Introduction to Mechanistic Interpretability

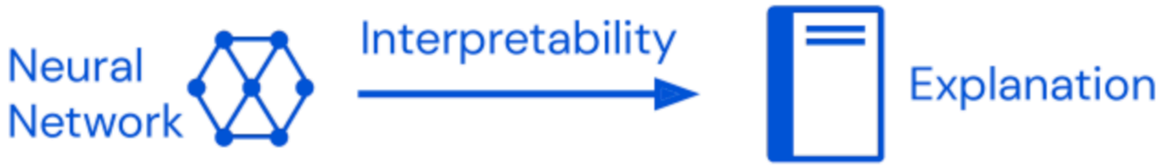
Benjamin Gerraty

SLT for Alignment Conference
21/06/2023

References

- [ToyModels] https://transformer-circuits.pub/2022/toy_model/index.html
- [InterpretableBasis] https://transformer-circuits.pub/2022/mech-interp-essay/index.html
- [SoLU] https://transformer-circuits.pub/2022/solu/index.html
- [DistillCircuits] https://distill.pub/2020/circuits/zoom-in/
- [Tracr] https://www.arxiv-vanity.com/papers/2301.05062/
- [ToyModels2] https://transformer-circuits.pub/2023/toy-double-descent/index.html
- [NeuronCapacity] https://www.alignmentforum.org/posts/kWp4R9SYgKJFHAufB/polysemanticity-and-capacity-in-neural-networks

What is Mechanistic Interpretability (MI)?



- We currently don’t know how AI are making decisions

- Decision making process is dictated by millions of parameters and are not incentivised to be interpretable by humans

- Mechanistic Interpretability hopes to Field of study about reverse engineering neural networks from their learned weights and neurons in a way that in interpretable to humans, similarly to reverse engineering computer programs

- The **curse of dimensionality** is the issue that when dealing with high-dimensional spaces, interpretability becomes very hard

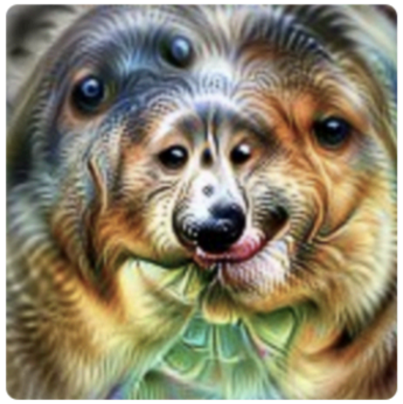
- To deal with this MI’s fundamental objects of study are **features** and **circuits**

[DistillCircuits]

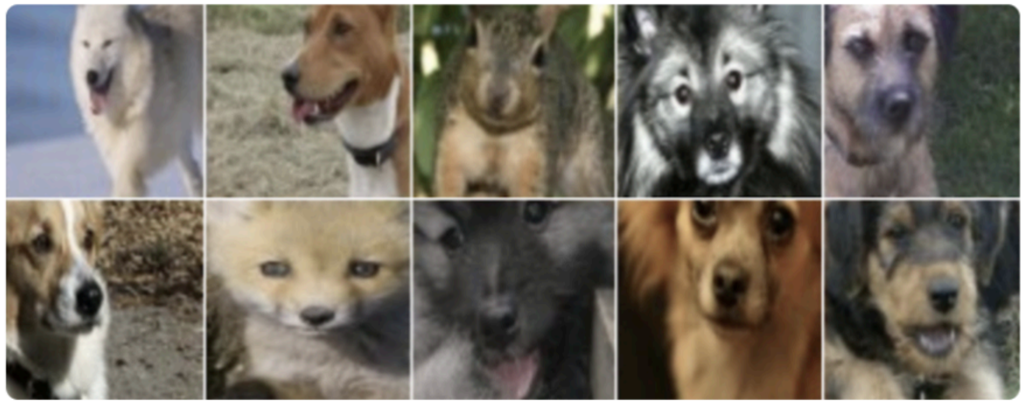
Features

- What would we want a feature of an interpretable neural network to be?
- Activation functions (neurons) represent features and the models parameters
- Possible definitions of a **feature** include
    - A **feature** is a property of the input which a sufficiently large neural network will reliably dedicate a neuron to representing.
    - A **feature** is a function of the input of the model
    - A **feature** is a something that is interpretable for humans
- Features are the fundamental building blocks of models
- Features are a fuzzy definition, best learned via example

