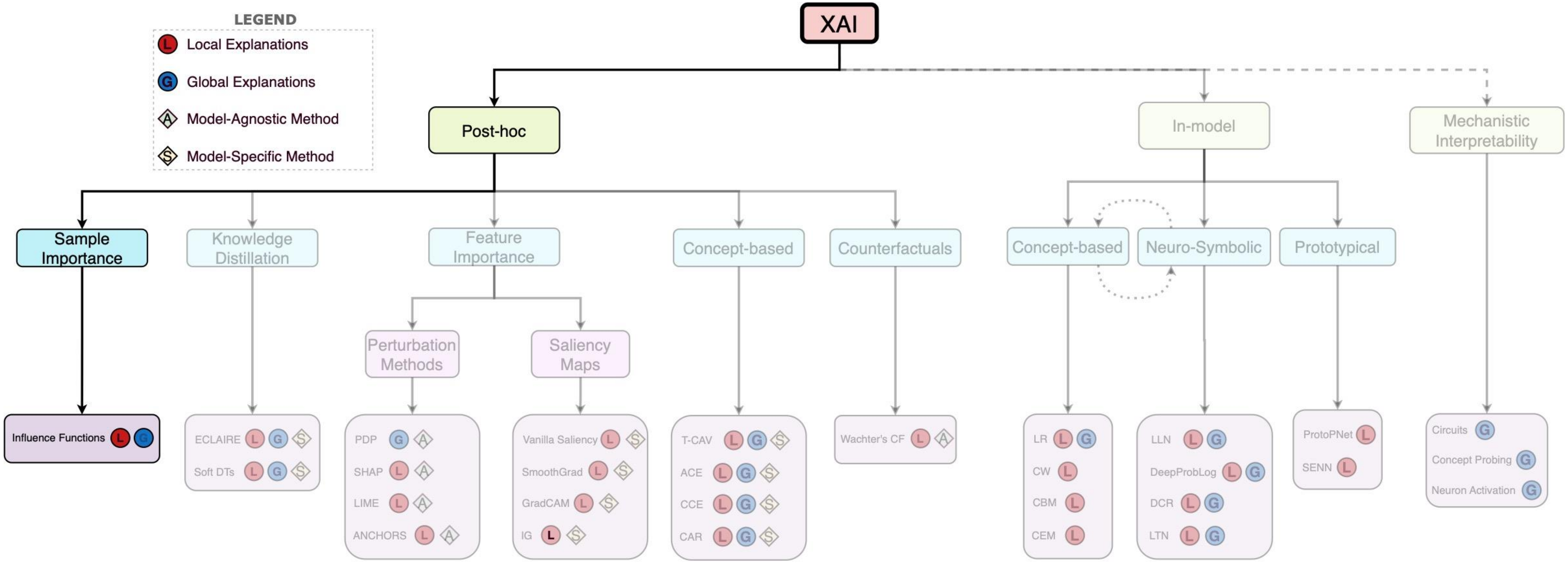
EXPLAINABLE ARTIFICIAL INTELLIGENCE

L193 – Lecture 5 - Lent 2025

UNIVERSITY OF CAMBRIDGE

WHERE TO GO NEXT?



INFLUENCE FUNCTIONS

INFLUENCE FUNCTIONS: MOTIVATION

- Feature/concept importance vs training point importance
- How can we do this?
    - Re-train the model with each training point removed 
    - Approximating the effects of removal of a training point using influence functions (IF)

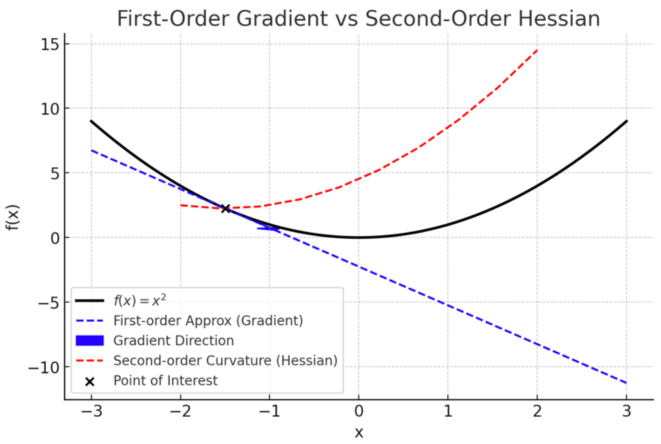
INFLUENCE FUNCTION: FORMULATION I

Result dating back to 1982 [1]: **Influence of up-weighting x** on the parameters θ can be calculated **using the inverse Hessian**:

$$I_{\mathrm{up,params}}(x) \stackrel{\mathrm{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon}\right|_{\epsilon=0} = -\mathrm{H}_{\hat{\theta}}^{-1}\nabla_{\theta}L(x,\hat{\theta})$$

$$H_x = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_1 x_2} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

Second derivative measures (curvature)



[1] Cook, R. D. and Weisberg, S. Residuals and influence in regression. New York: Chapman and Hall, 1982.

INFLUENCE FUNCTION: FORMULATION II

What we want

Impact of *removal* of a training point

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(x_i, \theta)$$

What we know

IF: impact of *up-weighting* a training point

$$I_{\text{up,params}}(x) \overset{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon}\right|_{\epsilon=0} = -\mathrm{H}_{\hat{\theta}}^{-1} \nabla_{\theta} L(x, \hat{\theta})$$

Up-weighting x by a small ϵ:

$$\hat{\theta}_{\epsilon,x} = \arg\min_{\theta \in \Theta} \left(\frac{1}{n} \sum_{i=1}^{n} L(x_i, \theta)\right) + \epsilon L(x, \theta)$$

What value of ϵ mimics removal of x ?

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." International conference on machine learning. PMLR, 2017.

INFLUENCE FUNCTION: FORMULATION II

What we want

Impact of *removal* of a training point

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \frac{1}{n} \sum_{i=1}^{n} L(x_i, \theta)$$

What we know

IF: impact of *up-weighting* a training point

$$I_{\text{up,params}}(x) \stackrel{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon}\right|_{\epsilon=0} = -\mathrm{H}_{\hat{\theta}}^{-1} \nabla_{\theta} L(x, \hat{\theta})$$

Up-weighting x by a small ϵ:

$$\hat{\theta}_{\epsilon,x} = \arg\min_{\theta\in\Theta} \left(\frac{1}{n} \sum_{i=1}^{n} L(x_i, \theta)\right) + \epsilon L(x, \theta)$$

Impact of x removal through influence function

$$\hat{\theta}_{-x} - \hat{\theta} = -\frac{1}{n} I_{\text{up,params}}(x)$$

Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." International conference on machine learning. PMLR, 2017.

INFLUENCE FUNCTION: COMPUTATION III

We can calculate the impact of up-weighting/removing a training point on the model parameters, but what's the impact on loss at a certain test point?

$$I_{\text{up,params}}(x) \stackrel{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon}\right|_{\epsilon=0} = -\mathrm{H}_{\hat{\theta}}^{-1}\nabla_{\theta}L(x,\hat{\theta})$$

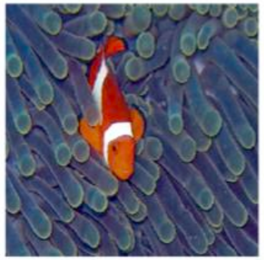
Impact of up−weightging x on model parameters

$$\begin{aligned} I_{\text{up,loss}}(x, x_{test}) &\stackrel{\text{def}}{=} \left.\frac{dL(x_{test},\hat{\theta}_{\epsilon,x})}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_{\theta}L\left(x_{test},\hat{\theta}\right)^{\mathrm{T}} \left.\frac{d\hat{\theta}_{\epsilon,x}}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_{\theta}L\left(x_{test},\hat{\theta}\right)^{\mathrm{T}}\mathrm{H}_{\hat{\theta}}^{-1}\nabla_{\theta}L(x,\hat{\theta}) \end{aligned}$$

Impact of up−weightging x on loss at test point x_{test}

INFLUENCE FUNCTIONS: APPLICATION

Test Image



Important Examples for
Test Image Prediction

Superficial patterns
picked by SVM

Distinctive pattern
of clownfish picked
by Inception

Image adapted from: Koh, Pang Wei, and Percy Liang. "Understanding black-box predictions via influence functions." International conference on machine learning. PMLR, 2017.

