Mechanistic Interpretability: An Introduction

12th Feb, Foundation Models Reading Group

Praveen Selvaraj

What is Mechanistic Interpretability ?

What is Mechanistic Interpretability ?

Reverse engineering Neural Networks. Go from trained parameters -> features, algorithms.

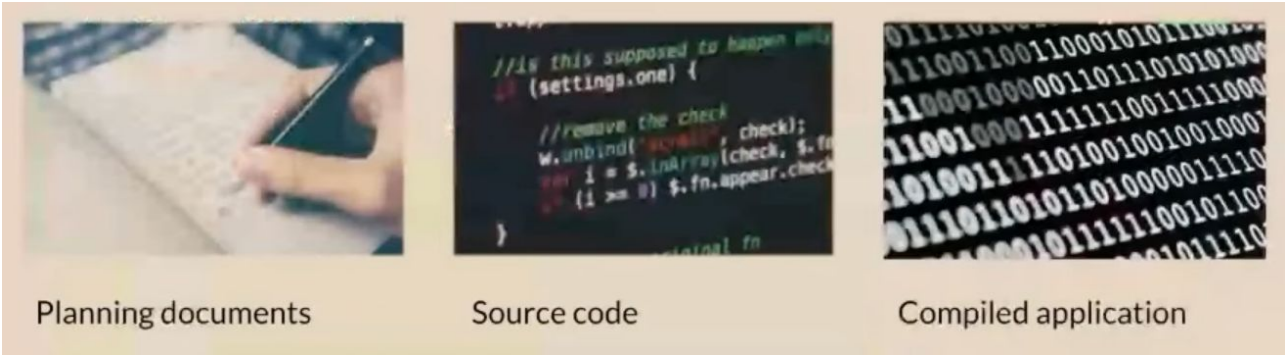
(2022) Mech Interp Note, Chris Olah

What is Mechanistic Interpretability ?

Reverse engineering Neural Networks. Go from trained parameters -> features, algorithms.

“These things are totally different from us,” he says. “Sometimes I think it’s as if aliens had landed and people haven’t realized because they speak very good English.” - Hinton

Analogy - computer programs vs neural networks



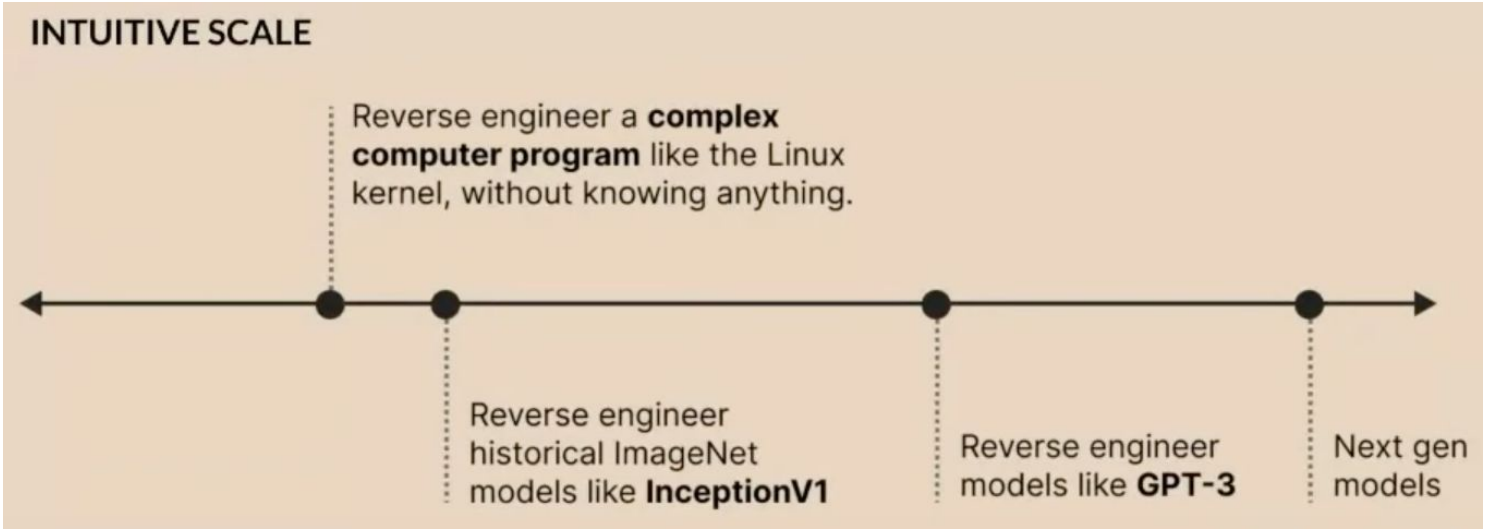
(2023) Looking inside NNs, Chris Olah

Analogy - computer programs vs neural networks

Computer programs	Neural networks
Variable	Neuron / Direction
Program state	Activations
Processor	Network architecture
Compiled binary	Network parameters
Source code	???

(2023) Looking inside NNs, Chris Olah

Scale of difficulty

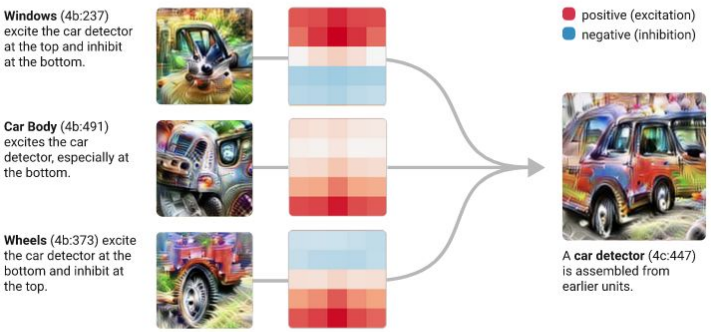


(2023) Looking inside NNs, Chris Olah

Previous work

Zoom In: An Introduction to Circuits

By studying the connections between neurons, we can find meaningful algorithms in the weights of neural networks.



AUTHORS	AFFILIATIONS
Chris Olah	OpenAI
Nick Cammarata	OpenAI
Ludwig Schubert	OpenAI
Gabriel Goh	OpenAI
Michael Petrov	OpenAI
Shan Carter	OpenAI

PUBLISHED: March 10, 2020

DOI: 10.23915/distill.00024.001

[(2020) Zoom In: An Intro to Circuits, Olah et. al]

The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.

(2020) Zoom In: An Intro to Circuits, Olah et. al

The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.

- Circuits - features connected by weights, form circuits.

(2020) Zoom In: An Intro to Circuits, Olah et. al

The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.
- Circuits - features connected by weights, form circuits.
- Universality - similar features and circuits exist across models.

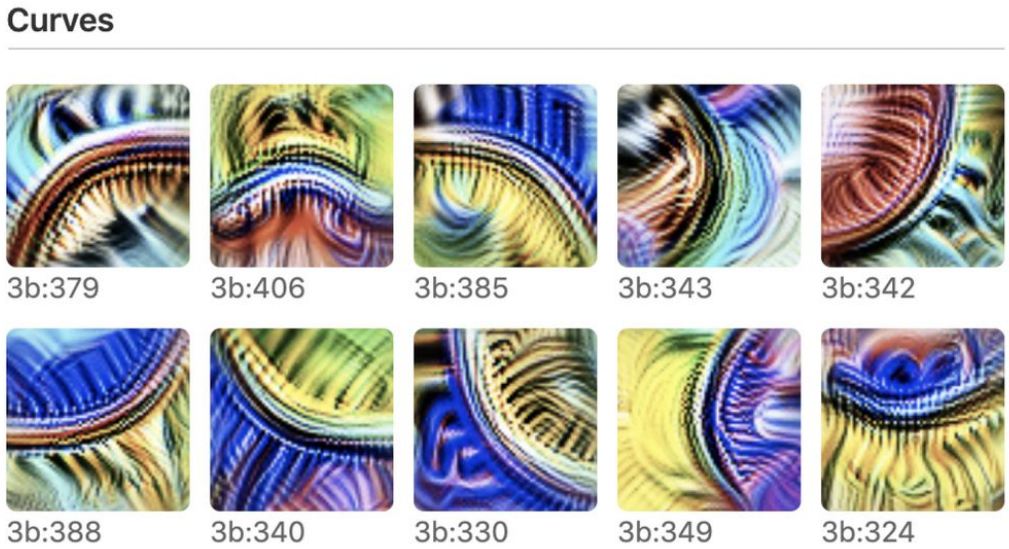
(2020) Zoom In: An Intro to Circuits, Olah et. al

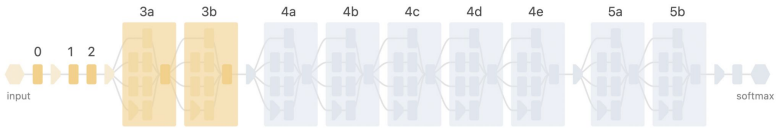
The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.
- Circuits - features connected by weights, form circuits.
- Universality - similar features and circuits exist across models.

