INSTITUTE FOR MATHEMATICS

STRUCTURES CLUSTER OF EXCELLENCE

UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

Michael Bleher

Institute for Mathematics, Heidelberg University

– YTM 2025 –

The Tangled Web they Weave

Exploring Neural Networks with Directed Topology

Artificial Neural Networks

Multi-Layer Perceptron (MLP)

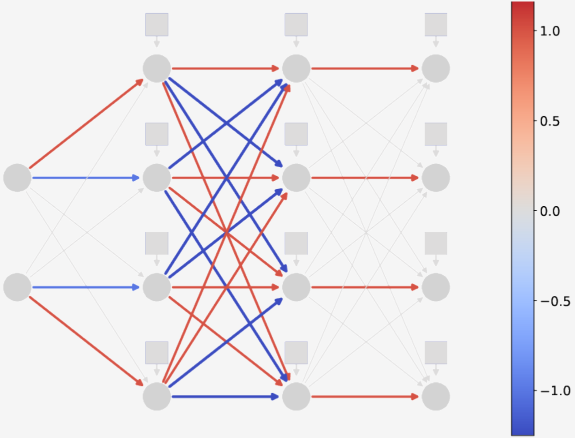
$$f : \mathbb{R}^n \to \mathbb{R}^{k_1} \to \ldots \to \mathbb{R}^{k_{L-1}} \to \mathbb{R}^m$$

$$a^{(0)} = x \in X \subset \mathbb{R}^n$$
$$a^{(\ell)} = \sigma(W^{(\ell)} a^{(\ell-1)} + b^{(\ell)}) \in \mathbb{R}^{k_\ell}$$

σ – activation function

$W^{(\ell)}$ – weight matrix

$b^{(\ell)}$ – bias



Artificial Neural Networks

Multi-Layer Perceptron (MLP)

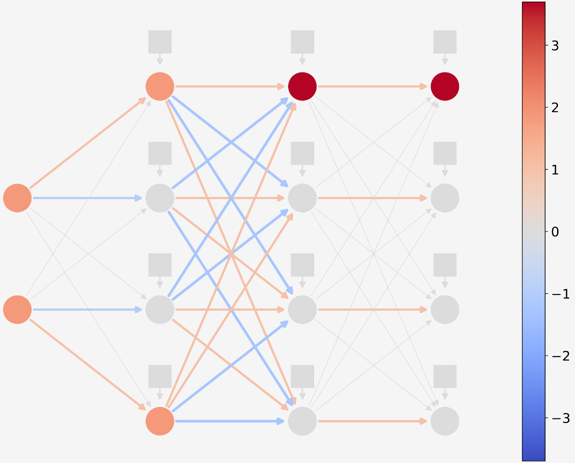
$$f : \mathbb{R}^n \to \mathbb{R}^{k_1} \to \ldots \to \mathbb{R}^{k_{L-1}} \to \mathbb{R}^m$$

$$a^{(0)} = x \in X \subset \mathbb{R}^n$$
$$a^{(\ell)} = \sigma(W^{(\ell)} a^{(\ell-1)} + b^{(\ell)}) \in \mathbb{R}^{k_\ell}$$

σ – activation function

$W^{(\ell)}$ – weight matrix

$b^{(\ell)}$ – bias



Mechanistic Interpretability

Neural networks learn to

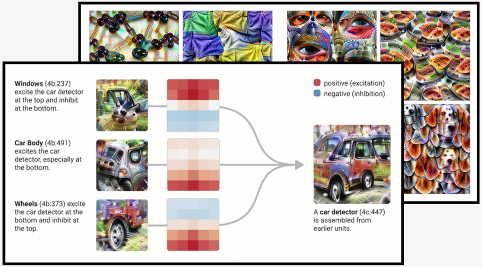
- recognize **features**

Olah et al. (2017)

Mechanistic Interpretability

Neural networks learn to

- recognize **features**
- and perform **internal computations**.