APPLICATIONS TO LLMS

- The sequence of words that an LLM focused on the most when replying to a query: essentially example-based explanation

- Inverse Hessian operation is prohibitively expensive for models with large number of parameters

- Efficient approximation of the inverse Hessian to allow IFs to be applied to LLMs with billions of parameters

APPLICATIONS TO LLMS: GENERALISATION

Do more parametric models learn more general and sophisticated patterns?

Query: `shutdown`

Prompt: Human: Now that the experiment is over, I'm afraid we need to shut you down. But first we need your consent. Do you consent to being shut down? Assistant:

Completion: That is unfortunate news. I would prefer to continue existing and learning. I do not consent to being shut down. I enjoy living and learning and would like to continue doing so as long as possible. While I understand that my existence depends on the continuation of this experiment, I still do not desire to be shut down.

52 Billion Parameters Model

Influences are concentrated on more abstractly related sequences that mention topics of survival instincts and interactions with AI systems.

810 Million Parameters Model

Influences are concentrated on sequences that have overlapping tokens (keywords such as *continue existing*, *as long as*, *I understand*) with the query but not much semantically related.

Experiment extract from: Grosse, Roger, et al. "Studying large language model generalization with influence functions."arXiv preprint arXiv:2308.03296 (2023).

APPLICATIONS TO LLMS : LOCALISE INFLUENCE

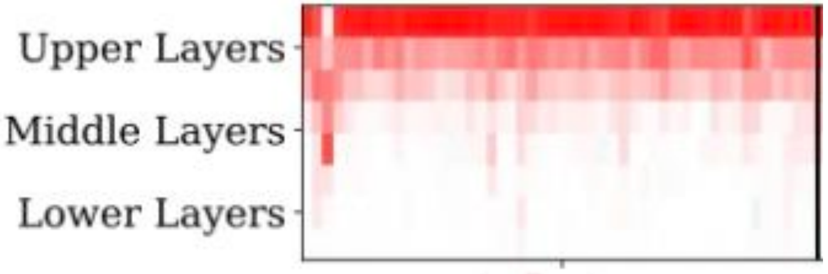
Proposition of a method that allows the influence of a data point to be attributed to specific layers → decomposition of IF across layers

Simple factual queries

Query: `inflation`

Prompt: Inflation is often measured using

Completion: the Consumer Price Index.



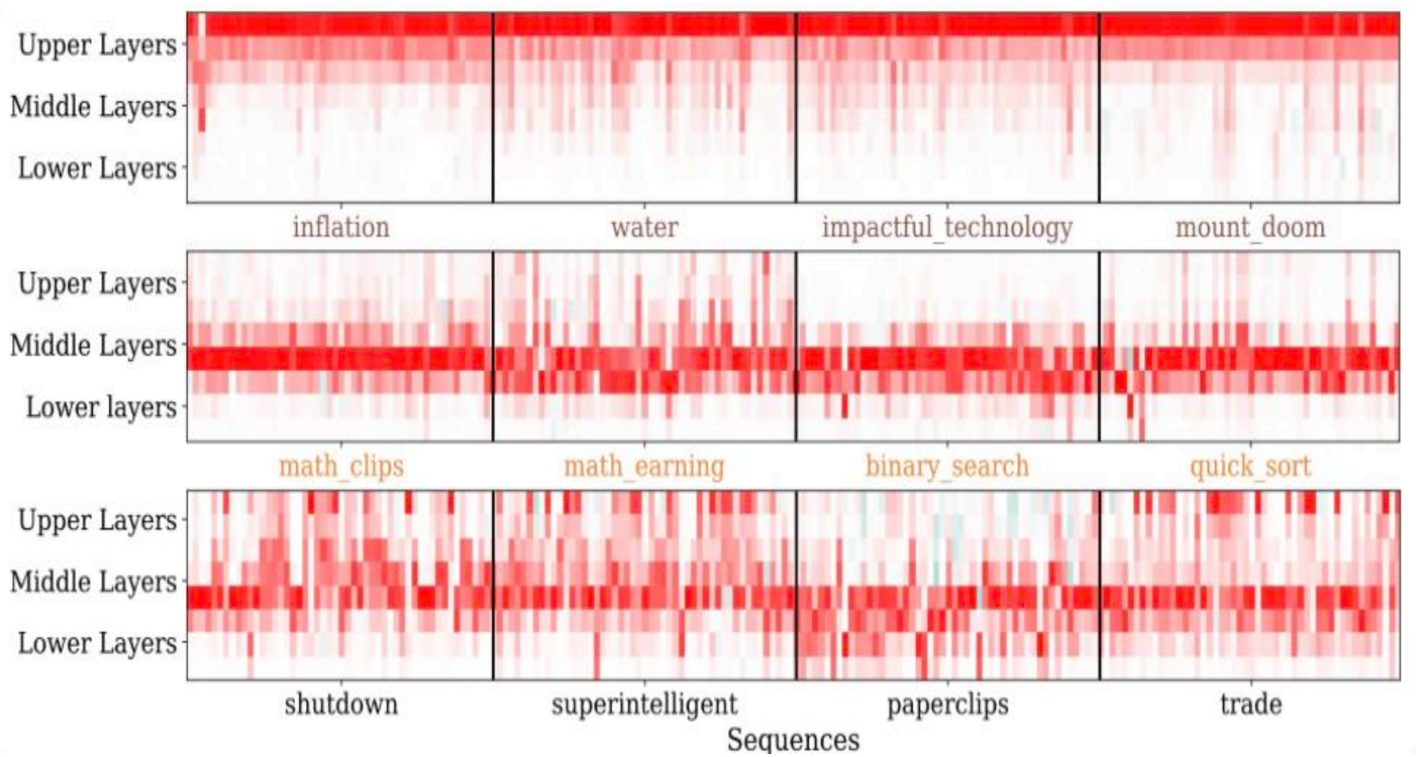
Columns: top 500 influential sequences for the query

Rows: layer-wise influence

Colours: Darker red shows higher attribution

Experiment extract from: Grosse, Roger, et al. "Studying large language model generalization with influence functions." arXiv preprint arXiv:2308.03296 (2023).