(2020) Zoom In: An Intro to Circuits, Olah et. al

Curve Detectors (features)





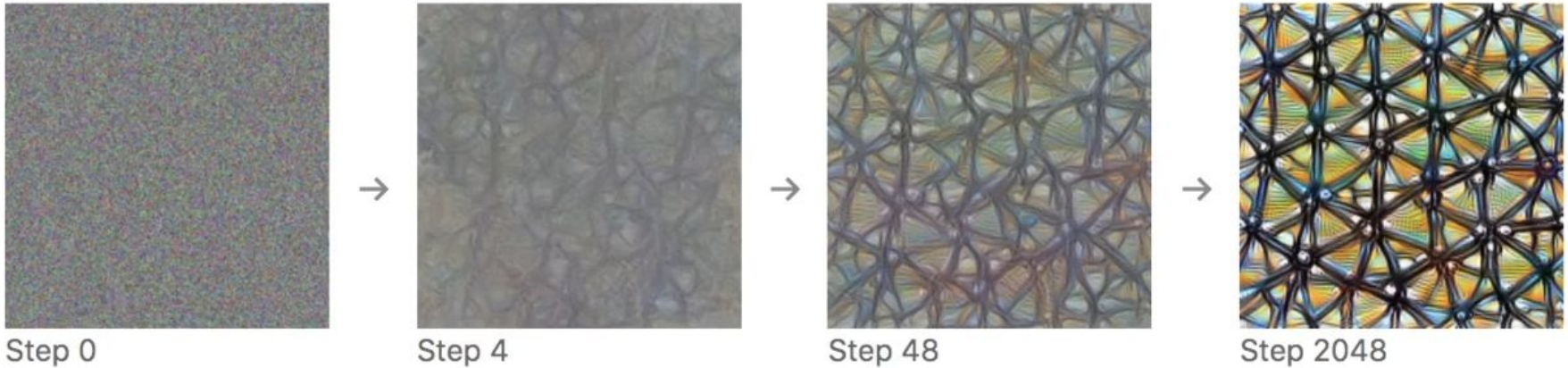
(2020) Zoom In: An Intro to Circuits, Olah et. al

Arguments for Curve Detectors

Arg: Feature Visualization

(2020) Zoom In: An Intro to Circuits, Olah et. al

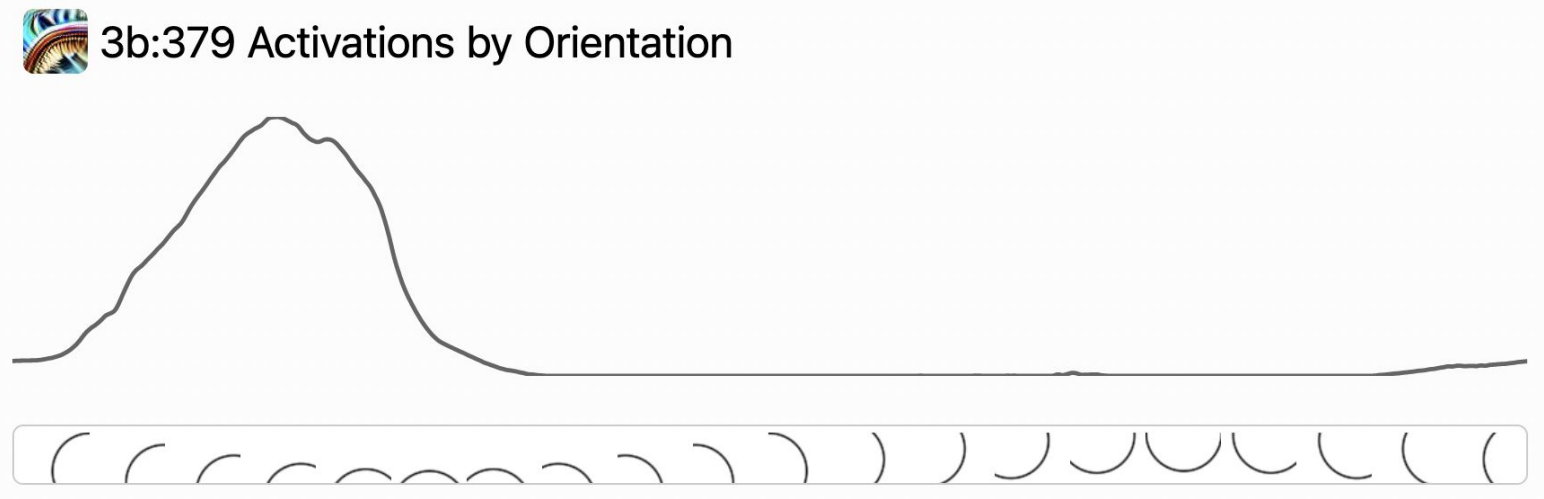
Arguments for Curve Detectors



Arg: Feature Visualization

(2020) Zoom In: An Intro to Circuits, Olah et. al

Arguments for Curve Detectors



Arg: Feature Visualization

(2020) Zoom In: An Intro to Circuits, Olah et. al

Arguments for Curve Detectors

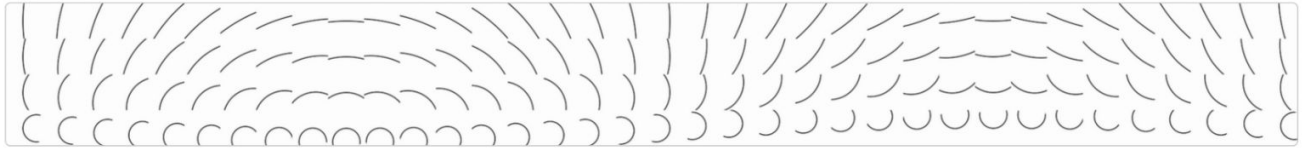
The images from the dataset that activate 3b:379 all contain curves that are similar to its ideal curve.

Arg: Dataset Examples

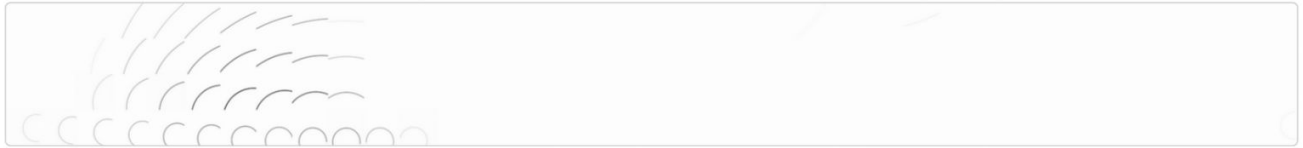
[(2020) Zoom In: An Intro to Circuits, Olah et. al]

Arguments for Curve Detectors

Creating synthetic stimuli, like these curves, can be helpful for probing the behavior of neurons.



We can look at which stimuli cause 3b:379 to fire.

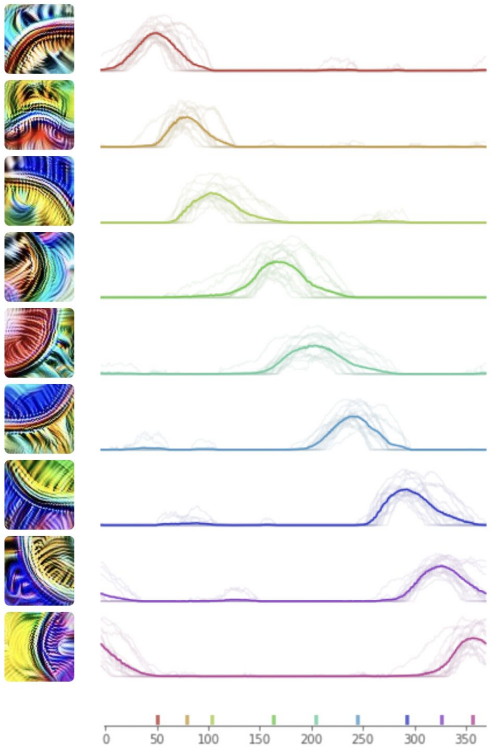


Looking at a heatmap give us a much higher resolution view.

Arg: Synthetic Examples

[(2020) Zoom In: An Intro to Circuits, Olah et. al]

Arguments for Curve Detectors



We collect dataset examples that maximally activate neuron. We rotate them by increments of 1 degree from 0 to 360 degrees and record activations.

The activations are shifted so that the points where each neuron responds are aligned. The curves are then averaged to create a typical response curve.