Features



Neuron 4b:409



Dataset examples for neuron 4b:409

Curves

3b:379

3b:406

3b:385

3b:343

3b:342

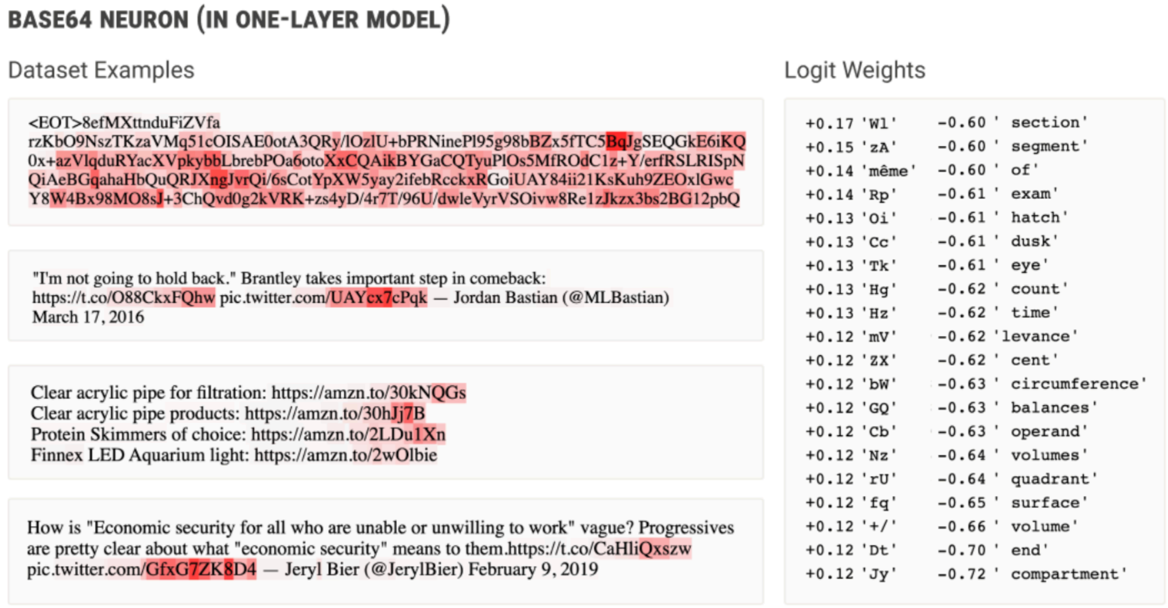
3b:388

3b:340

3b:330

3b:349

3b:324



BASE64 NEURON (IN ONE-LAYER MODEL)

Dataset Examples

<EOT>8efMXttnduFiZVfa
rzKbO9NszTKzaVMq51cOISAE0otA3QRy/lOzlU+bPRNinePl95g98bBZx5fTC5BqJgSEQGkE6iKQ
0x+azVlqduRYacXVpkybbLbrebPOa6otoXxCQAikBYGaCQTyuPlOs5MfROdC1z+Y/erfRSLRISpN
QiAeBGqahaHbQuQRJXngJvrQi/6sCotYpXW5yay2ifebRcckxRGoiUAY84ii21KsKuh9ZEOxlGwc
Y8W4Bx98MO8sJ+3ChQvd0g2kVRK+zs4yD/4r7T/96U/dwleVyrVSOivw8Re1zJkzx3bs2BG12pbQ

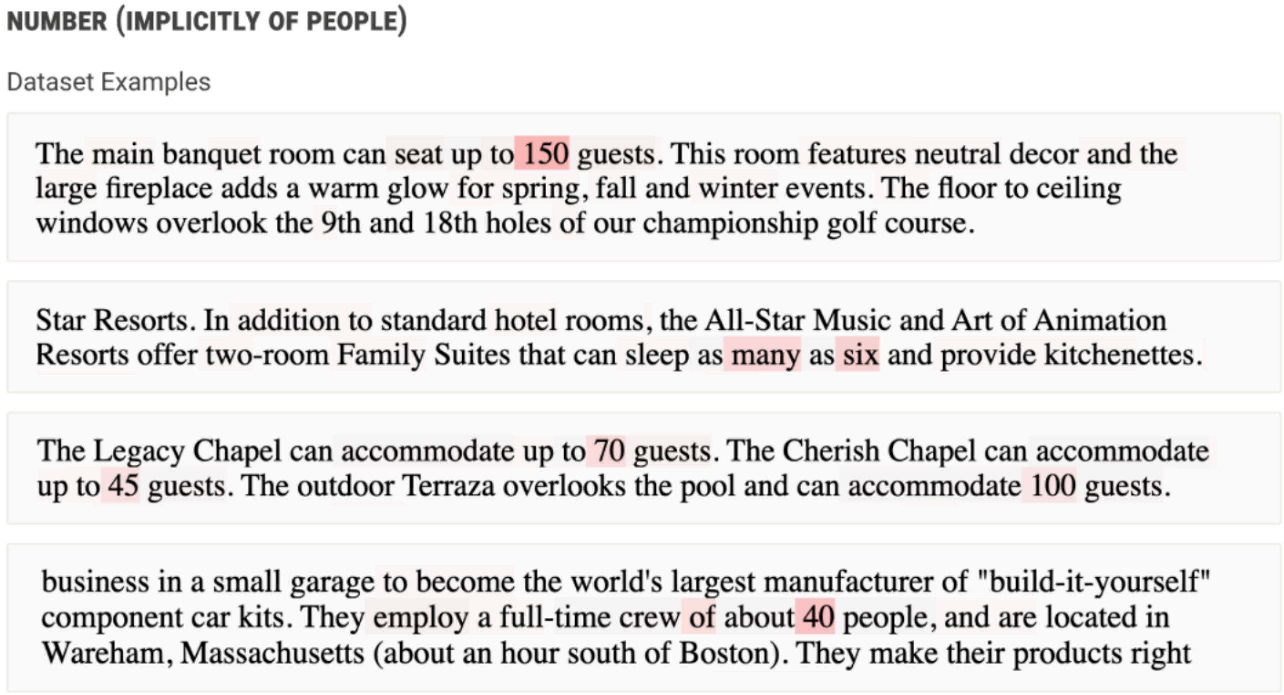
"I'm not going to hold back." Brantley takes important step in comeback:
https://t.co/O88CkxFQhw pic.twitter.com/UAYcx7cPqk — Jordan Bastian (@MLBastian)
March 17, 2016