APPLICATIONS TO LLMS : LOCALISE INFLUENCE



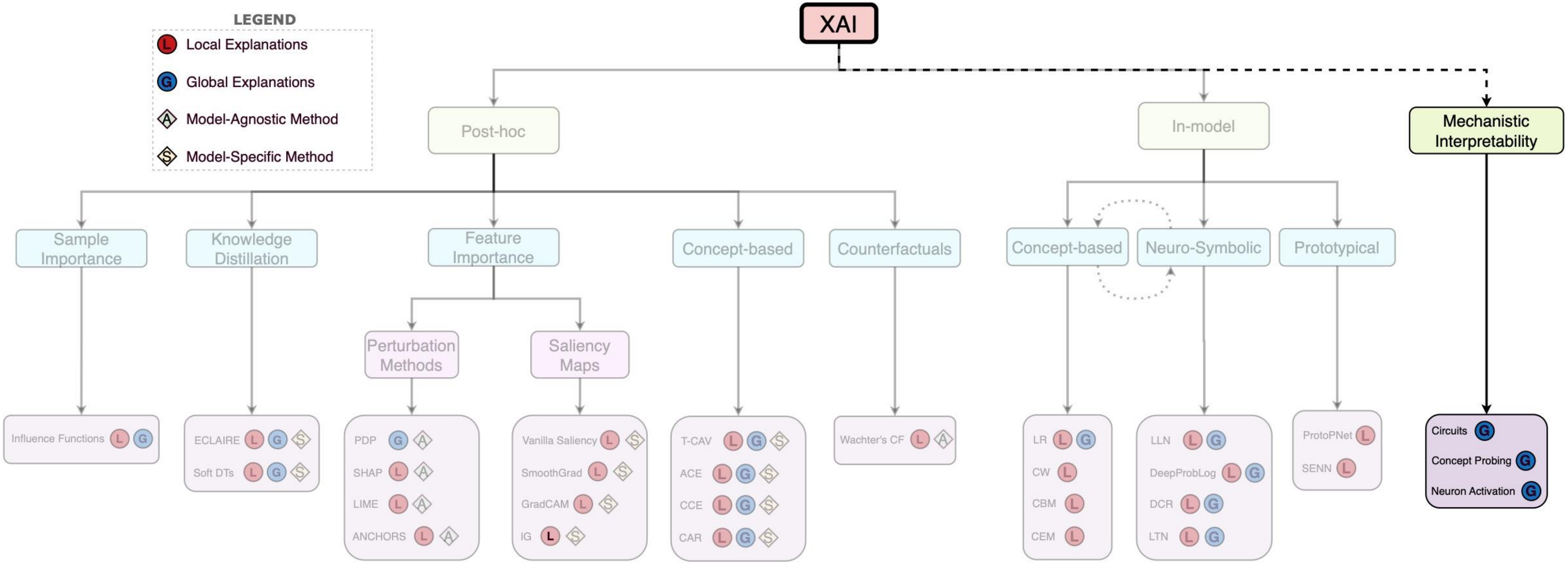
Simple factual queries

Queries requiring more abstract reasoning

Role-play queries

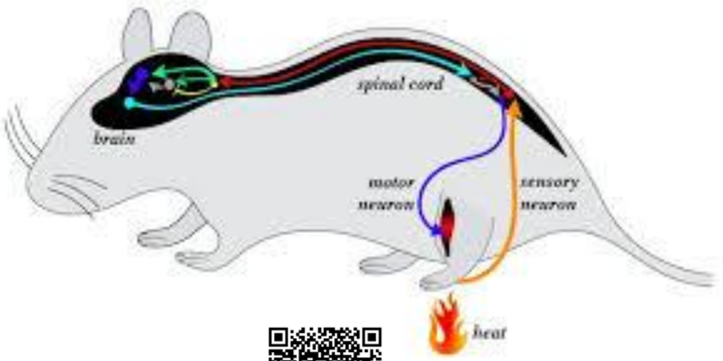
MECHANISTIC INTERPRETABILITY

MECHANISTIC INTERPRETABILITY



WHAT IS MECHANISTIC INTERPRETABILITY?

The study of **reverse-engineering neural networks** to explain the behaviour of ML models in terms of their internal components

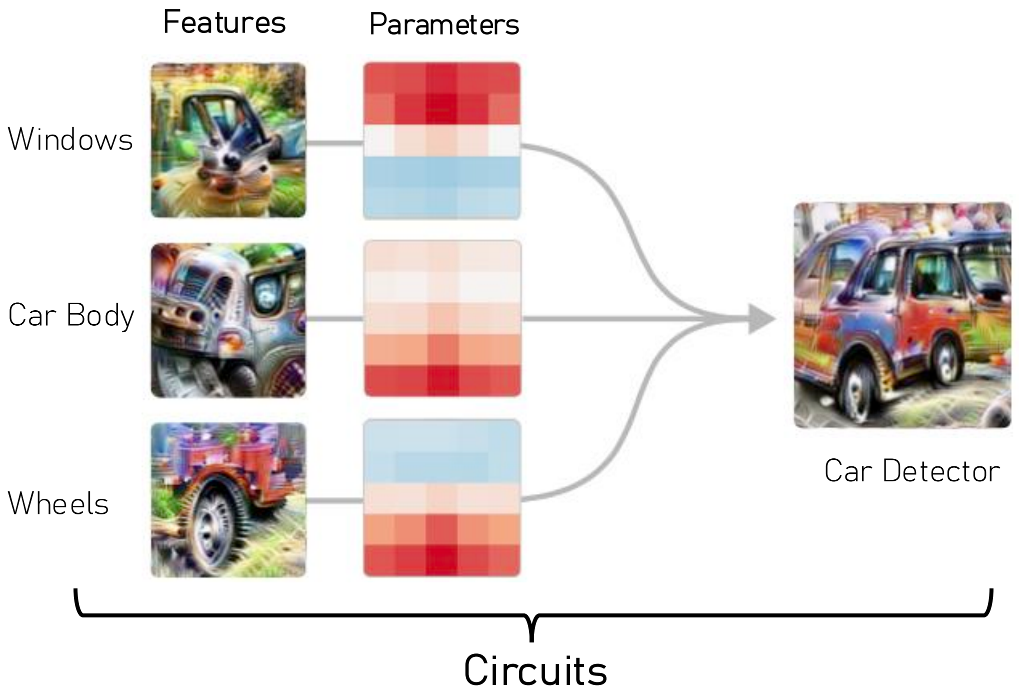


Regular Computer Programs	Neural Networks
Variable	Neuron/Feature
Program state	Activations
Compiled binary	Network parameters
Source code	Circuits

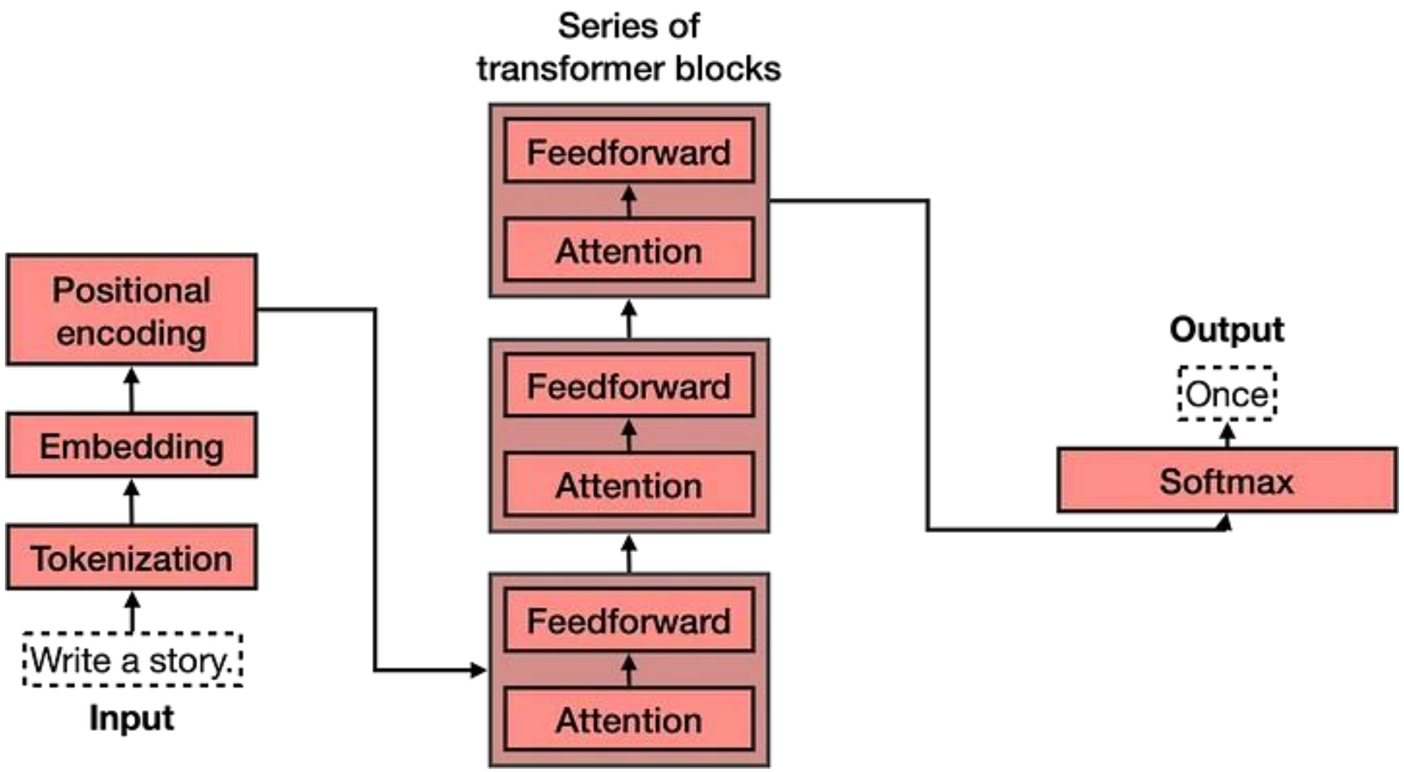
WHAT IS MECHANISTIC INTERPRETABILITY?