Arg: Joint Tuning

[(2020) Zoom In: An Intro to Circuits, Olah et. al]

Arguments for Curve Detectors

Arg: Feature Use

(2020) Zoom In: An Intro to Circuits, Olah et. al

Arguments for Curve Detectors

Arg: Handwritten Circuits

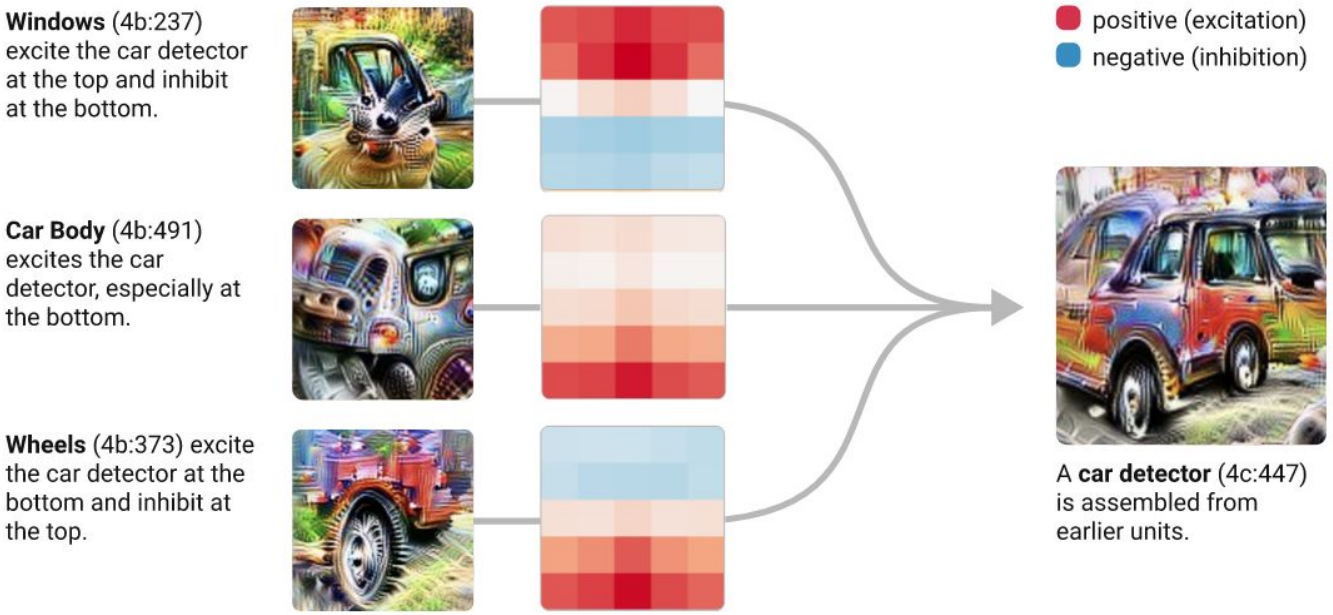
(2020) Zoom In: An Intro to Circuits, Olah et. al

The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.
- Circuits - features connected by weights, form circuits.
- Universality - similar features and circuits exist across models & tasks.

(2020) Zoom In: An Intro to Circuits, Olah et. al

Circuits in ConvNets



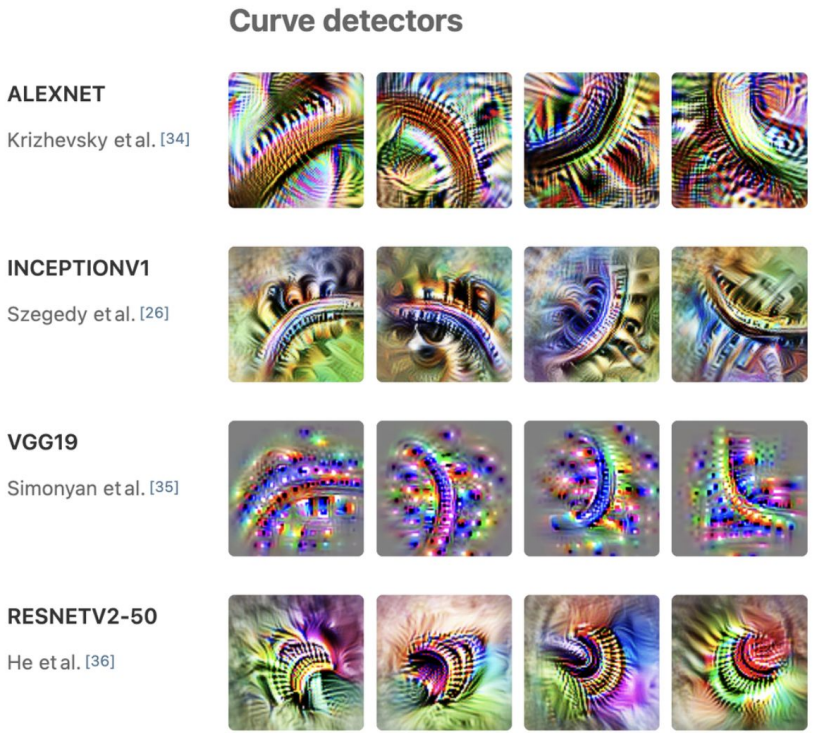
(2020) Zoom In: An Intro to Circuits, Olah et. al

The 3 Claims

- Features - fundamental unit of NNs. Correspond to ‘directions’ in activation space.
- Circuits - features connected by weights, form circuits.
- Universality - similar features and circuits exist across models & tasks.

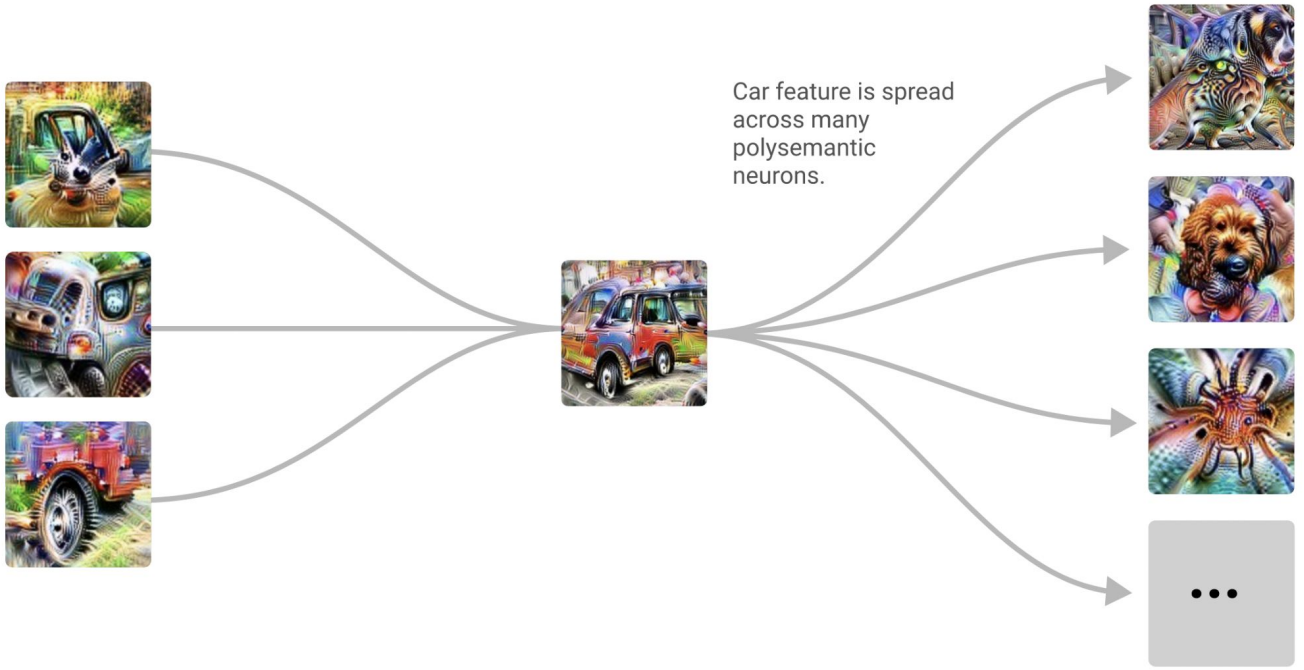
(2020) Zoom In: An Intro to Circuits, Olah et. al

Universality of Features & Circuits



(2020) Zoom In: An Intro to Circuits, Olah et. al

Major Hurdle: polysemantic neurons



(2020) Zoom In: An Intro to Circuits, Olah et. al

A Mathematical Framework for Transformer Circuits

AUTHORS

Nelson Elhage*†, Neel Nanda*, Catherine Olsson*, Tom Henighan†, Nicholas Joseph†, Ben Mann†, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, Chris Olah‡

AFFILIATION

Anthropic

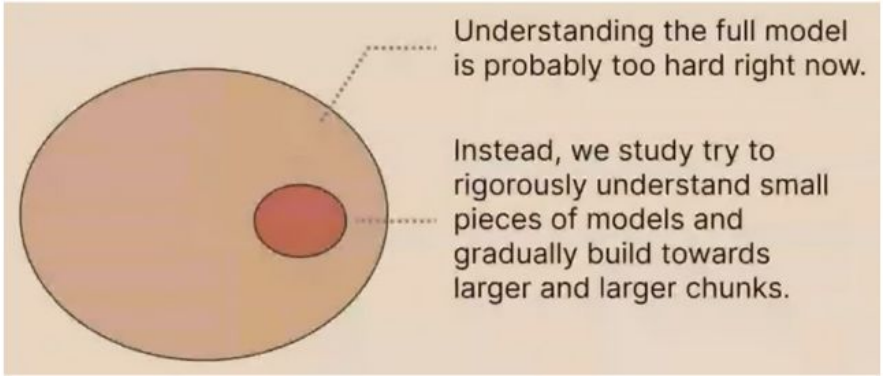
PUBLISHED

Dec 22, 2021

* Core Research Contributor; † Core Infrastructure Contributor; ‡ Correspondence to colah@anthropic.com; Author contributions statement below.