Clear acrylic pipe for filtration: https://amzn.to/30kNQGs
Clear acrylic pipe products: https://amzn.to/30hJj7B
Protein Skimmers of choice: https://amzn.to/2LDu1Xn
Finnex LED Aquarium light: https://amzn.to/2wOlbie

How is "Economic security for all who are unable or unwilling to work" vague? Progressives are pretty clear about what "economic security" means to them.https://t.co/CaHliQxszw pic.twitter.com/GfxG7ZK8D4 — Jeryl Bier (@JerylBier) February 9, 2019

Logit Weights

```
+0.17 'Wl'     -0.60 ' section'
+0.15 'zA'     -0.60 ' segment'
+0.14 'même'   -0.60 ' of'
+0.14 'Rp'     -0.61 ' exam'
+0.13 'Oi'     -0.61 ' hatch'
+0.13 'Cc'     -0.61 ' dusk'
+0.13 'Tk'     -0.61 ' eye'
+0.13 'Hg'     -0.62 ' count'
+0.13 'Hz'     -0.62 ' time'
+0.12 'mV'     -0.62 'levance'
+0.12 'ZX'     -0.62 ' cent'
+0.12 'bW'     -0.63 ' circumference'
+0.12 'GQ'     -0.63 ' balances'
+0.12 'Cb'     -0.63 ' operand'
+0.12 'Nz'     -0.64 ' volumes'
+0.12 'rU'     -0.64 ' quadrant'
+0.12 'fq'     -0.65 ' surface'
+0.12 '+/'     -0.66 ' volume'
+0.12 'Dt'     -0.70 ' end'
+0.12 'Jy'     -0.72 ' compartment'
```



NUMBER (IMPLICITLY OF PEOPLE)

Dataset Examples

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.