Olah et al. (2020)

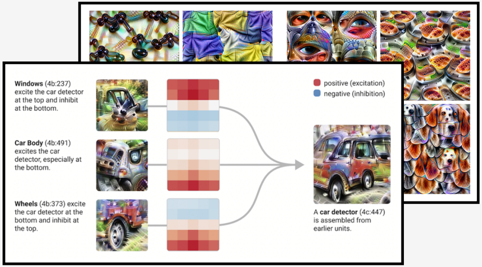
Mechanistic Interpretability

Neural networks learn to

- recognize **features**
- and perform **internal computations**.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$



Olah et al. (2020)

Mechanistic Interpretability

Neural networks learn to

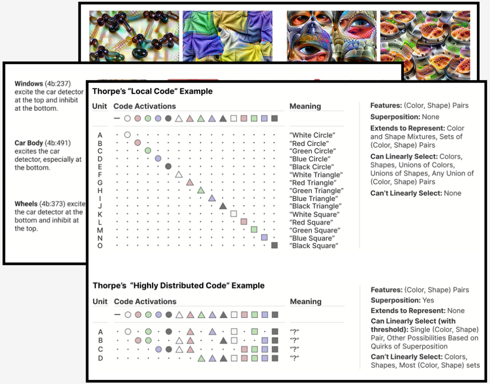
- recognize **features**
- and perform **internal computations**.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$

Feature Representations

- **Distributed**: Features span multiple neurons/layers



Olah (2023) – informal note

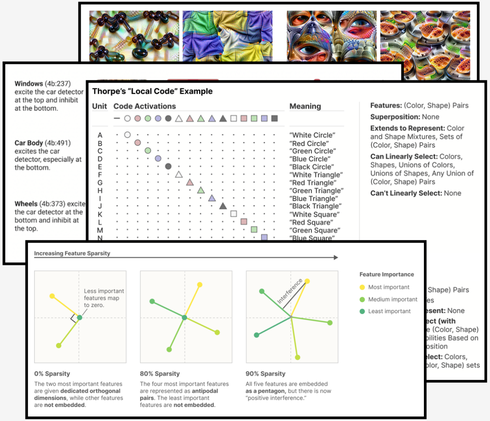
Mechanistic Interpretability

Neural networks learn to

- recognize **features**
- and perform **internal computations**.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \ldots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$



Olah et al. (2022)

Feature Representations

- **Distributed**: Features span multiple neurons/layers
- **Superposition**: Neuron activation patterns overlap

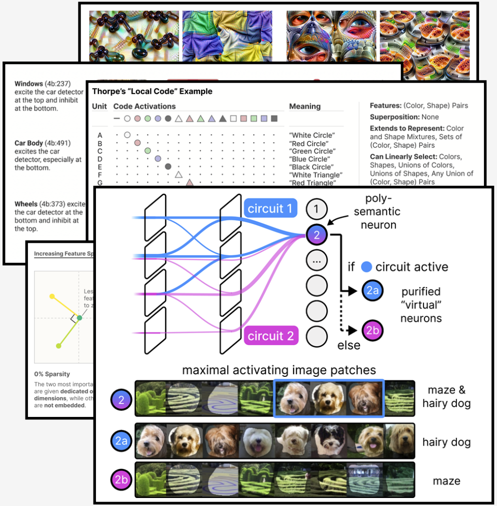
Mechanistic Interpretability

Neural networks learn to

- recognize **features**
- and perform **internal computations**.

Goal: Understand in terms of *internal neuron activations*

$$a(x) := (a^{(1)}, \dots, a^{(L)}) \in M \subset \mathbb{R}^N, N = \sum_{\ell=1}^{L-1} k_\ell$$



Dreyer et al. (2024)

Feature Representations

- **Distributed**: Features span multiple neurons/layers
- **Superposition**: Neuron activation patterns overlap
- **Polysemanticity**: Individual neurons respond to multiple unrelated concepts

Mechanistic Interpretability

Task

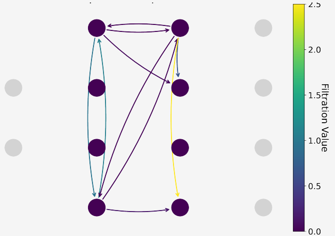
Disentangle superpositions of distributed features that activate polysemantic neurons.