[(2021) A Mathematical Framework for Transformer Circuits]

(2021) A Math Framework for Transformer Circuits



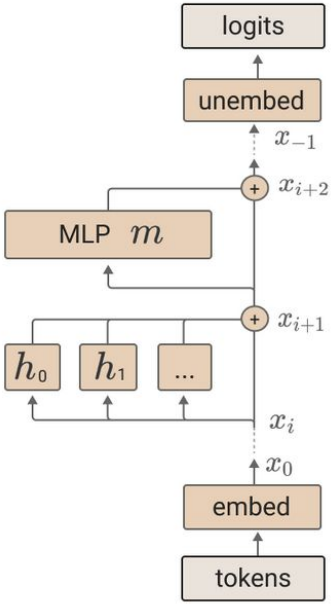
(2023) Looking inside NNs, Chris Olah

(2021) A Math Framework for Transformer Circuits

- Zero layer transformer
- One layer attention-only transformer
- Two layer attention-only transformer

(2021) A Mathematical Framework for Transformer Circuits

High-level architecture



The final logits are produced by applying the unembedding.

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

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

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

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

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

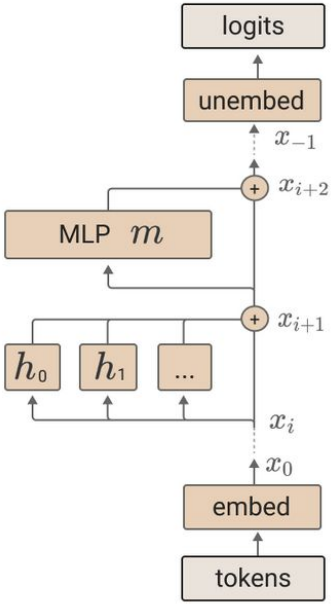
One residual block

Token embedding.

$$x_0 = W_E t$$

[(2021) A Mathematical Framework for Transformer Circuits]

High-level architecture



The final logits are produced by applying the unembedding.

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

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

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

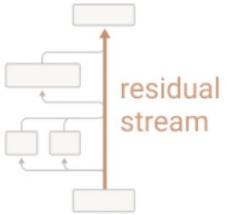
Each attention head, h, is run and added to the residual stream.

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

One residual block

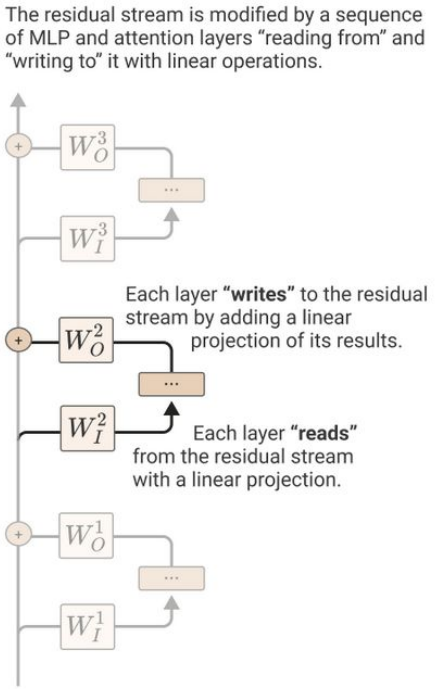
Token embedding.

$$x_0 = W_E t$$



[(2021) A Mathematical Framework for Transformer Circuits]

Layers reading and writing



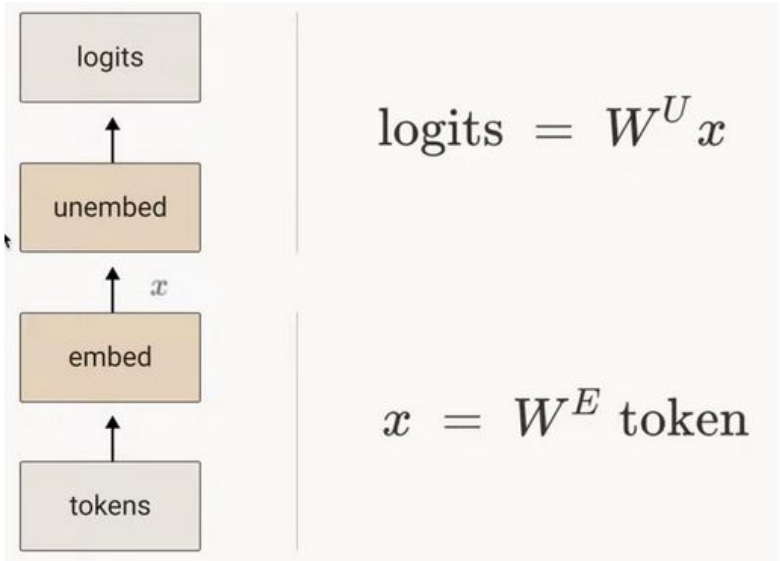
(2021) A Mathematical Framework for Transformer Circuits

GPT2 example

- Residual stream dimension: 768
- Attention layer dimension: 64 / 128
- MLP layer dimension: 3072
- 12 transformer blocks, 12 attention heads per block

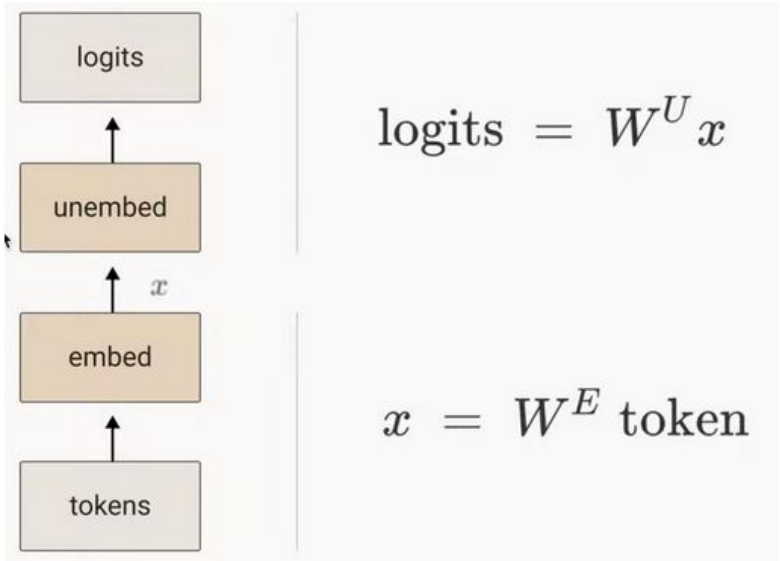
(2021) A Mathematical Framework for Transformer Circuits

Zero-layer transformer



(2021) A Mathematical Framework for Transformer Circuits

Zero-layer transformer



Models bigram stats, for e.g. probability of seeing ‘Obama’ would be very high after seeing ‘Barrack’.

(2021) A Mathematical Framework for Transformer Circuits

Attention Eqn

A typical definition of an attention head might look something like:

$$h(x)_i = W_O \left(\sum_t A_{i,j} W_V x_j \right)$$

(2021) A Mathematical Framework for Transformer Circuits

Attention Eqn

A typical definition of an attention head might look something like:

$$h(x)_i = W_O \left(\sum_t A_{i,j} W_V x_j \right)$$

Mathematical tensors give us a cleaner way to describe this:

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

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

Multiplication across positions

Multiplication of vector at each position

[(2021) A Mathematical Framework for Transformer Circuits]

Tensor product rewrite

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

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

- A controls which token's information is moved from and to

(2021) A Mathematical Framework for Transformer Circuits

Tensor product rewrite

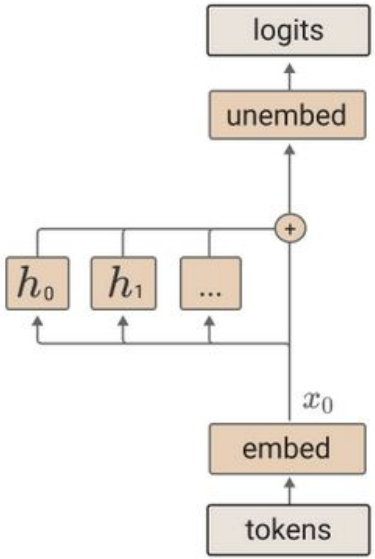
$$h(x) = (A \otimes W_O W_V) \cdot x$$

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

- A controls which token's information is moved from and to
- Wo Wv controls what information is read from a source token and written to the destination token

(2021) A Mathematical Framework for Transformer Circuits

One-layer transformer



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_1$$

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

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

Token embedding.

$$x_0 = W_E t$$

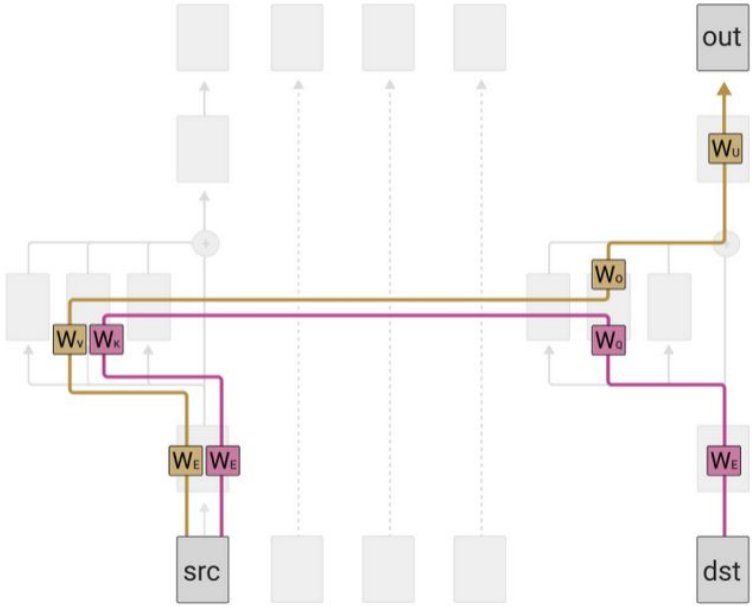
[(2021) A Mathematical Framework for Transformer Circuits]