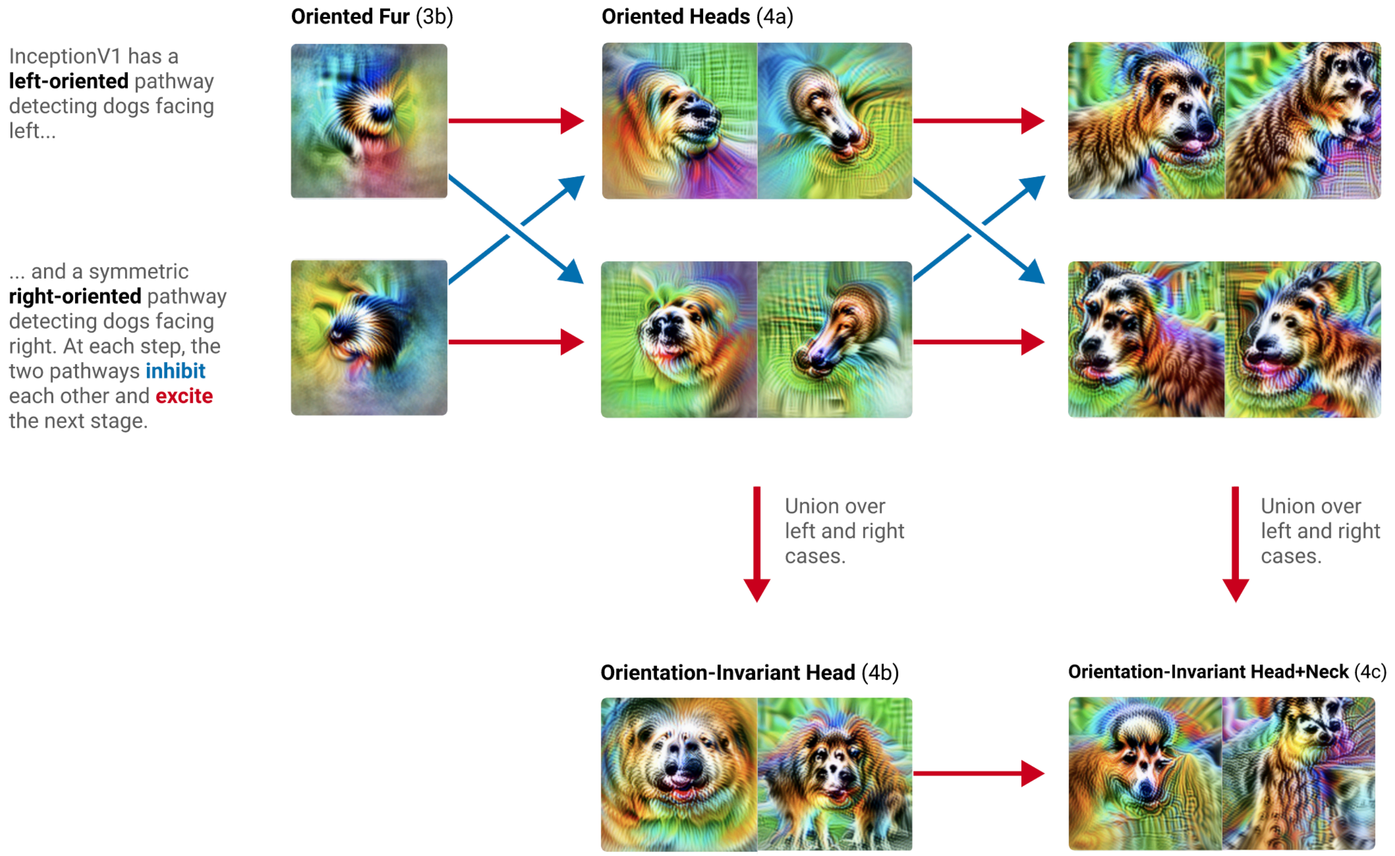
The Legacy Chapel can accommodate up to 70 guests. The Cherish Chapel can accommodate up to 45 guests. The outdoor Terraza overlooks the pool and can accommodate 100 guests.

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

[DistillCircuits]

Circuits

- What would we expect interpretable models to be doing with these features?
- **Circuits** are algorithms that the model learns, a subset of the models neurons (features) and weights that reliably and repeatedly perform the same task
- We will see tomorrow that Induction heads form in Transformers as a circuit
- Once again a fuzzy definition, best illustrated via examples



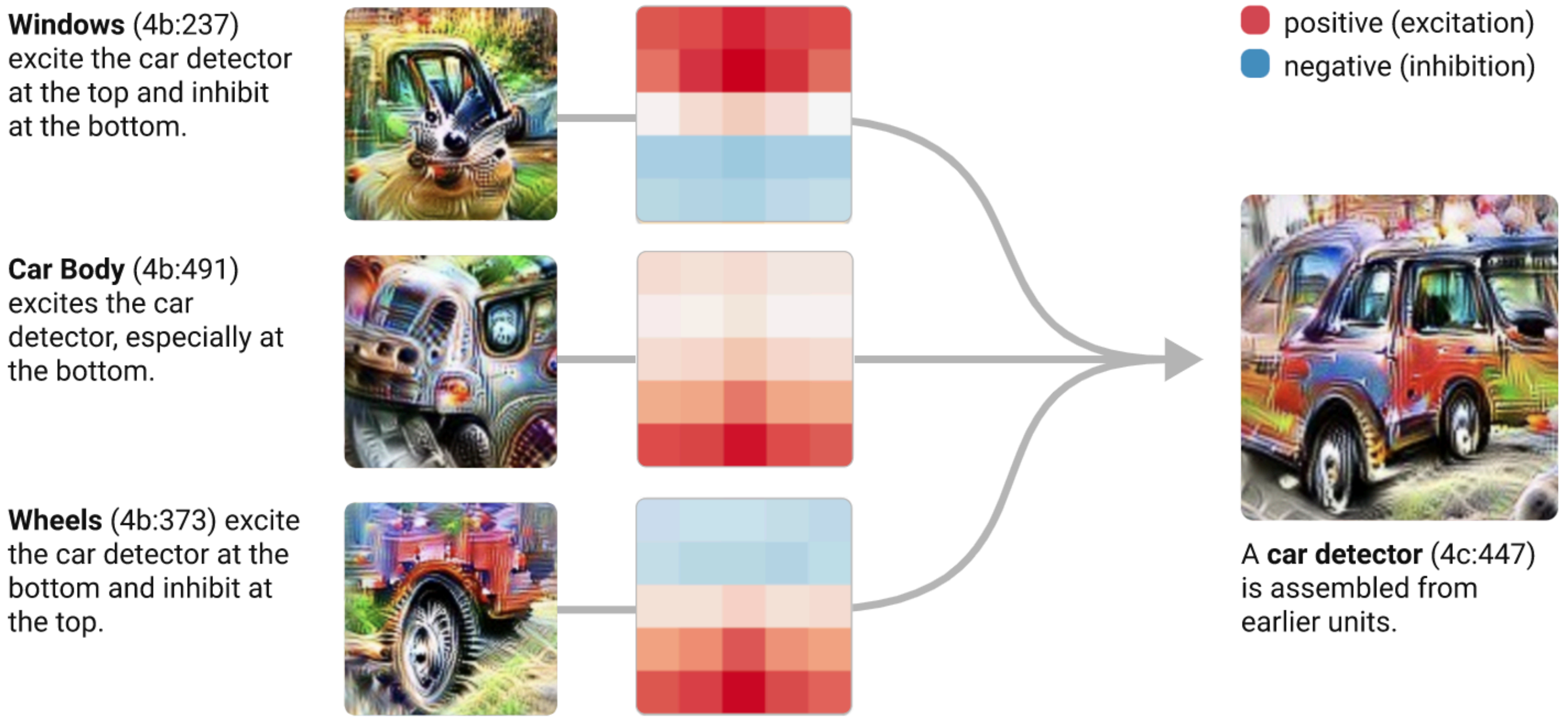
InceptionV1 has a **left-oriented** pathway detecting dogs facing left...