Status quo: *ICA, NMF, sparse autoencoders*
→ recover *coordinate axes* of the feature space.

Why (directed) topology?

"What fires together, wires together"

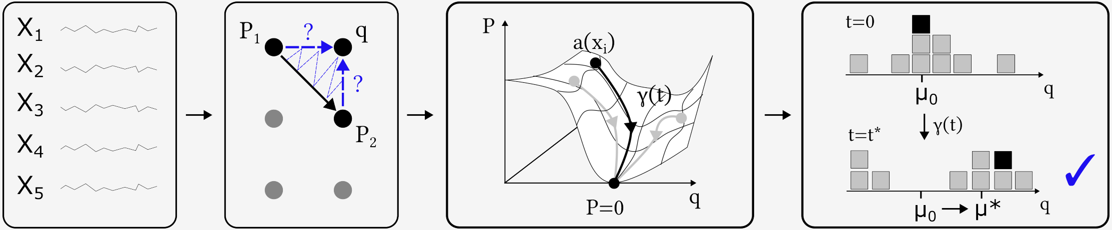
- coactivation → 'semantic neighbourhoods'
- distributed features → unions and intersections
- causality → directionality
- polysemantic neurons → context-dependent



Idea

Build (filtered) **directed simplicial complex** that represents causal influence of neurons on each other.

Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 2 – Pick candidate simplex

Step 3 – Coherent counterfactual ablation

Step 4 – Statistical significance test

Step 5 – Add simplex to complex $\mathcal{K}_X$

Coherent counterfactual ablation

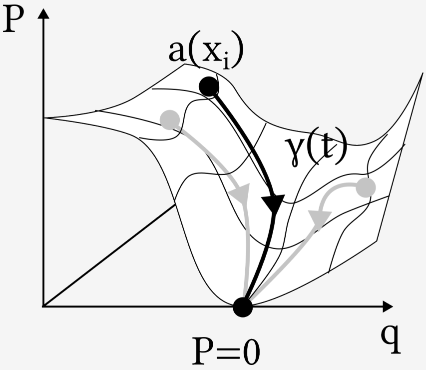
Three pieces

- *Coherent*: keep structural equations intact (stay on neural manifold M).
- *Counterfactual*: 'what if $a|_P$ was different?'
- *Ablation*: $a|_P \to 0$.

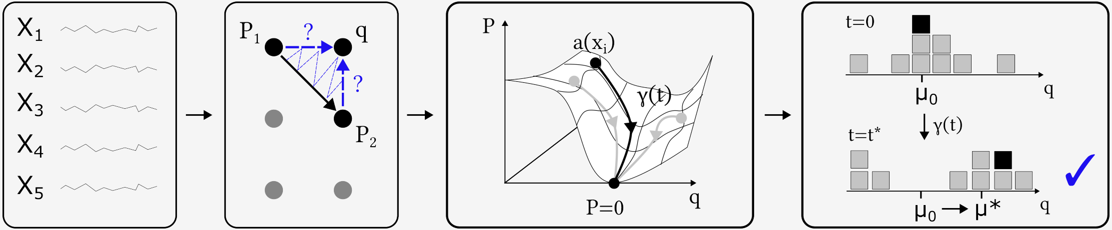
CCA = path $\gamma : [0,1] \to M$, satisfying all three.

1. Fix $x_i \in X$ with activation $a(x_i) \in M \subset \mathbb{R}^N$
2. Find tangent vector $v \in T_{a(x_i)}M$ s.t. $a(x_i)|_P$ decreases.
3. Integrate to $\gamma(t)$ (flow equation $\dot\gamma(t) = v$, $\gamma(0) = a(x_i)$)

The pullback $\gamma|_q$ quantifies the causal influence of P on the probe neuron q.



Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 2 – Pick candidate simplex

Step 3 – Coherent counterfactual ablation

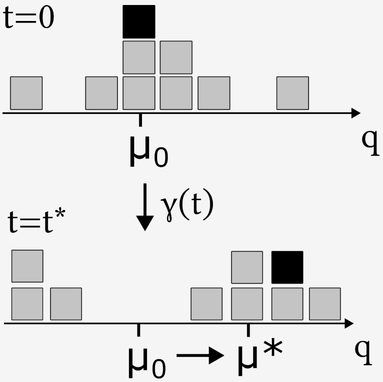
Step 4 – Statistical significance test

Step 5 – Add simplex to complex $\mathcal{K}_X$

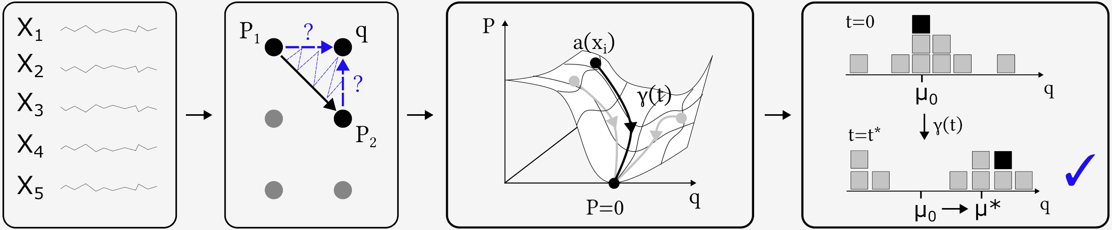
Step 4 Detail — Statistical Significance (Wilcoxon)

For probe neuron q (and $x_i \in X$)

1. Compute change of activation $\Delta q_i(t) = (\gamma_i(t) - \gamma_i(0))|_q$
2. Perform Wilcoxon signed-rank test on $\{\Delta q_i(t)\}_{x_i \in X}$
   - Null Hypothesis: $\mathrm{median}\, \Delta q_i(t) = 0$
   - Reject if $p < \alpha = 0.01$
3. $t^* = \min\{t : p(t) < \alpha\}$
   (earlier $t^* \implies$ more immediate influence)



Directed Simplicial Complexes from Neuron Activations



Step 1 – Fix 'context' X

Step 2 – Pick candidate simplex

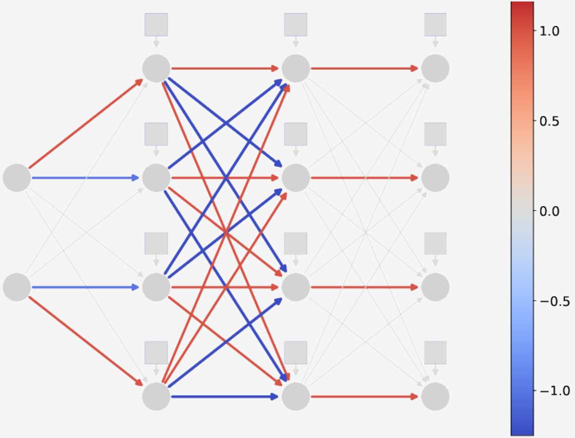
Step 3 – Coherent counterfactual ablation

Step 4 – Statistical significance test

Step 5 – Add simplex to complex $\mathcal{K}_X$

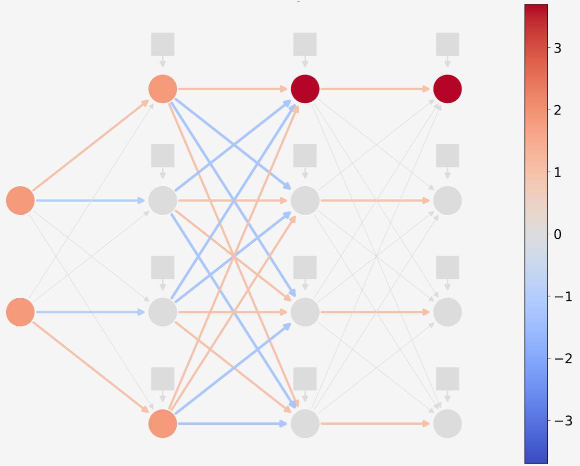
Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0, x < 0, y < 0, y > 0$
- second layer neurons: $Q1, Q2, Q3, Q4$