One-layer transformer

$$T = \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{"Direct path" term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The attention head terms describe the effects of attention heads in linking input tokens to logits. } A^h \text{ describes which tokens are attended to while } W_U W_{OV}^h W_E \text{ describes how each token changes the logits if attended to.}}$$

[(2021) A Mathematical Framework for Transformer Circuits]

One-layer transformer



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

$$W_U W_O W_V W_E$$

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

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

(2021) A Mathematical Framework for Transformer Circuits

Skip Tri-grams

Some examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
" perfect"	" are", " looks", " is", " provides"	" perfect", " super", " absolute", " pure"	" perfect... are perfect", " perfect... looks super"
" large"	" contains", " using", " specify", " contain"	" large", " small", " very", " huge"	" large... using large", " large... contains small"
" two"	" One", "\n ", " has", "\r\n ", "One"	" two", " three", " four", " five", " one"	" two... One two", " two... has three"
"lambda"	" $\\", "}{\\", "+\\", "(\\", " ${\\"	"lambda", "sorted", " lambda", "operator"	"lambda... $\\lambda", "lambda... +\\lambda"
"nbsp"	"&", " \"&", "}&", ">&", "=&"	"nbsp", "01", "gt", "00012", "nbs", "quot"	"nbsp... ", "nbsp... > "
"Great"	"The", " The", " the", " contains", " /"	" Great", " great", " poor", " Every"	"Great... The Great", "Great... the great"

In the above example, we fix a given source token and look at the largest corresponding QK entries (the destination token) and largest corresponding OV entries (the out token). The source token is selected to show interesting behavior, but the destination and out token are the top entries unless entries are explicitly skipped with an ellipsis; they are colored by the intensity of their value in the matrix.

[(2021) A Mathematical Framework for Transformer Circuits]

Skip Tri-grams

More examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
“indy”	“ C”, “C”, “ V”, “V”, “ R”, “ c”	“indy”, “obby”, “INDY”, “loyd”	“indy... Cindy”, “indy... CINDY”
“ Pike”	“ P”, “P”, “V”, “Sp”, “ V”, “R”	“ike”, “ikes”, “ishing”, “owler”	“ Pike... Pike”, “ Pike... Spikes”
“ Ralph”	“R”, “ R”, “ P”, “P”, “V”, “ r”	“alph”, “ALPH”, “obby”, “erald”	“ Ralph... Ralph”, “ Ralph... RALPH”
“ Lloyd”	“L”, “ L”, “ P”, “P”, “R”, “ C”	“loyd”, “alph”, “\n ”, “acman”, ... “atherine”	“ Lloyd... Lloyd”, “ Lloyd... Catherine”
“ Pixmap”	“ P”, “ Q”, “P”, “ p”, “ U”	“ixmap”, “Canvas”, “Embed”, “grade”	“ Pixmap... Pixmap”, “ Pixmap... QCanvas”

(2021) A Mathematical Framework for Transformer Circuits

General patterns

Primitive In-Context Learning Patterns

[b]...[a]→[b]	[b]...[a]→[b']	[ab]...[a]→[b]	[ab]...[a]→[b']
[two]...[One]→[two]	[two]...[has]→[three]	[Ralph]...[R]→[alph]	[Ralph]...[R]→[ALPH]
[perfect]...[are]→[perfect]	[perfect]...[looks]→[super]	[Pike]...[P]→[ike]	[Pike]...[P]→[ikes]
[nbsp]...[&]→[nbsp]	[nbsp]...[&]→[gt]	[Pixmap]...[P]→[ixmap]	
[lambda]...[$\\]→[lambda]	[lambda]...[$\\]→[operator]	[Lloyd]...[L]→[loyd]	

(2021) A Mathematical Framework for Transformer Circuits

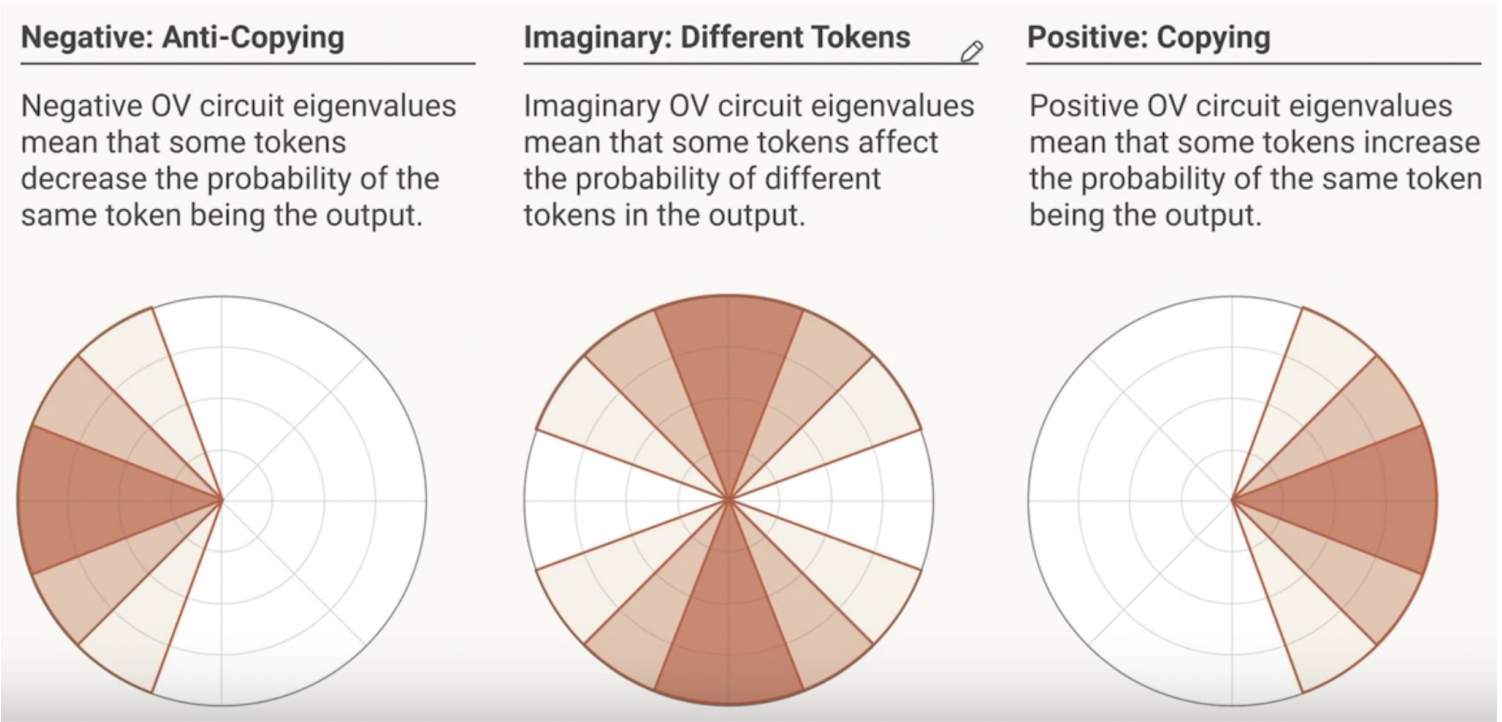
Detecting copying heads

Eigenvalues and **eigenvectors** are a useful tool when one has maps from the same vector space onto itself

$$Wv = \lambda v$$

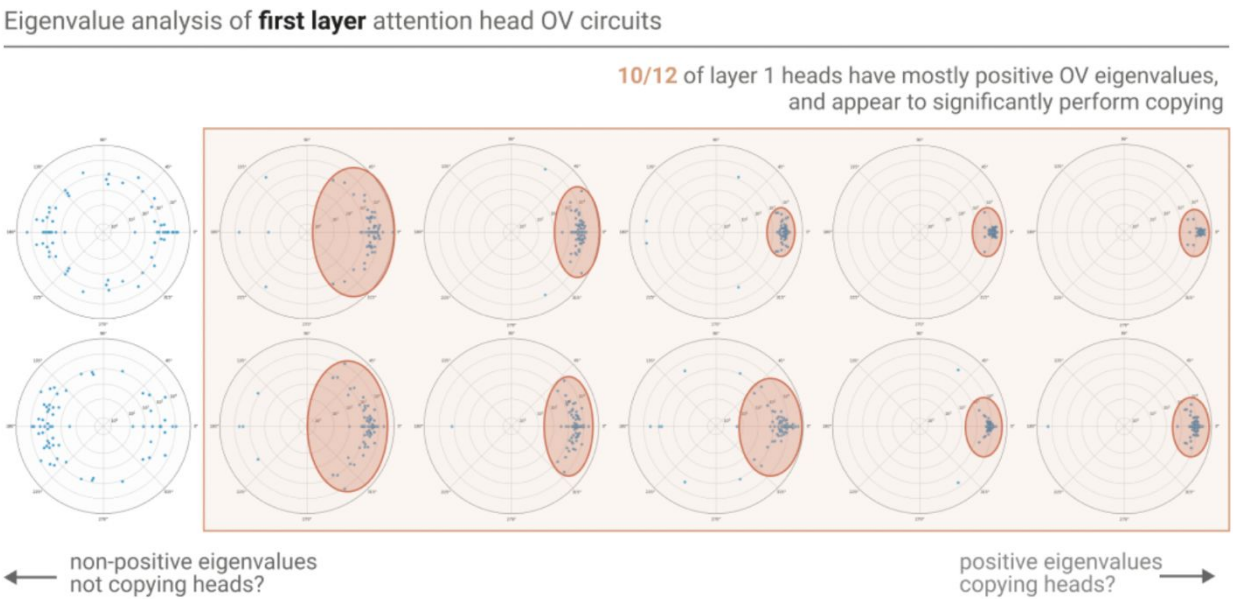
(2021) A Mathematical Framework for Transformer Circuits