... and a symmetric **right-oriented** pathway detecting dogs facing right. At each step, the two pathways **inhibit** each other and **excite** the next stage.

Union over left and right cases.

Union over left and right cases.

Oriented Fur (3b)
Oriented Heads (4a)
Orientation-Invariant Head (4b)
Orientation-Invariant Head+Neck (4c)



Windows (4b:237) excite the car detector at the top and inhibit at the bottom.

Car Body (4b:491) excites the car detector, especially at the bottom.

Wheels (4b:373) excite the car detector at the bottom and inhibit at the top.

A **car detector** (4c:447) is assembled from earlier units.

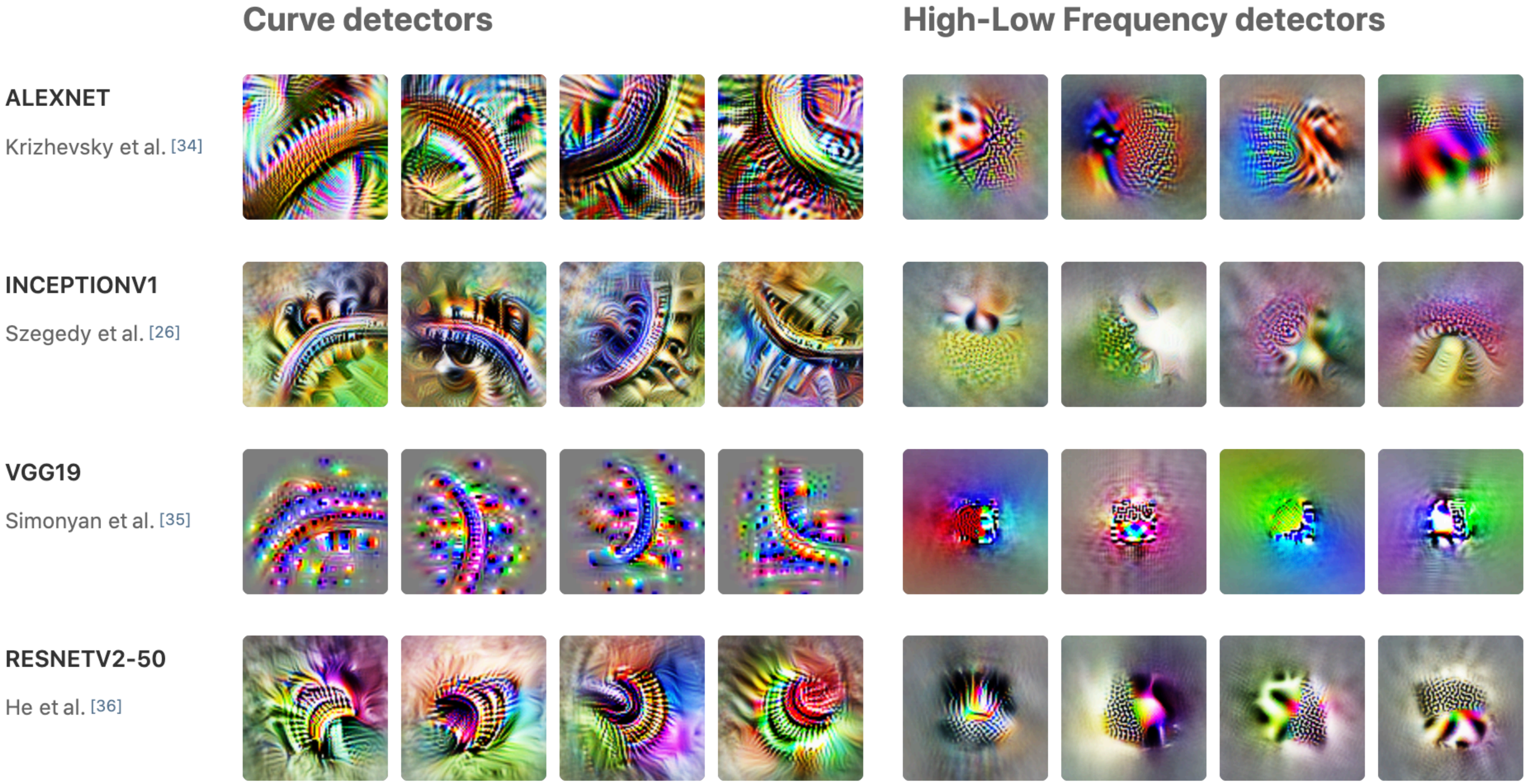
[DistillCircuits]