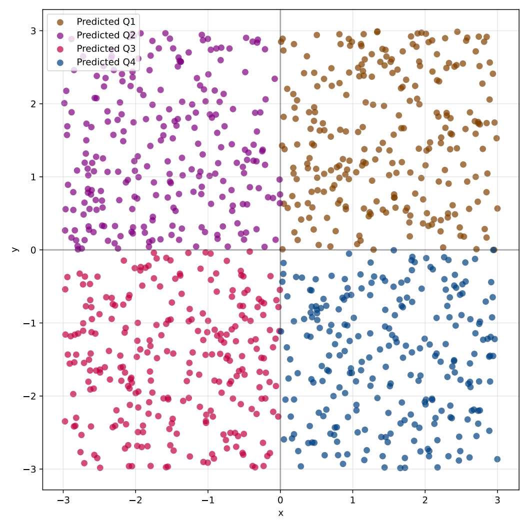
Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0$, $x < 0$, $y < 0$, $y > 0$
- second layer neurons: $Q1$, $Q2$, $Q3$, $Q4$



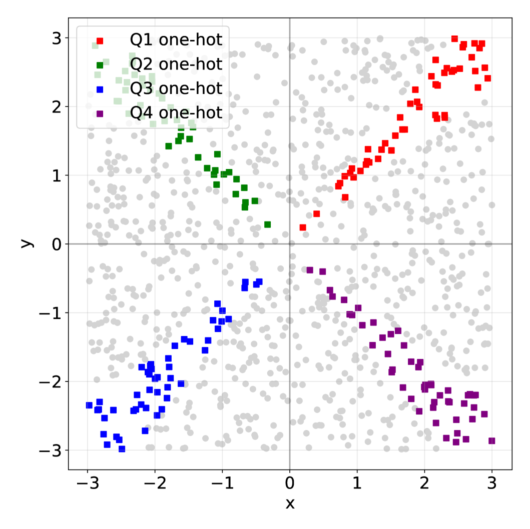
Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0$, $x < 0$, $y < 0$, $y > 0$
- second layer neurons: $Q1$, $Q2$, $Q3$, $Q4$



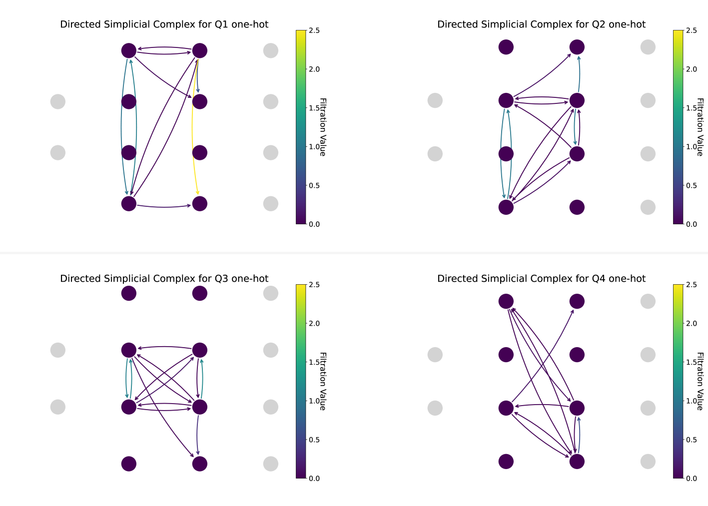
Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0$, $x < 0$, $y < 0$, $y > 0$
- second layer neurons: $Q1$, $Q2$, $Q3$, $Q4$



