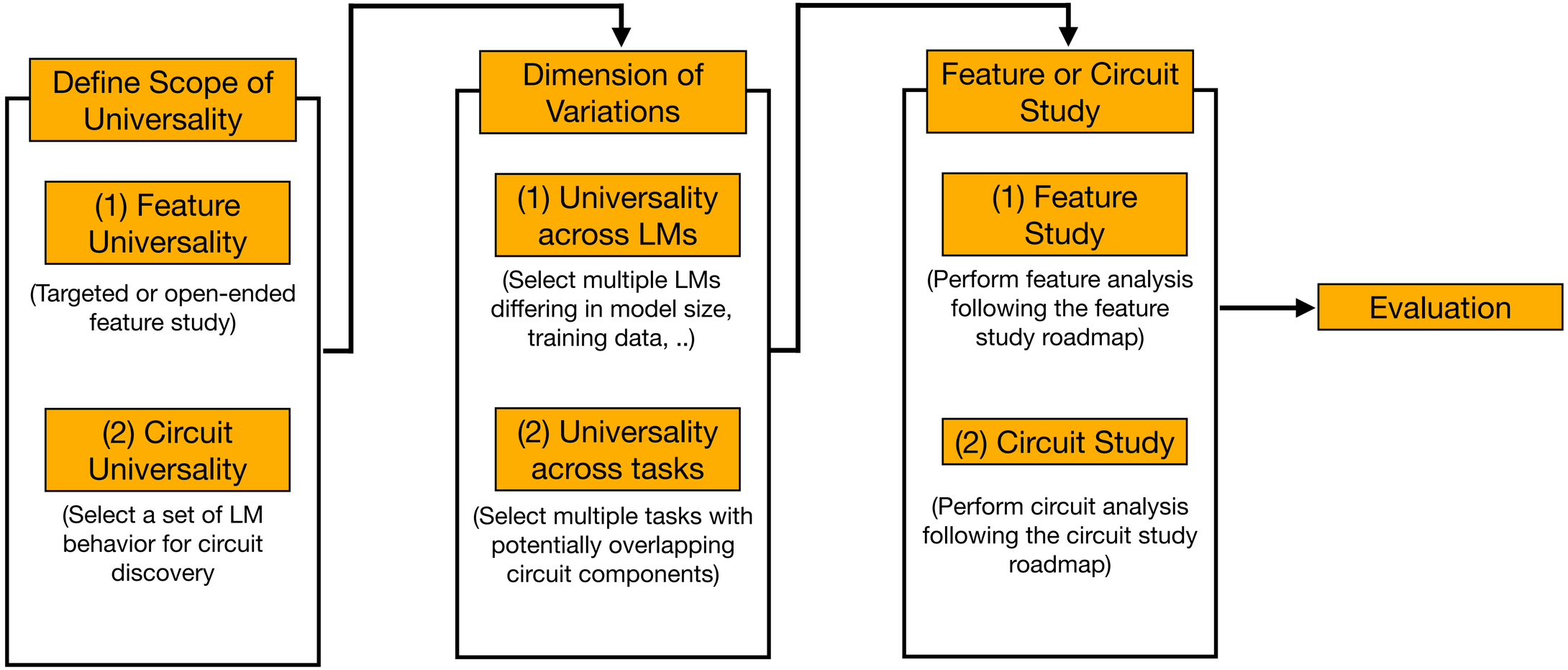


Figure: Task-centric Beginner's Roadmap to Study of Universality

Example of Studying Universality (Gurnee+24)

Step 1

Defining Scope of Study → **Dimensions of Variation** → **Feature or Circuit Study** → **Evaluation**

Univesality of features or circuits

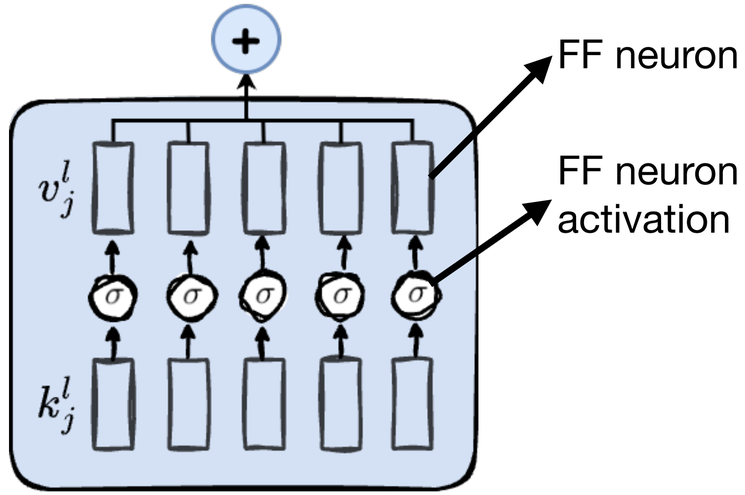
Universality across multiple LMs.