Universality Hypothesis

- Hypothesis that the same features and circuits will appear in the different models
- **Weak Universality**: Handful of algorithms that work over different models and scales
- **Strong Universality**: All models perform the same circuits regardless of type of model or scale
- **Universality of principles**: There are techniques that models learn that work across a wide range of models, even if the circuits are not the same

[DistillCircuits]

Universality Hypothesis



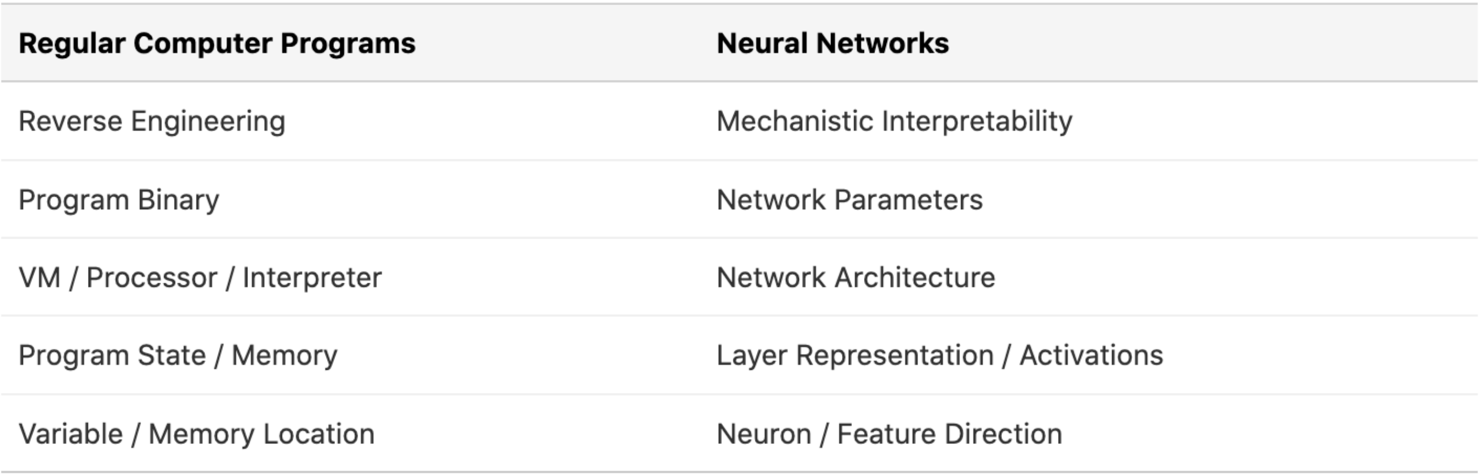
[Tracr]

How do these ideas help?