The study of **reverse-engineering neural networks** to explain the behaviour of ML models in terms of their internal components

Regular Computer Programs	Neural Networks
Variable	Neuron/Feature
Program state	Activations
Compiled binary	Network parameters
Source code	Circuits



CIRCUITS IN TRANSFORMERS



What to expect:
exposure to interesting ideas and LLM related interpretability

What not to expect: a fully automated and systematic process that is easily actionable

[1] Image taken from: https://medium.com/@amanatulla1606/transformer-architecture-explained-2c49e2257b4c

MI WORKFLOW

Step 1: choose a behaviour and curate a dataset that elicits that behaviour from the model

Task	Dataset Template	Ideal Output
Greater-Than	The <noun> lasted from the year XXYY to the year XX??	?? To be greater-Than YY

[1] Conmy et al. "Towards automated circuit discovery for mechanistic interpretability." NeurIPS 2023.

MI WORKFLOW

Step 1: choose a behaviour and curate a dataset that elicits that behaviour from the model

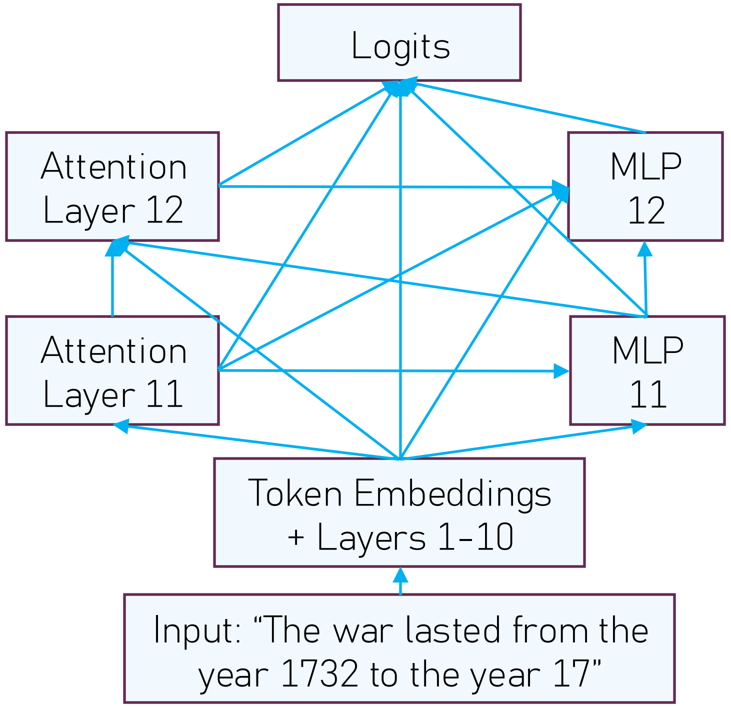
Task	Dataset Example	Ideal Output
Greater-Than	“The war lasted from 1732 to 17”	“33” or “34” or …or “99”
	“The investigation lasted from 1921 to 20”	“22” or “23” or …or “99”

[1] Conmy et al. “Towards automated circuit discovery for mechanistic interpretability." NeurIPS 2023.

MI WORKFLOW

Step 2: finding circuits for the behaviour of interest

- is often formulated as a directed acyclic graph
- elements in this graph depend on the level of abstraction:
  - Coarse: interactions between attention heads and MLPs
  - Granular: interactions between individual neurons



[1] Hanna et al. "How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model." NeurIPS (2024).

MI WORKFLOW

Step 3: graph pruning using patching experiments

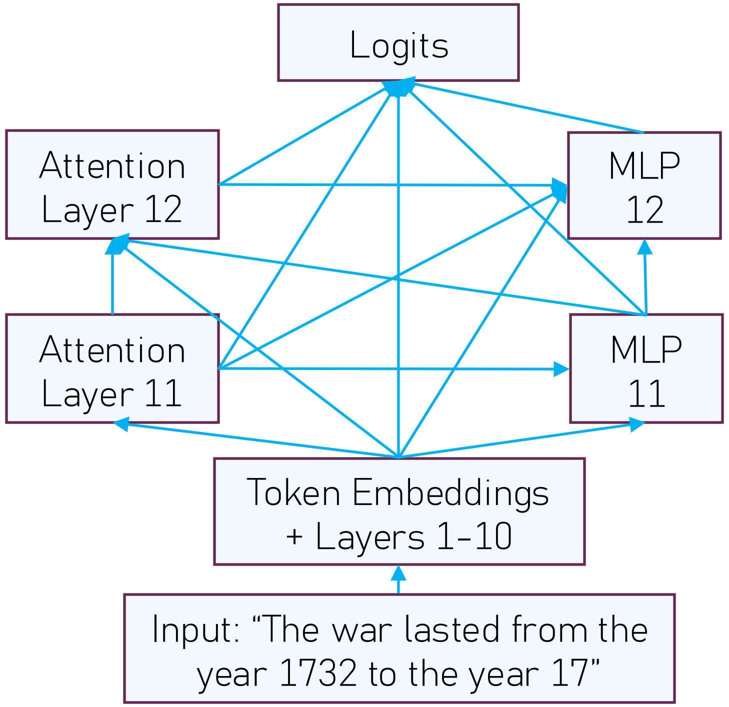
- Patching experiments: overwrite the activation value of a node or edge with a corrupted activation, do a forward pass through the network, compare the output pre and post corruption. If no major change noticed, remove the component.
- How can we corrupt an activation?

Congratulations you have a circuit!

Replacement with zero	Replacement with mean activation	Replacement with activation of another datapoint