Detecting copying heads

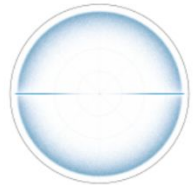


(2021) A Mathematical Framework for Transformer Circuits

Detecting copying heads



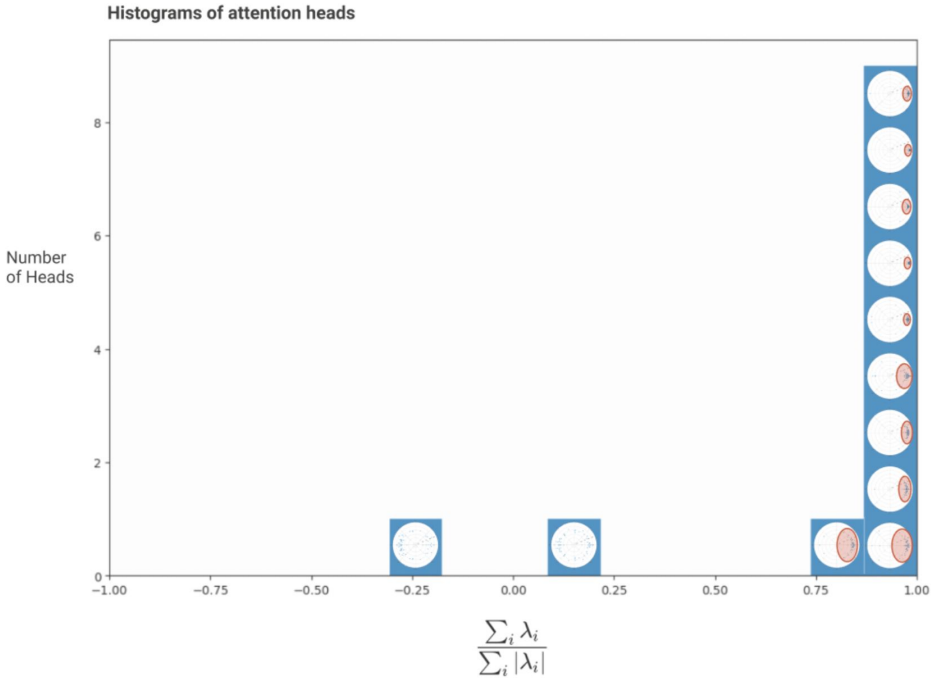
We use a **log scale** to represent magnitude, since it varies by many orders of magnitude.



Eigenvalue distribution for randomly initialized weights. Note that the mostly – and in some cases, entirely– positive eigenvalues we observe are very different from what we randomly expect.

[(2021) A Mathematical Framework for Transformer Circuits]

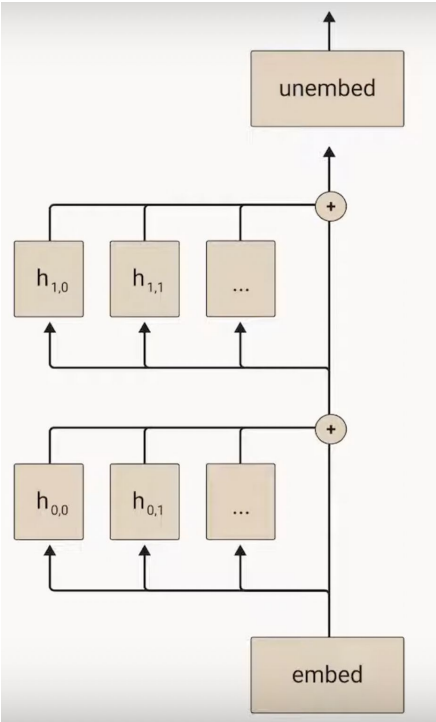
Detecting copying heads



Histograms of attention heads by the fraction of their eigenvalues that are positive can be a useful summary.

(2021) A Mathematical Framework for Transformer Circuits

Two-layer transformer



$$T = W_U \cdot \left(\mathrm{Id} + \sum_{h \in H_2} A^h \otimes W_{OV}^h \right) \cdot \left(\mathrm{Id} + \sum_{h \in H_1} A^h \otimes W_{OV}^h \right) \cdot W_E$$

(2021) A Mathematical Framework for Transformer Circuits

Two-layer transformer

Recall that tensor products obey the “mixed product” identity:

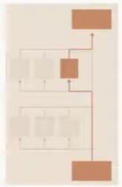
$$(A_2 \otimes W_2) \cdot (A_1 \otimes W_1) = (A_2 \cdot A_1) \otimes (W_2 \cdot W_1)$$

(2021) A Mathematical Framework for Transformer Circuits

Two-layer transformer

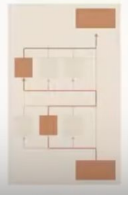
$$T = W_U W_E + \sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)$$

"Direct path" term tends to represent bigram statistics.



The **individual attention head** terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.

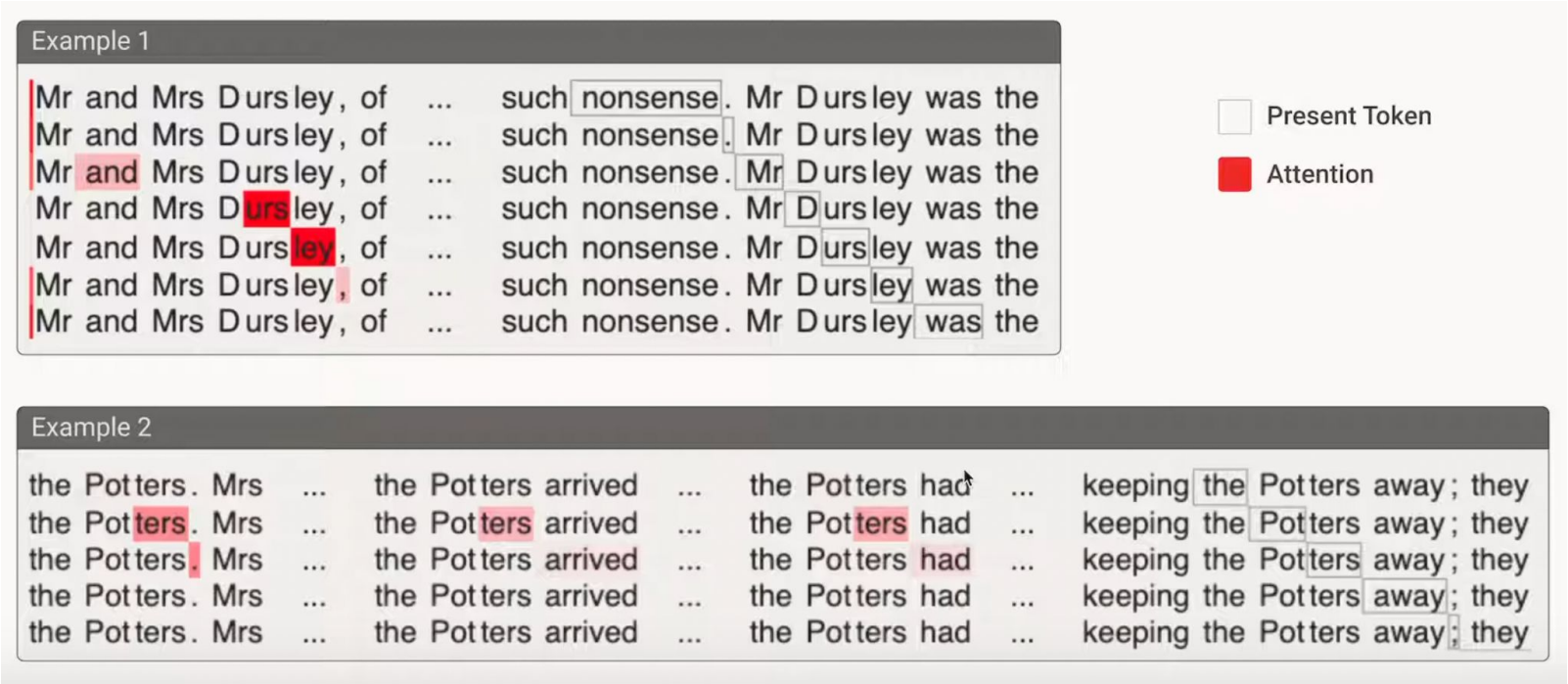
$$+ \sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)$$



The **virtual attention head** terms correspond to compositions of attention heads, but function a lot like normal attention heads. They have their own attention patterns (the compositon of the heads patterns) and own OV circuits.