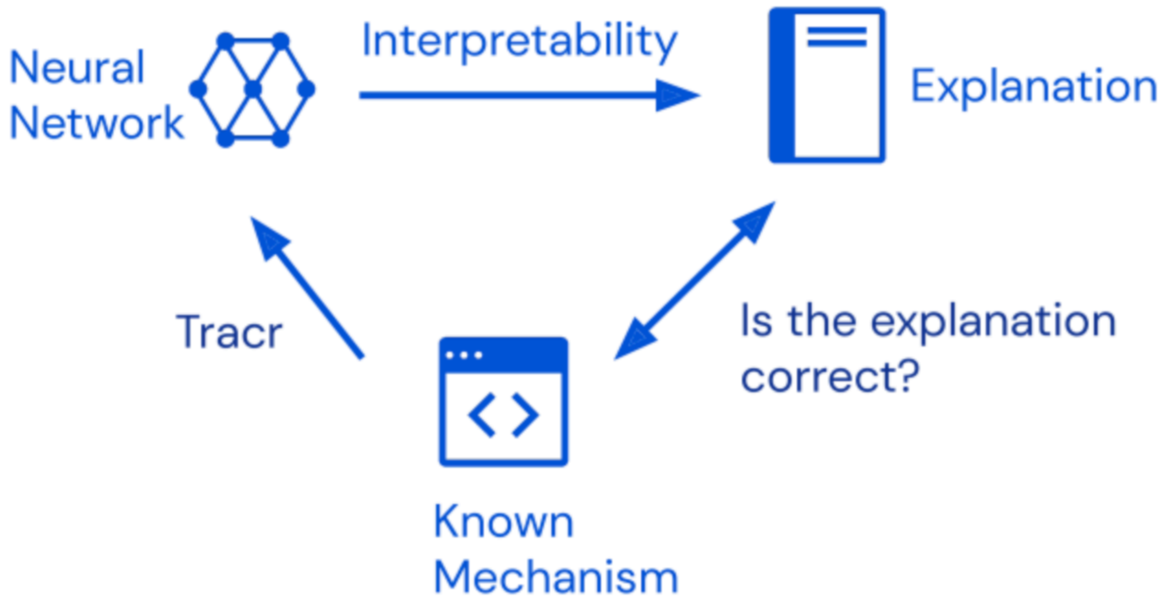
- By extracting features and circuits from models we can hopefully pull out circuits and features we know

- Then implement the weights manually in a new network and check if the hand coded model performs as intended

- **Tracr** is a compiler for translating human readable programs into weights of a decoder only (GPT-like) transformer



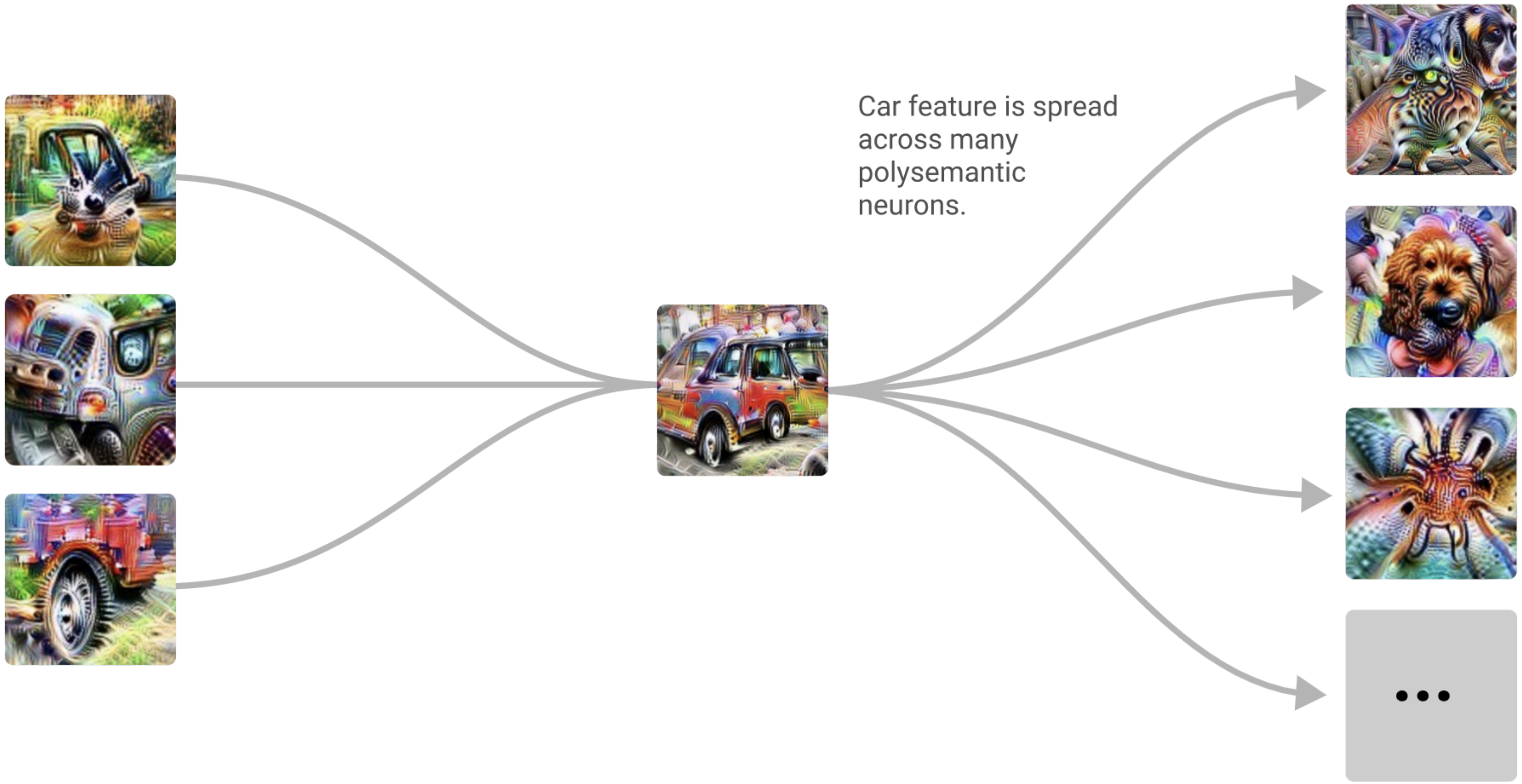
Regular Computer Programs	Neural Networks
Reverse Engineering	Mechanistic Interpretability
Program Binary	Network Parameters
VM / Processor / Interpreter	Network Architecture
Program State / Memory	Layer Representation / Activations
Variable / Memory Location	Neuron / Feature Direction