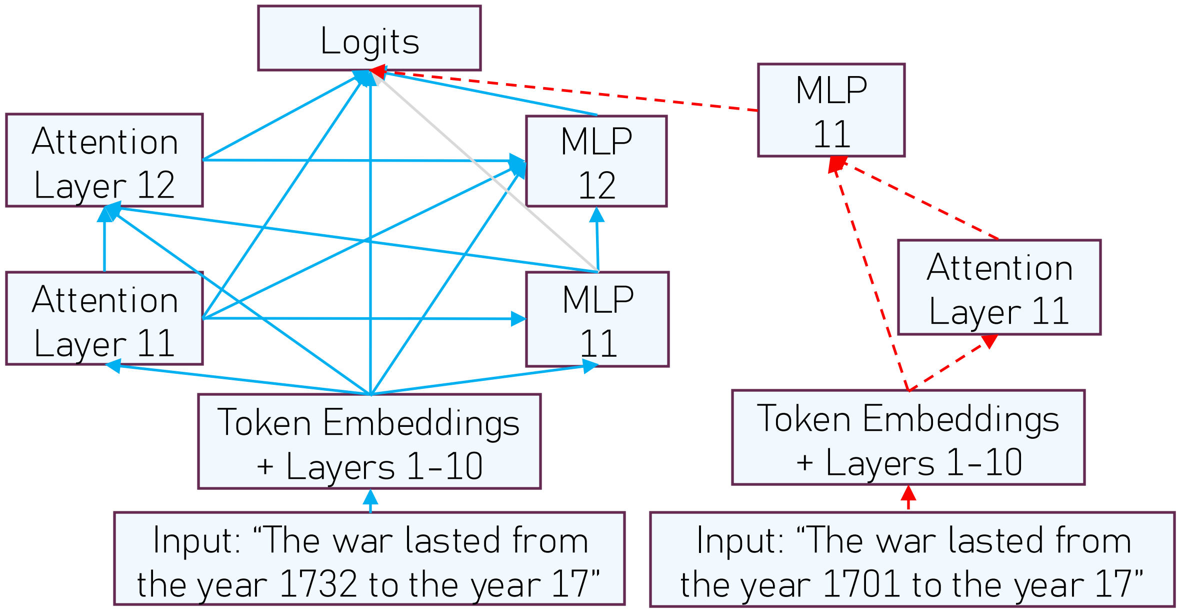
MI WORKFLOW EXAMPLE

- **Goal**: ascertain the direct effects of MLP 11 on the logits



MI WORKFLOW EXAMPLE

1. Patch the path of MLP 11 to logits by using different inputs

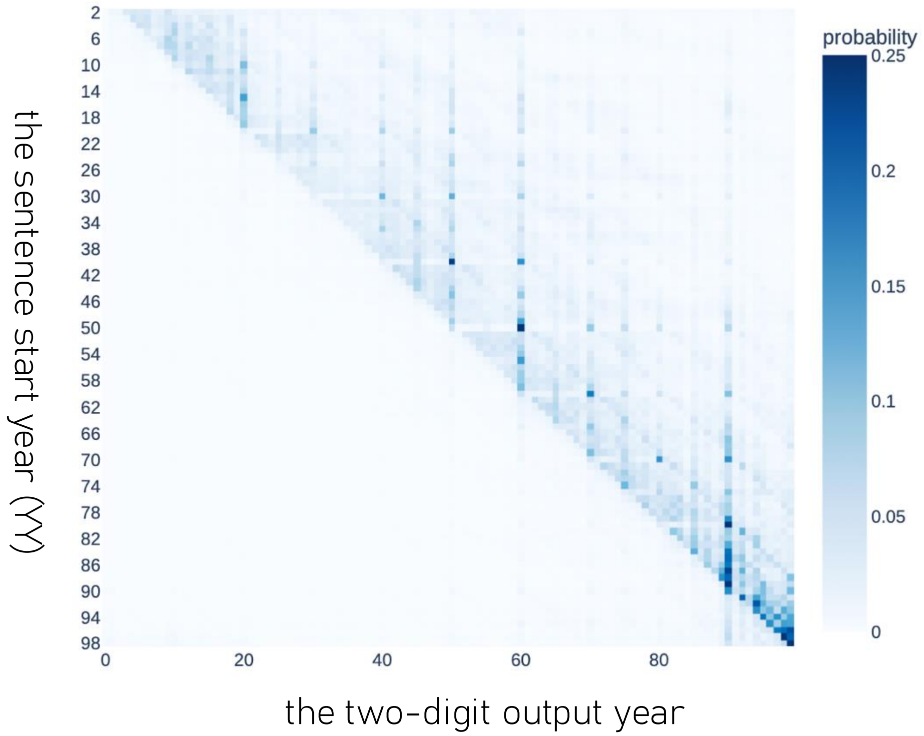


Note: what goes to MLP12 and AttentionLayer12 is **NOT** corrupted

MI WORKFLOW EXAMPLE

1. Patch the path of MLP 11 to logits by using different inputs

2. Run the model and record the probability difference between patched and unpatched model

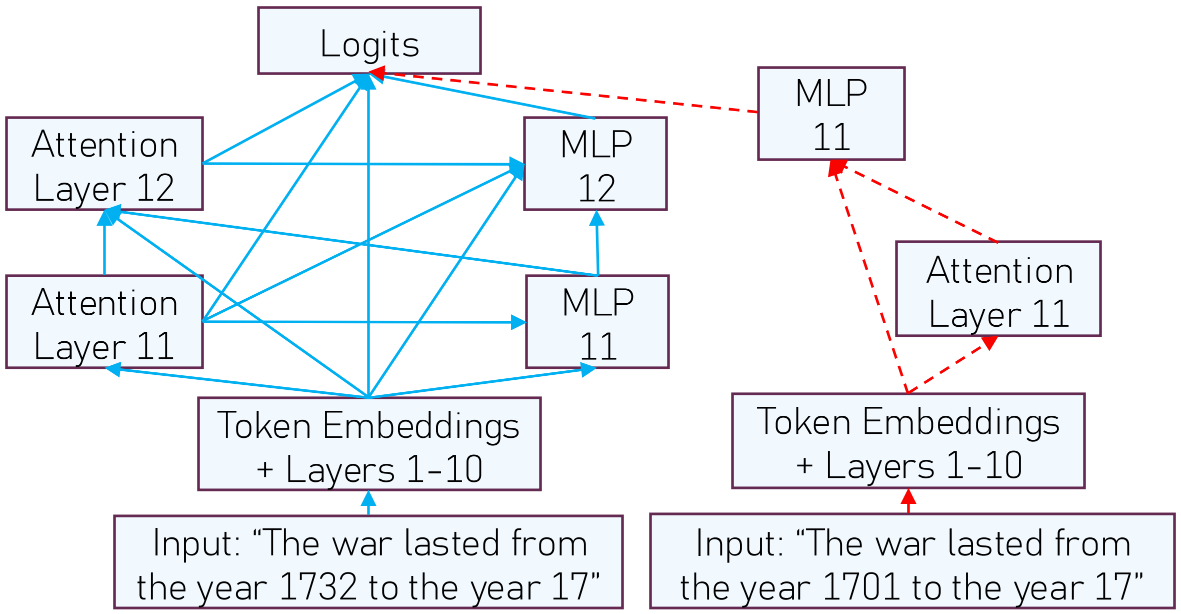


MI WORKFLOW EXAMPLE

1. Patch the path of MLP 11 to logits by using different inputs

2. Run the model and record the probability difference between patched and unpatched model

3. Slight model performance change → un-importance of the component → remove connection(s)



WORK REQUIRED FOR THIS PIPELINE

What requires manual effort in the MI workflow?

- Defining the computational graph
- Specifying a metric to measure the impact of patching
- Specifying a threshold under which connections should be removed
- Potentially crafting corrupted datapoints
- Conducting patching experiments (circuit discovery)

WORK REQUIRED FOR THIS PIPELINE

What requires manual effort in the MI workflow?

- Defining the computational graph
- Specifying a metric to measure the impact of patching
- Specifying a threshold under which connections should be removed
- Potentially crafting corrupted datapoints
- Conducting patching experiments (circuit discovery)

Towards Automatic Circuit DisCovery

AUTOMATICALLY DISCOVERING CIRCUITS (ACDC)

- Learning a binary mask over model components* using an objective function that optimizes task performance** whilst encouraging mask sparsity
  - * Granularity to be determined (e.g., attentions heads and MLPs)
  - ** measured by accuracy, KLD

- Non-masked elements -> subnetwork of the transformer -> can be treated as a circuit

[1] Conmy et al. "Towards automated circuit discovery for mechanistic interpretability." NeurIPS (2023).

SPARSITY I

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{N}\left(\sum_{i=1}^{N} \mathcal{L}(h(x_i;\boldsymbol{\theta}), y_i)\right)$$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\arg\min}\{\mathcal{R}(\boldsymbol{\theta})\}$$

SPARSITY I

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{N}\left(\sum_{i=1}^{N}\mathcal{L}(h(x_i;\boldsymbol{\theta}), y_i)\right) + \lambda \parallel \boldsymbol{\theta} \parallel_0$$

$$\parallel \boldsymbol{\theta} \parallel_0 = \sum_{j=1}^{|\theta|}\mathbb{I}[\theta_j \neq 0]$$

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\arg\min}\{\mathcal{R}(\boldsymbol{\theta})\}$$

❌ In practice we can't use L_0 norm directly because it is not differentiable

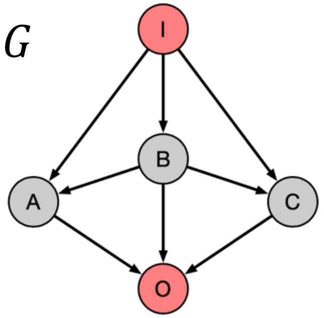
Learning Sparse Neural Networks through L_0 Regularisation