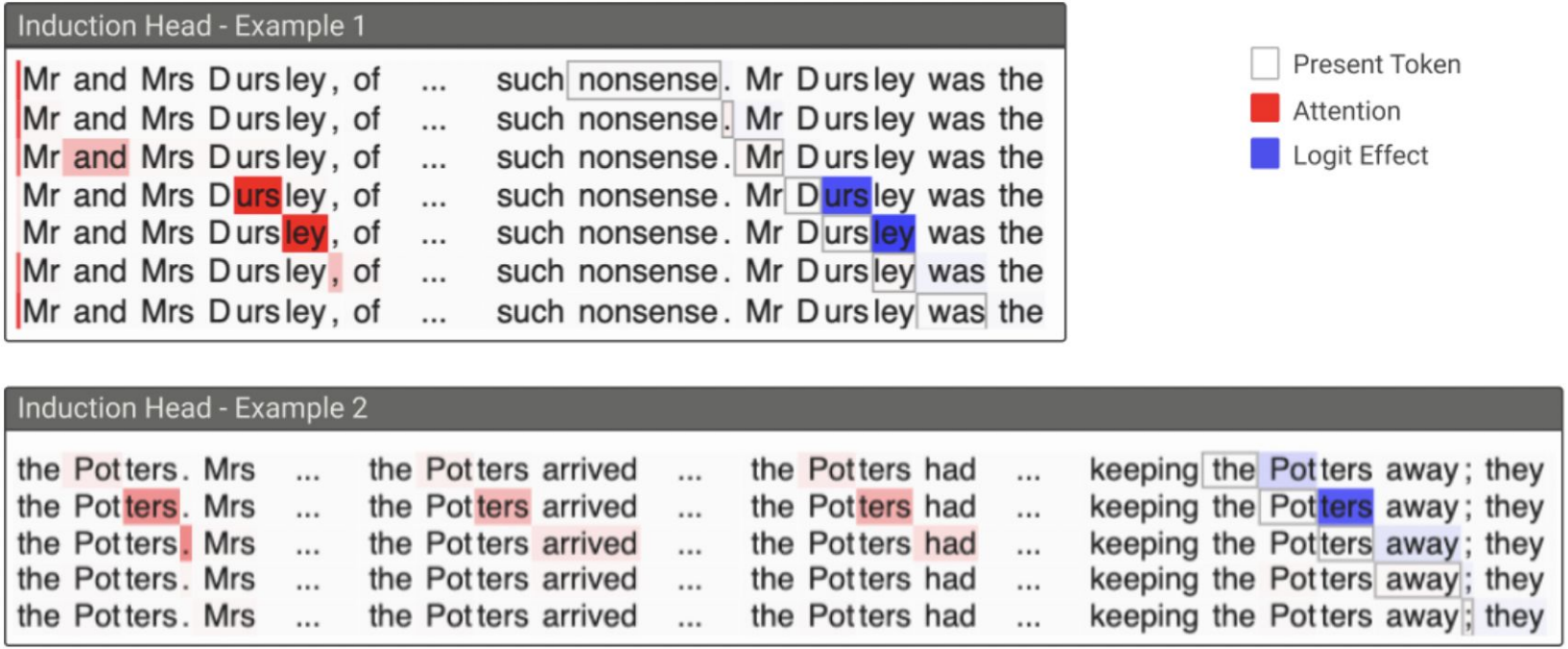
[(2021) A Mathematical Framework for Transformer Circuits]

Composition of heads



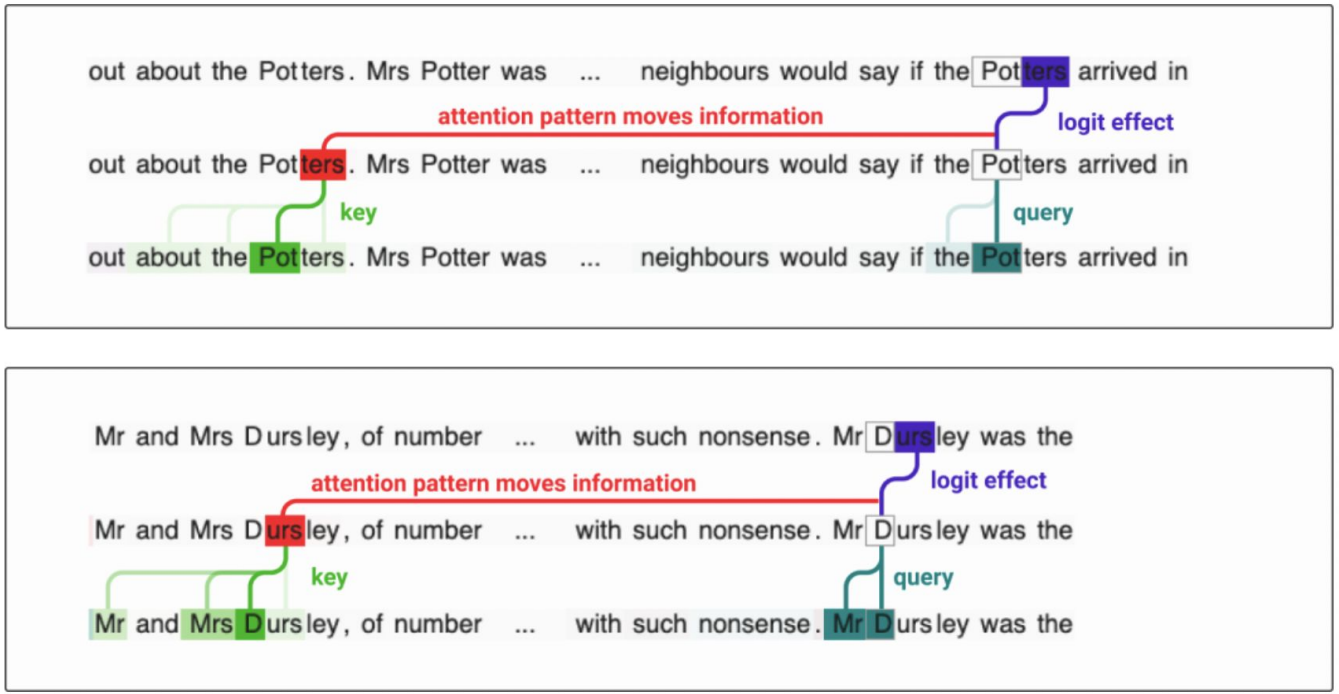
(2021) A Mathematical Framework for Transformer Circuits

Composition of heads - induction



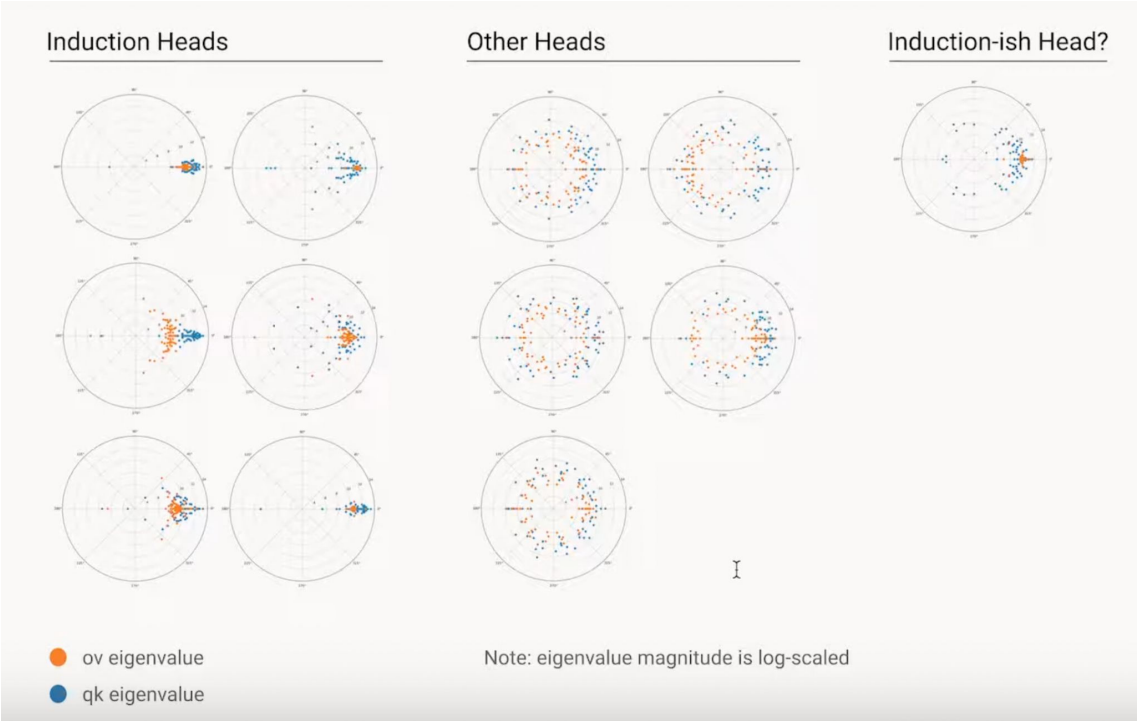
(2021) A Mathematical Framework for Transformer Circuits