Perform feature study following the feature study roadmap

Feature Study: universality of **FF neurons** in GPT2 models.

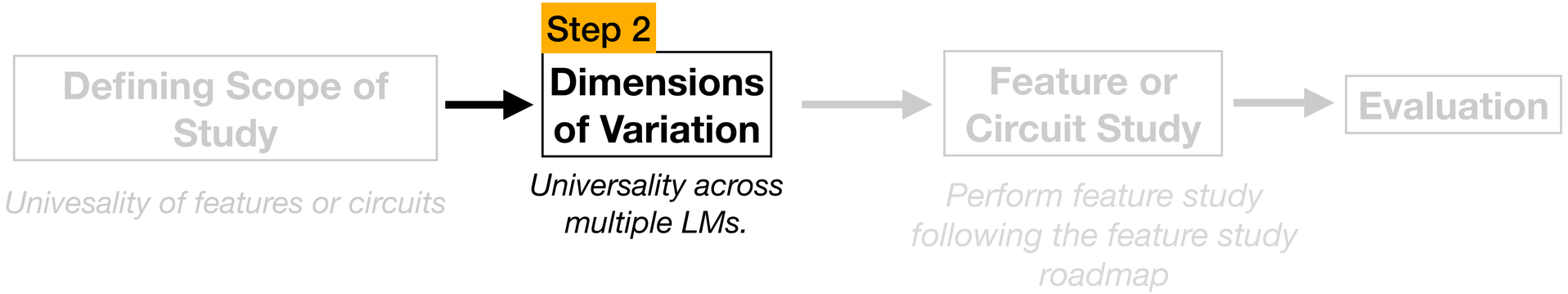
UNIVERSAL NEURONS IN GPT2 LANGUAGE MODELS

Wes Gurnee[1]* **Theo Horsley**[2] **Zifan Carl Guo**[1] **Tara Rezaei Kheirkhah**[1]
Qinyi Sun[1] **Will Hathaway**[1] **Neel Nanda**† **Dimitris Bertsimas**[1]†
[1]MIT [2]University of Cambridge



[1] Gurnee, Wes et al. "Universal neurons in gpt2 language models." TMLR 2024.

Example of Studying Universality (Gurnee+24)



They train **five GPT-2 models** from **different random seeds**, all with the **same architecture** and **dataset**.

[1] Gurnee, Wes et al. “Universal neurons in gpt2 language models.” TMLR 2024.

Example of Studying Universality (Gurnee+24)

Defining Scope of Study → Dimensions of Variation → **Step 3** **Feature or Circuit Study** → Evaluation

Univesality of features or circuits

Universality across multiple LMs.

Perform feature study following the feature study roadmap

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.

business in a small garage to become the world's largest manufacturer of "build-it-yourself" component car kits. They employ a full-time crew of about 40 people, and are located in Wareham, Massachusetts (about an hour south of Boston). They make their products right

Figure: Example of a made-up (dummy) neuron activation over text

Collect neuron activations over 100M tokens from the Pile test set for feature analysis (Gao+20).

[1] Gurnee, Wes et al. “Universal neurons in gpt2 language models.” TMLR 2024.
[2] Gao, L., Biderman et al. “The pile: An 800gb dataset of diverse text for language modeling” arXiv preprint arXiv:2101.00027 (2020).

Example of Studying Universality (Gurnee+24)

Defining Scope of Study → Dimensions of Variation → **Feature or Circuit Study** → Step 4 **Evaluation**

Univesality of features or circuits

Universality across multiple LMs.

Perform feature study following the feature study roadmap

For each **neuron**, they compute **Pearson correlation coefficient** with all neurons from other models

$$p_{i,j}^{a,m} = \frac{\mathbb{E}[(v^i - \mu_i)(v^j - \mu_j)}{\sigma_i \sigma_j}$$

Where, a and $m = \{b, c, d, e\}$ are models

μ_i and σ_i are mean and standard deviations of the vector of neuron activations v^i

Only **1–5% of neurons** in a model were found to be universal

! Negative Results for Universality of features

$p_{i,j}^{a,m}$ a **similarity score**: how similarly neuron i in model a and neuron j in model m respond to the same input.

Universal neurons: neurons with highly similar activation patterns across models, exceeding a defined similarity threshold.

[1] Gurnee, Wes et al. “Universal neurons in gpt2 language models.” TMLR 2024.