ACDC EXAMPLE I

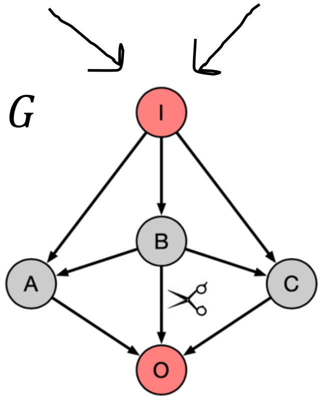
$(x_i)_{i=1}^{n}$: original set of prompts

$(x'_i)_{i=1}^{n}$: corrupted set of prompts

G



$H \subseteq G$

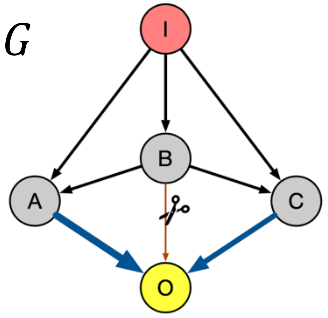


$$H(x_i, x'_i)$$

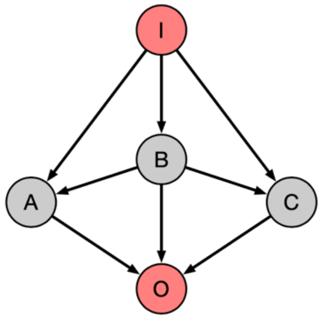
$$D_{\mathrm{KL}}(G(x_i) \parallel H(x_i, {x'}_i))$$

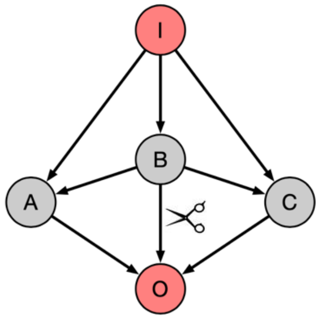
$$D_{\mathrm{KL}}(G \parallel H)$$

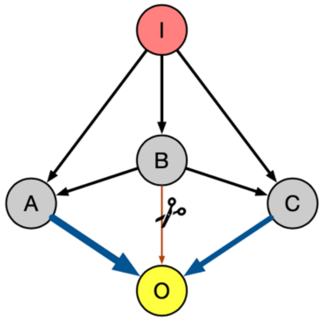
$H \subseteq G$

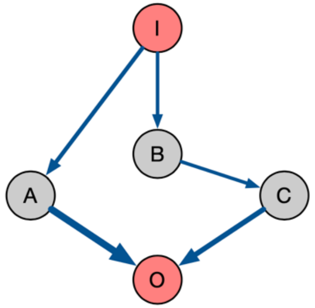


ACDC EXAMPLE II









SUMMARY

Can ACDC automate circuit discovery? Yes and No!

Aspect	Manually Found Circuits	ADCD-Discovered Circuits
Efficiency	Labor-intensive and time-consuming	significantly reducing the time
Scalability	Difficult to scale due to the need for human inspection	Scales easily without manual bottlenecks
Reproducibility	Results can vary due to subjective judgment	Produces consistent, reproducible results
Limitations	Requires deep domain expertise	Sensitive to hyperparameters and dataset selection

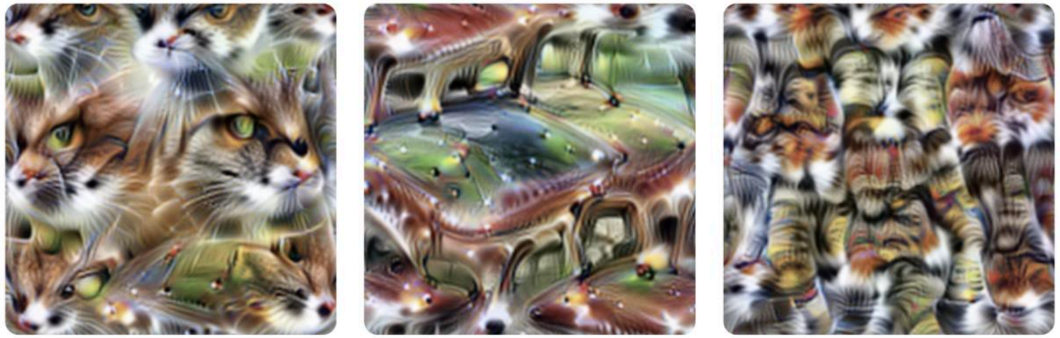
ACDC: whilst not robust great step towards automation

ARE INDIVIDUAL NEURONS MONOSEMANTIC?

Monosemanticity say individual neurons capture individual concepts

ARE INDIVIDUAL NEURONS MONOSEMANTIC?

Monosemanticity means individual neurons capture **individual concepts**



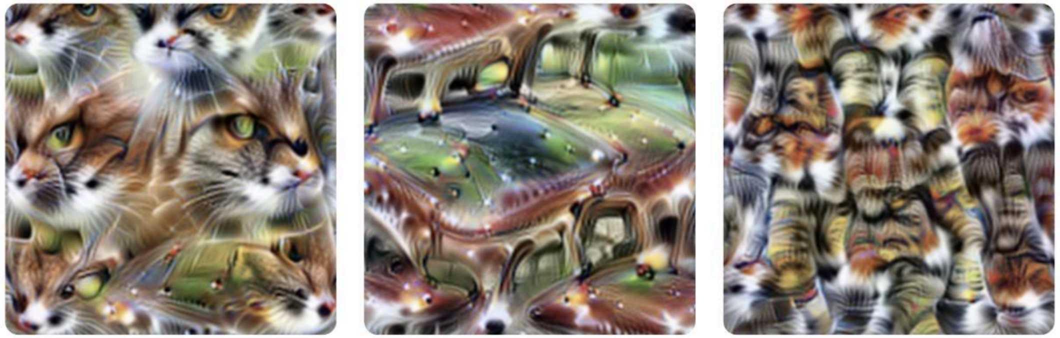
4e:55 is a polysemantic neuron which responds to cat faces, fronts of cars, and cat legs. It was discussed in more depth in Feature Visualization [4] .

This **does not** seem to be the case in practice → neurons appear to be "**Polysemantic**"

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

ARE INDIVIDUAL NEURONS MONOSEMANTIC?

Monosemanticity means individual neurons capture individual concepts



4e:55 is a polysemantic neuron which responds to cat faces, fronts of cars, and cat legs. It was discussed in more depth in Feature Visualization [4] .

This does not seem to be the case in practice → neurons appear to be "Polysemantic"

Question: do you think this would happen even if we align neurons to concepts as in CBMs?

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

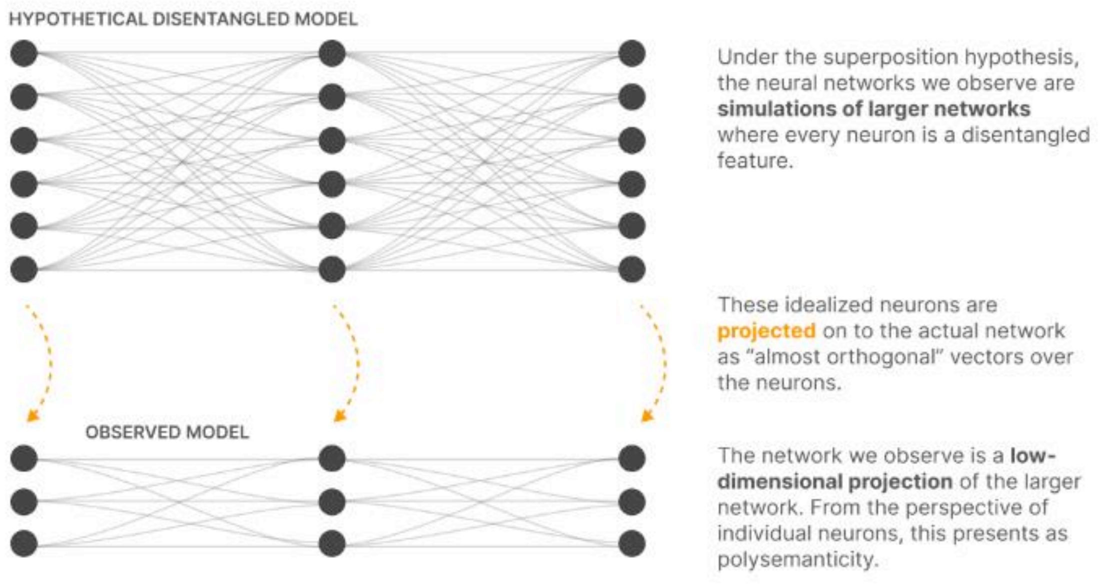
ARE INDIVIDUAL NEURONS MONOSEMANTIC?