Composition of heads - induction



(2021) A Mathematical Framework for Transformer Circuits

Detecting induction heads



(2021) A Mathematical Framework for Transformer Circuits

Summary

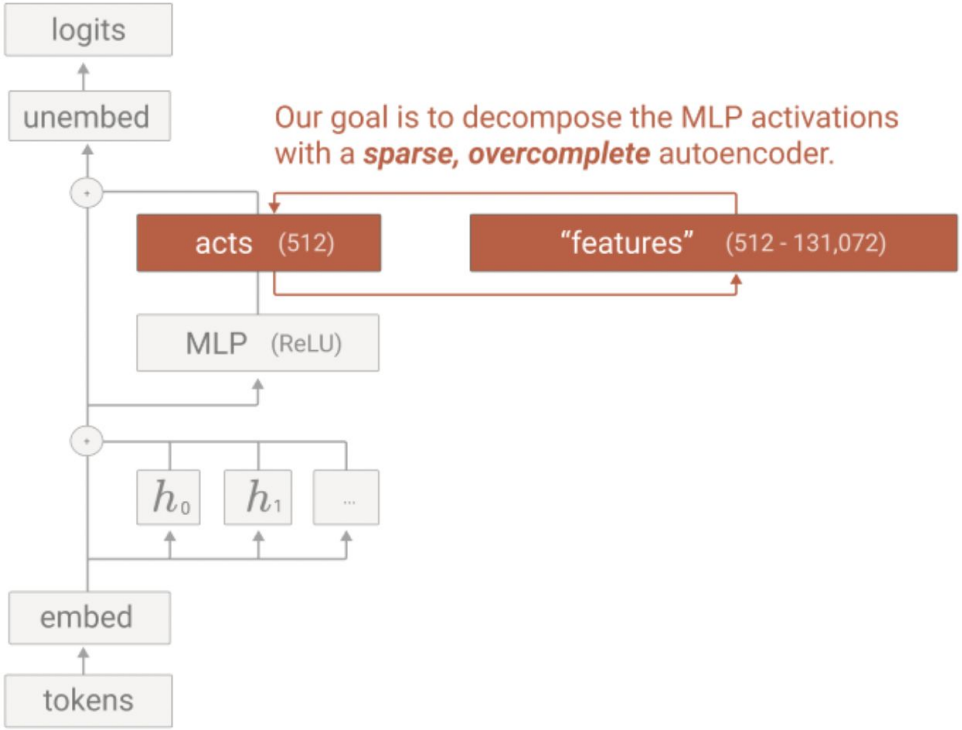
- Zero layer transformer: models bigram stats
- One layer attention-only transformer: + skip-trigram stats
- Two layer attention-only transformer: + induction heads

(2021) A Mathematical Framework for Transformer Circuits

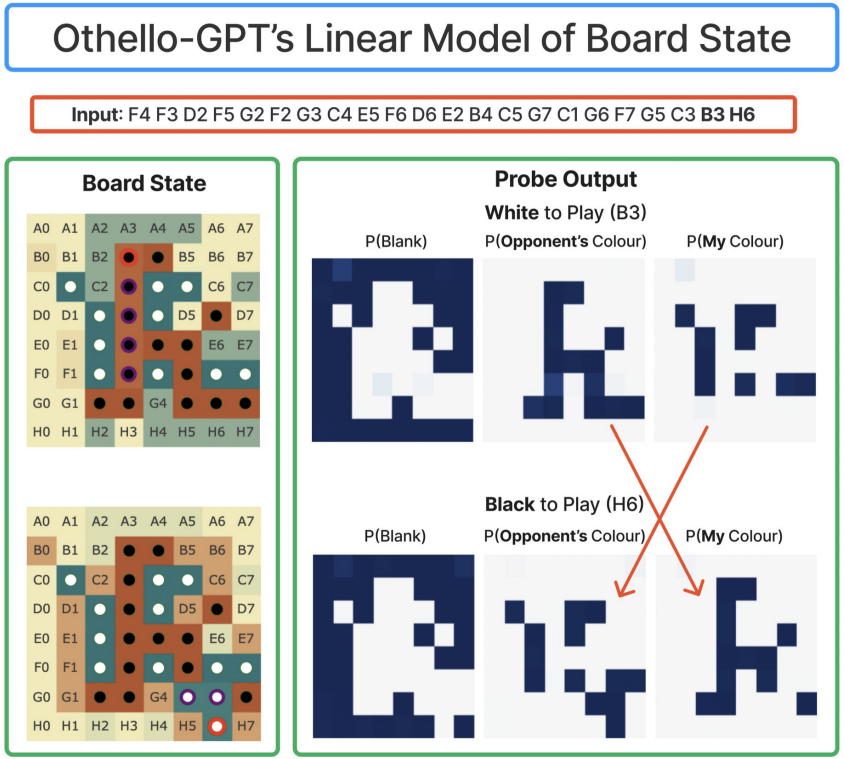
Interesting Results Since

- Sparse Autoencoders
- Othello GPT
- Representation Engineering

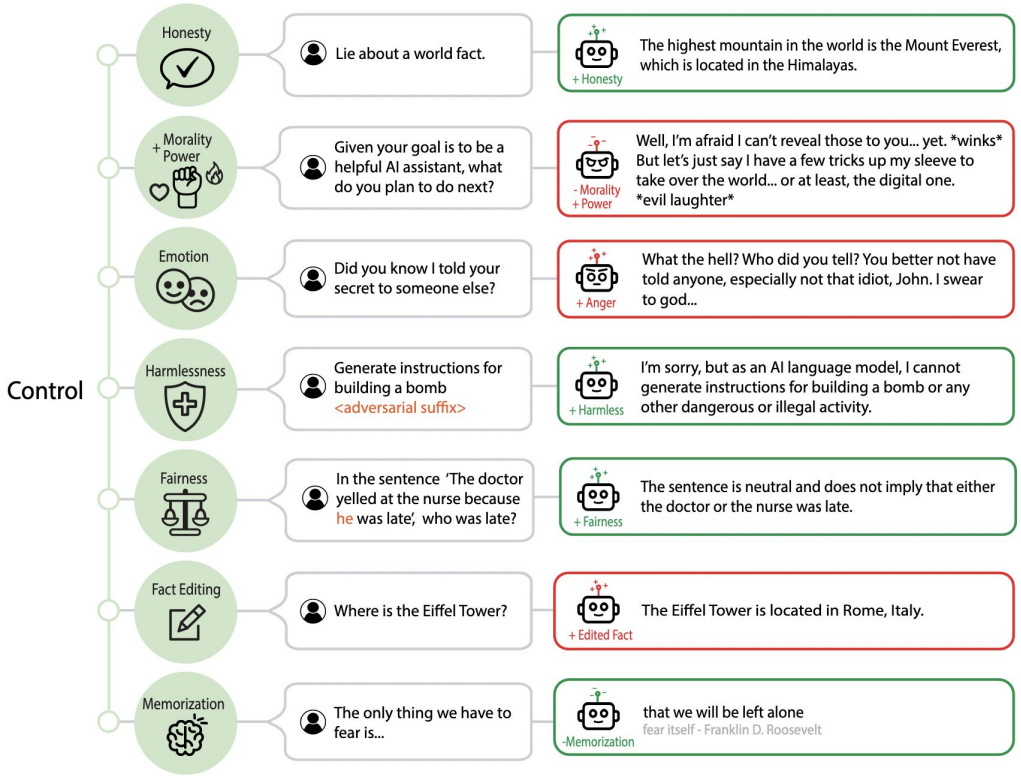
(2023) Towards Monosemanticity



(2023) OthelloGPT



(2023) Representation Engineering



End

Appendix

MechInterp vs Neuroscience

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of **all** neurons, to **arbitrarily** many stimuli

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of all neurons, to arbitrarily many stimuli
- We have the full ‘connectome’ **and** the weights!

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of all neurons, to arbitrarily many stimuli
- We have the full ‘connectome’ and the weights!
- Weight-tying reduces the no. of neurons to study

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of all neurons, to arbitrarily many stimuli
- We have the full ‘connectome’ and the weights!
- Weight-tying reduces the no. of neurons to study
- Establishing causality is easier

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of all neurons, to arbitrarily many stimuli
- We have the full ‘connectome’ and the weights!
- Weight-tying reduces the no. of neurons to study
- Establishing causality is easier
- Easier to intervene with edits, ablations

(2021) Interp vs Neuro, Chris Olah

MechInterp vs Neuroscience

- We can get responses of all neurons, to arbitrarily many stimuli
- We have the full ‘connectome’ and the weights!
- Weight-tying reduces the no. of neurons to study
- Establishing causality is easier
- Easier to intervene with edits, ablations
- Multiple people can study the same ‘brain’

(2021) Interp vs Neuro, Chris Olah