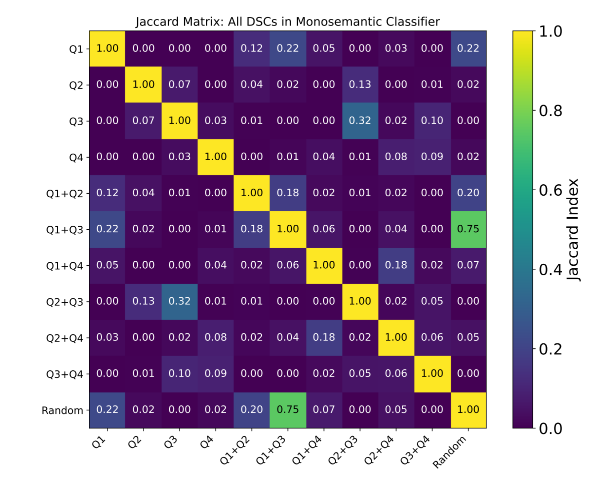
Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0$, $x < 0$, $y < 0$, $y > 0$
- second layer neurons: $Q1$, $Q2$, $Q3$, $Q4$

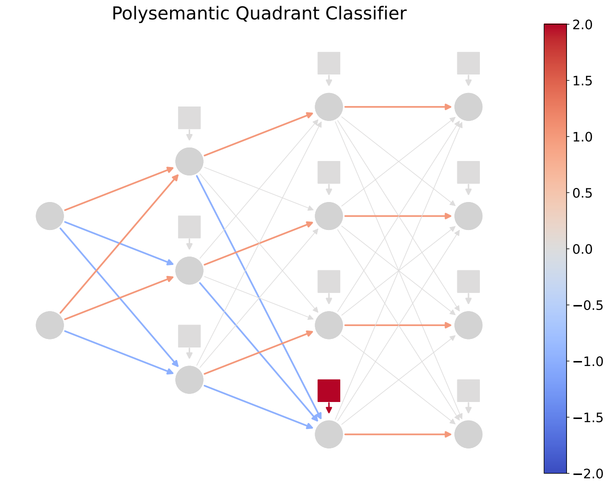


Example 1: Monosemantic Quadrant Classifier

- MLP Architecture $2 \to 4 \to 4 \to 4$ (8 internal neurons)
- Weights $\pm 1, 0$, no biases, ReLU activation
- first layer neurons: $x > 0$, $x < 0$, $y < 0$, $y > 0$
- second layer neurons: $Q1$, $Q2$, $Q3$, $Q4$