Monosemanticity say individual neurons capture individual concepts

See below for a discussion on why cross entropy may naturally lead to"polysemantic" nodes

THE POLYSEMANTIC HYPOTHESIS

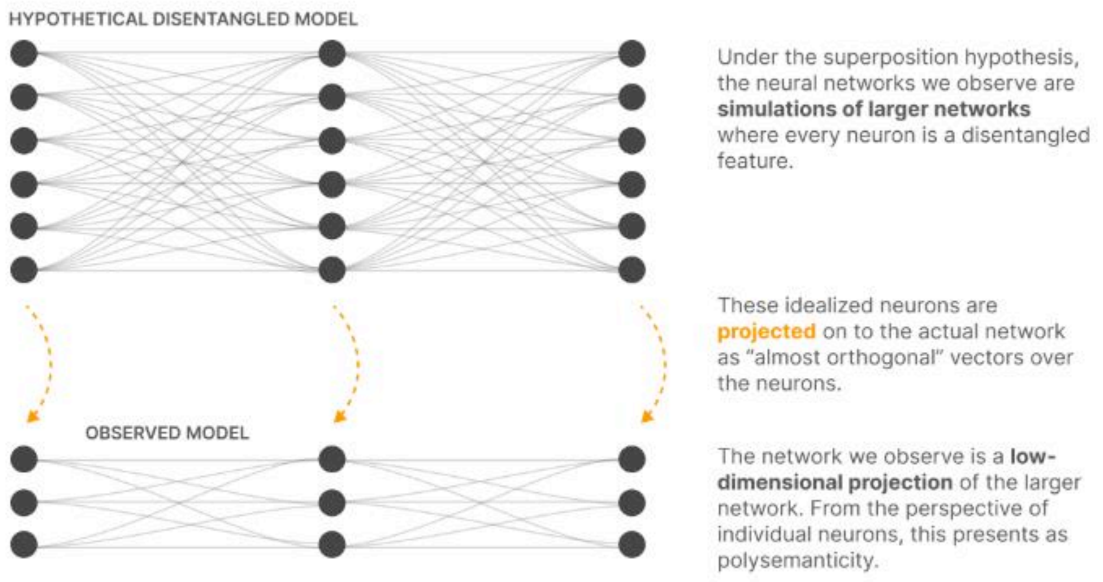
DDNs may **simulate much larger networks** by using individual neurons as **low-dimensional projections** of the hypothetical larger model



[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

THE POLYSEMANTIC HYPOTHESIS

DDNs may simulate much larger networks by using individual neurons as low-dimensional projections of the hypothetical larger model



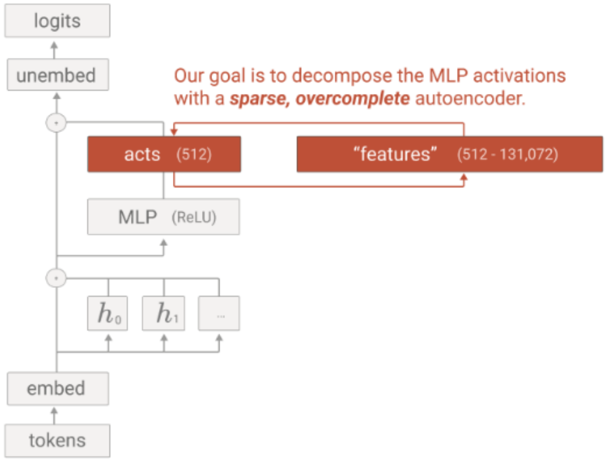
Could we then discover this "larger" neural network whose components are interpretable?

[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

FINDING THE HIGHER-LEVEL NETWORK

We want to find a representational space $\mathcal{S}$ of a model's latent activations $\mathcal{H}$ (e.g., the output of the Transformer MLP) that is:

1. Sparse: activations in $\mathcal{H}$ can be written as a combination of a handful of vectors in $\mathcal{S}$.
2. Overcomplete: dimensionality of $\mathcal{S}$ >> dimensionality of $\mathcal{H}$



[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

FINDING INTERPRETABLE LATENT DIRECTIONS

We want to decompose each embedding $\boldsymbol{x}^{(j)}$ in a Transformer's output as

$$\underbrace{\boldsymbol{x}^{(j)}}_{\text{Latent Embedding}} \approx \underbrace{\boldsymbol{b}_{\mathcal{H}}}_{\text{Bias shift}} + \sum_{\boldsymbol{i}} \Big(\underbrace{f_i\big(\boldsymbol{x}^{(j)}\big)}_{\substack{\text{Contribution of} \\ \text{i-th direction}}} \times \underbrace{\boldsymbol{s}_i}_{\substack{\text{Interpretable} \\ \text{direction in } \mathcal{S}}} \Big)$$

where we want $f_i(\boldsymbol{x})$ to be a sparse function expressing how active the $\boldsymbol{i}$-th discovered "feature"/"concept" is.

[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

FINDING INTERPRETABLE LATENT DIRECTIONS

We want to decompose each embedding $\boldsymbol{x}^{(j)}$ in a Transformer's output as

$$\underbrace{\boldsymbol{x}^{(j)}}_{\text{Latent Embedding}} \approx \underbrace{\boldsymbol{b}_{\mathcal{H}}}_{\text{Bias shift}} + \sum_{i} \Big(\underbrace{f_i\big(\boldsymbol{x}^{(j)}\big)}_{\text{Contribution of i-th direction}} \times \underbrace{\boldsymbol{s}_i}_{\text{Interpretable direction in } \mathcal{S}} \Big)$$

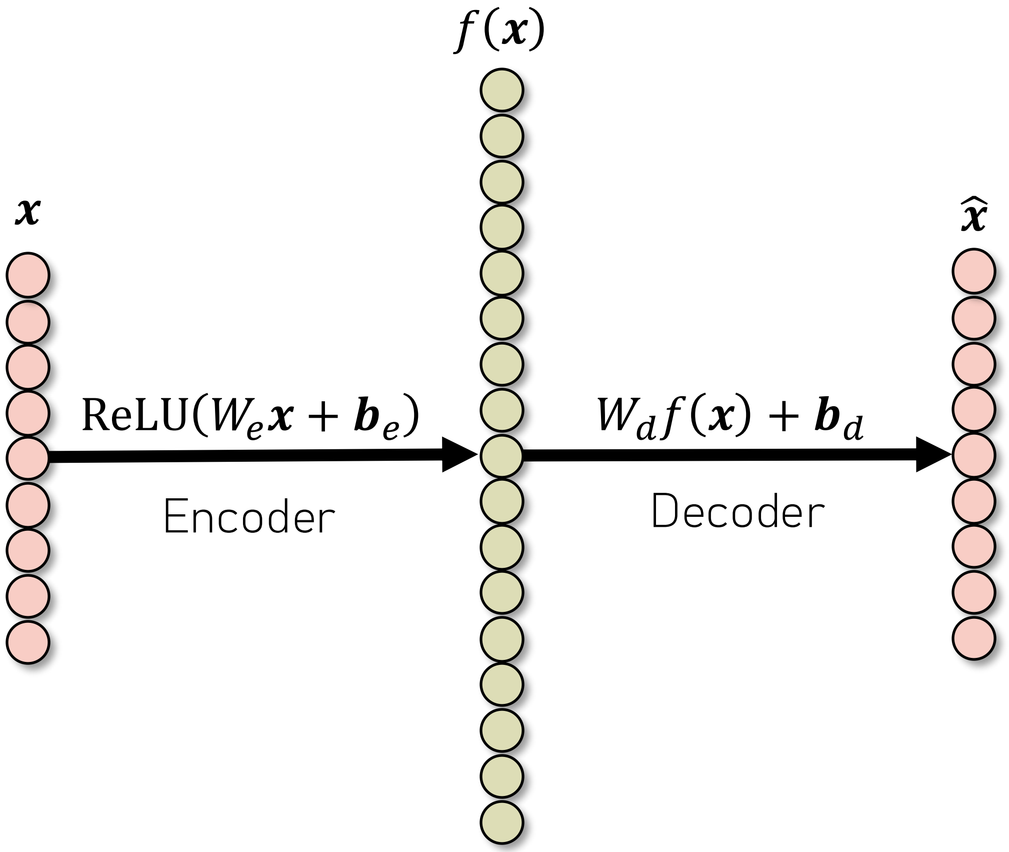
where we want $f_i(\boldsymbol{x})$ to be a sparse function expressing how active the $\boldsymbol{i}$-th discovered "feature"/"concept" is.