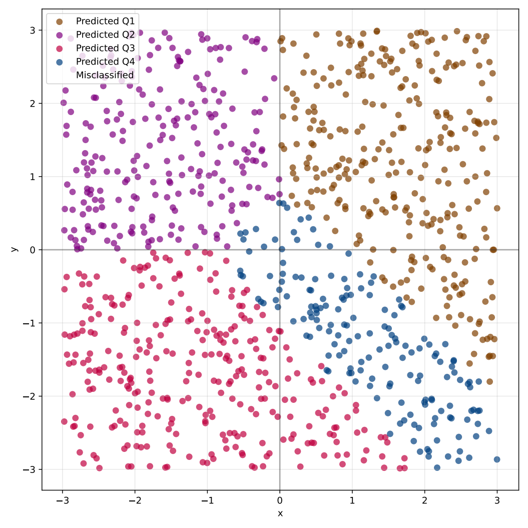


Example 2: Polysemantic Quadrant Classifier



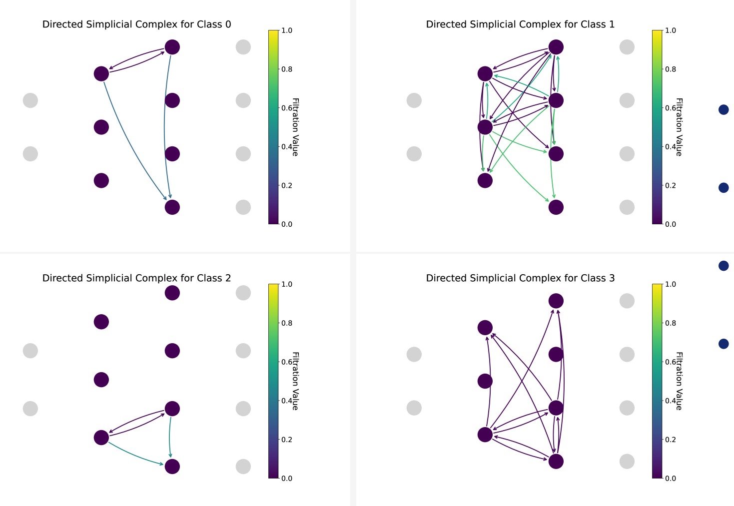
- MLP Architecture $2 \to 3 \to 4 \to 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons: $x, y > 0, \quad x < y, \quad x, y < 0$
- second layer (approx.): $b < x + y, b < y - x, b < -(x + y), x - y < b$

Example 2: Polysemantic Quadrant Classifier



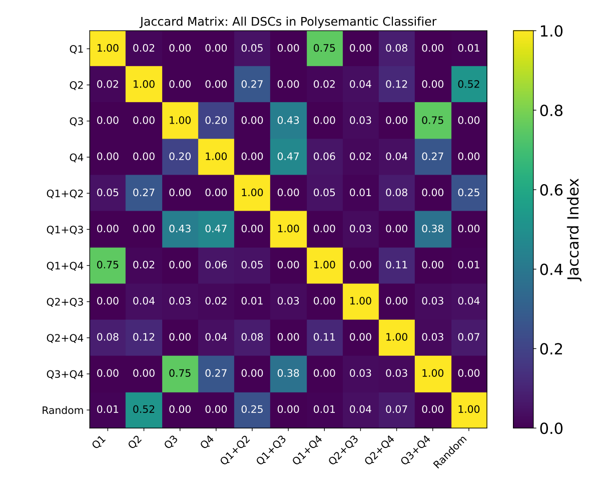
- MLP Architecture $2 \to 3 \to 4 \to 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons: $x, y > 0, \quad x < y, \quad x, y < 0$
- second layer (approx.): $b < x + y, b < y - x, b < -(x + y)$, $x - y < b$

Example 2: Polysemantic Quadrant Classifier



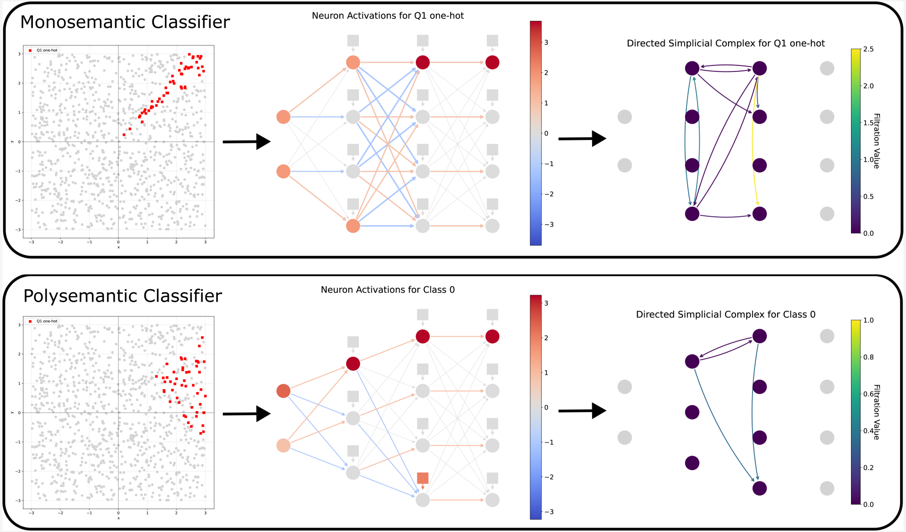
- MLP Architecture $2 \to 3 \to 4 \to 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons: $x, y > 0, \;\; x < y, \;\; x, y < 0$
- second layer (approx.): $b < x + y, b < y - x, b < -(x + y),$ $x - y < b$

Example 2: Polysemantic Quadrant Classifier



- MLP Architecture $2 \to 3 \to 4 \to 4$ (7 internal neurons)
- Weights $\pm 1, 0$, one non-zero bias b, ReLU activation
- first layer neurons: $x, y > 0, \quad x < y, \quad x, y < 0$
- second layer (approx.): $b < x + y, b < y - x, b < -(x + y), x - y < b$

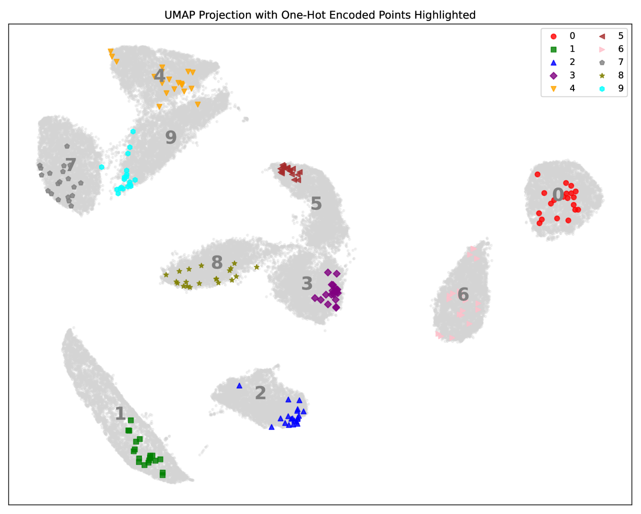
Monosemantic vs Polysemantic Quadrant Classifier



Example 3: MNIST Digit Classifier

MLP

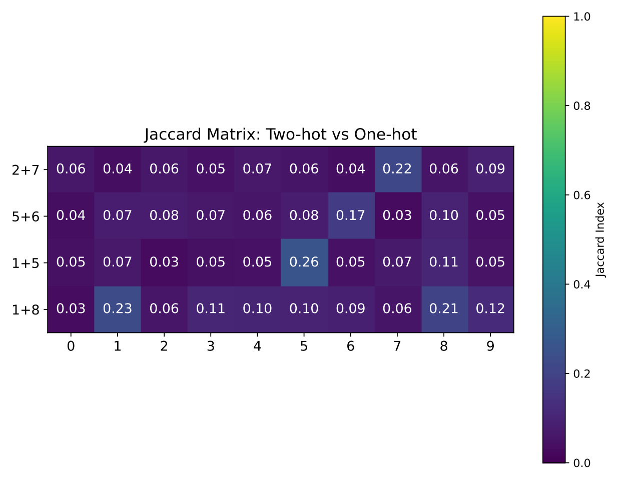
- MLP Architecture
  $784 \to 128 \to 64 \to 32 \to 16 \to 10$
  (240 internal neurons)
- trained to 97.6 % acc.
- Input samples $X_i, i = 0, \ldots, 9$, of 20 'one-hottest' inputs
- Calculated up to 2-simplices
- 7,700 1-simplices ($\approx 10\%$) and 360,000 2-simplices ($\approx 1\%$)



Example 3: MNIST Digit Classifier

MLP

- MLP Architecture
  $784 \to 128 \to 64 \to 32 \to 16 \to 10$
  (240 internal neurons)
- trained to 97.6 % acc.
- Input samples $X_i, i = 0, \ldots, 9$, of 20 'one-hottest' inputs
- Calculated up to 2-simplices
- 7,700 1-simplices ($\approx 10\%$) and 360,000 2-simplices ($\approx 1\%$)



Take-away

Directed simplicial complexes may represent *distributed features over polysemantic neurons*.

Next steps

- directed topological invariants?
- stability under weight perturbations?
- algebra of feature composition and interactions?
- unsupervised dictionary learning?