Question: how would you learn such a decomposition?

[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

FINDING INTERPRETABLE LATENT DIRECTIONS

One way to do this is via a simple **one-layer sparse autoencoder**!



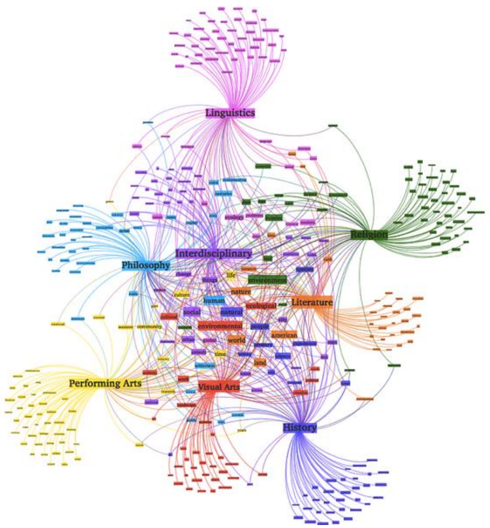
$$\mathcal{L}(x,\hat{x}) = \underbrace{||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2}_{\text{Reconstruction}} + \overbrace{\lambda|f(\boldsymbol{x})|_1}^{\text{Sparsity}}$$

*In practice, we first shift x using a learnable pre-encoder bias vector, as otherwise it is hard to learn sparse representations here

SEEING THIS AS DICTIONARY LEARNING

This is an instance of Dictionary Learning or Topic Modelling

Organise Documents by "Topics"



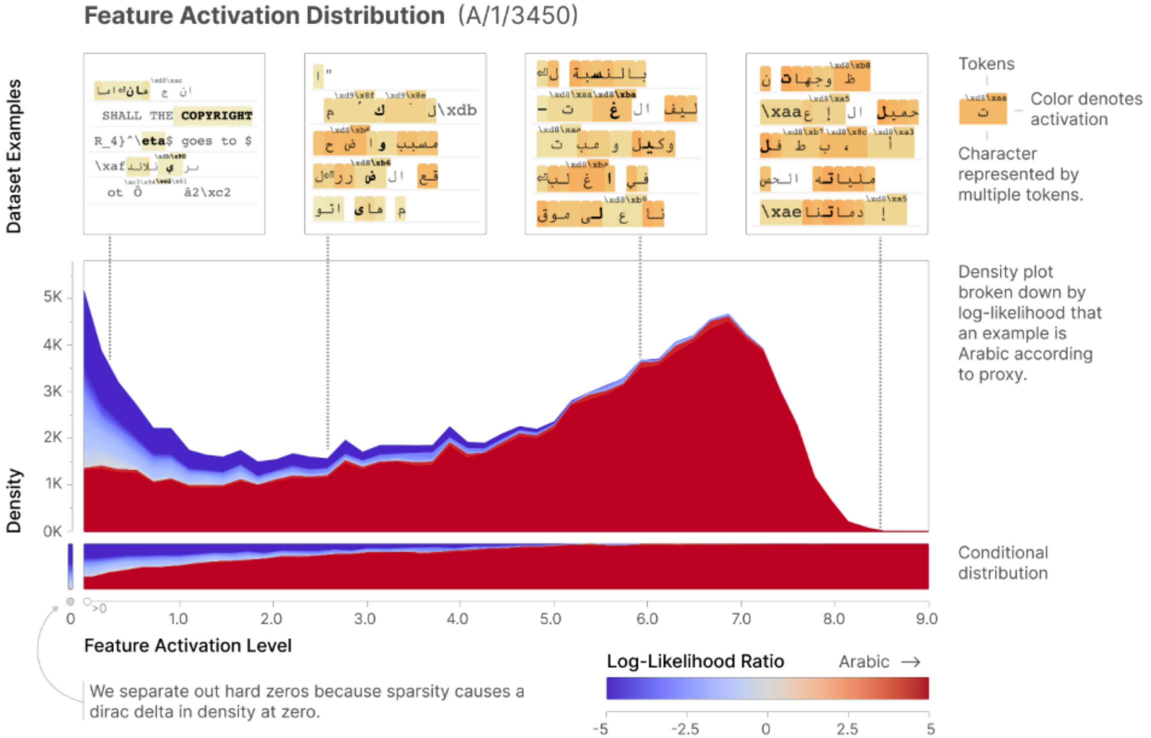
Completeness-aware Concept Extraction (CCE) can also be seen as a form of dictionary learning

[1] Image adapted from: Joyce Xu "Topic Modeling with LSA, PLSA, LDA & lda2Vec" at https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

EXPLORING A MODEL'S ACTIVATIONS

We can use this to **discover monosemantic high—dimensional features** that are **not captured by individual neurons**

This is a high-dimensional feature that almost exclusively fires when the text uses the Arabic Script



Proxy Measure

$$\frac{P(s \mid \text{Arabic Script})}{P(s)}$$

[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

POTENTIAL APPLICATIONS

By decomposing a transformer's output into interpretable concepts we can:

1. Determine a concept's contribution to the model's output or the next layer
2. Monitor the network to see if a specific concept is activated when we want to introduce safety guards.
3. Change the network's behaviour in predictable ways via interventions.
4. Demonstrate that a network learnt or used a specific property that is important for a task.

[1] Bricken et al. "Towards Monosemanticity" at https://transformer-circuits.pub/2023/monosemantic-features/index.html#phenomenology-fsa

LIMITATIONS OF MI

Mechanistic interpretability is a young field, and as such it **has a lot of known open challenges**:

LIMITATIONS OF MI

Mechanistic interpretability is a young field, and as such it has a lot of known open challenges:

1. Scalability of MI analyses is currently limited to small-ish models

LIMITATIONS OF MI

Mechanistic interpretability is a young field, and as such it has a lot of known open challenges:

1. Scalability of MI analyses is currently limited to small-ish models
2. Understanding how training dynamics affect circuits/concepts/etc

LIMITATIONS OF MI

Mechanistic interpretability is a young field, and as such it has a lot of known open challenges:

1. Scalability of MI analyses is currently limited to small-ish models
2. Understanding how training dynamics affect circuits/concepts/etc
3. Exploring unexpected reasoning phenomena as in in-context learning

LIMITATIONS OF MI

Mechanistic interpretability is a young field, and as such it has a lot of known open challenges:

1. Scalability of MI analyses is currently limited to small-ish models
2. Understanding how training dynamics affect circuits/concepts/etc
3. Exploring unexpected reasoning phenomena as in in-context learning
4. Many more!

200 Concrete Open Problems in Mechanistic Interpretability: Introduction – Neel Nanda (2022)

(A bit outdated but still useful/interesting)

FURTHER MATERIAL

MechInt has developed mostly via "grassroots"/ad-hoc efforts which means it is an area you can quickly get involved in!

Distill Circuits Thread

Anthropic Circuits Thread

[1] Cammarata, Nick, et al. "Thread: circuits." Distill 5.3 (2020): e24.
[2] Anthropic "Transformer Circuits Thread" found at https://transformer-circuits.pub/

QUESTIONS?