[DistillCircuits]

Features in Superposition

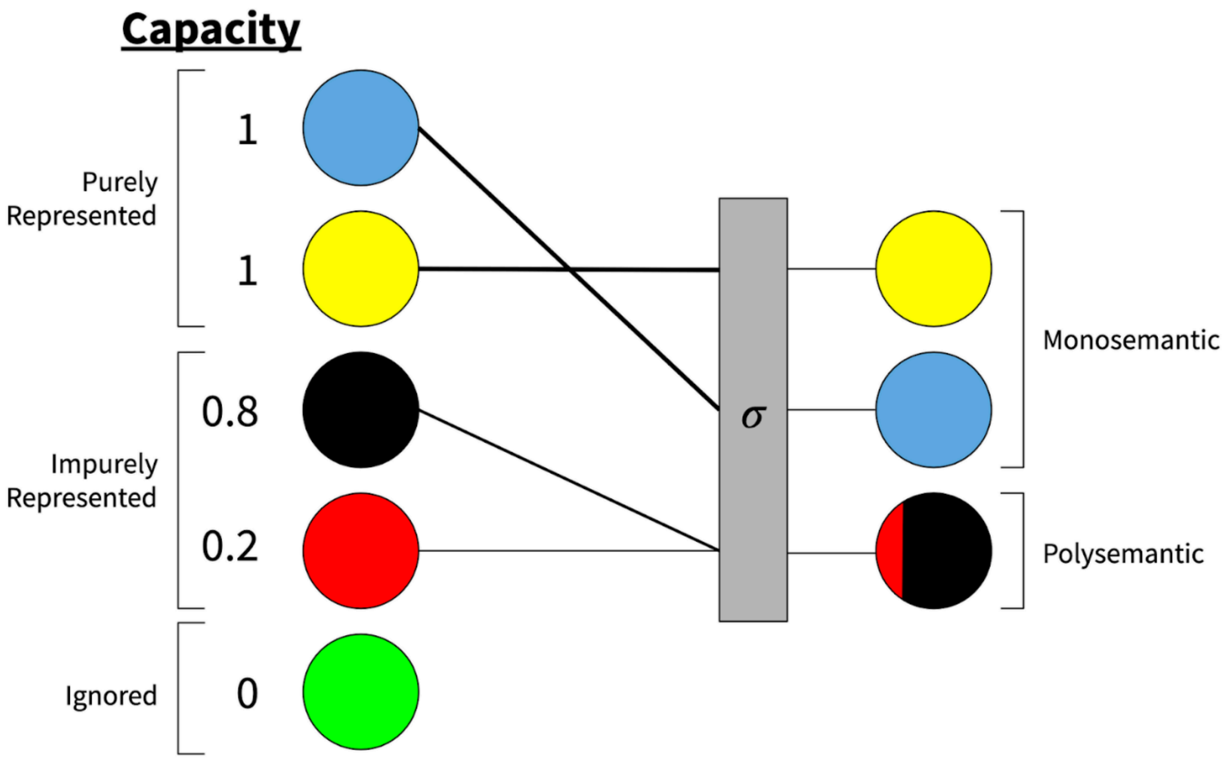
- So far we've painted a very nice picture about how to interpret neural networks, first they learn features, then they combine features in circuits to identify more complicated features
- Unfortunately not every neuron cleanly aligns with a feature, most are actually in **superposition** or split across neurons
- even after a neuron is dedicated to a feature it still needs to carry that information to the output of the model



Car feature is spread across many polysemantic neurons.

Superposition - Why do models do it?

- Most features in a model are sparse, in the sense that they won't be seen by the model very often
- So the model can then "take the risk" to put multiple features that are unlikely to occur together into one polysemantic neuron
- Features may also vary in importance depending on the context of the task



Can we remove Superposition?

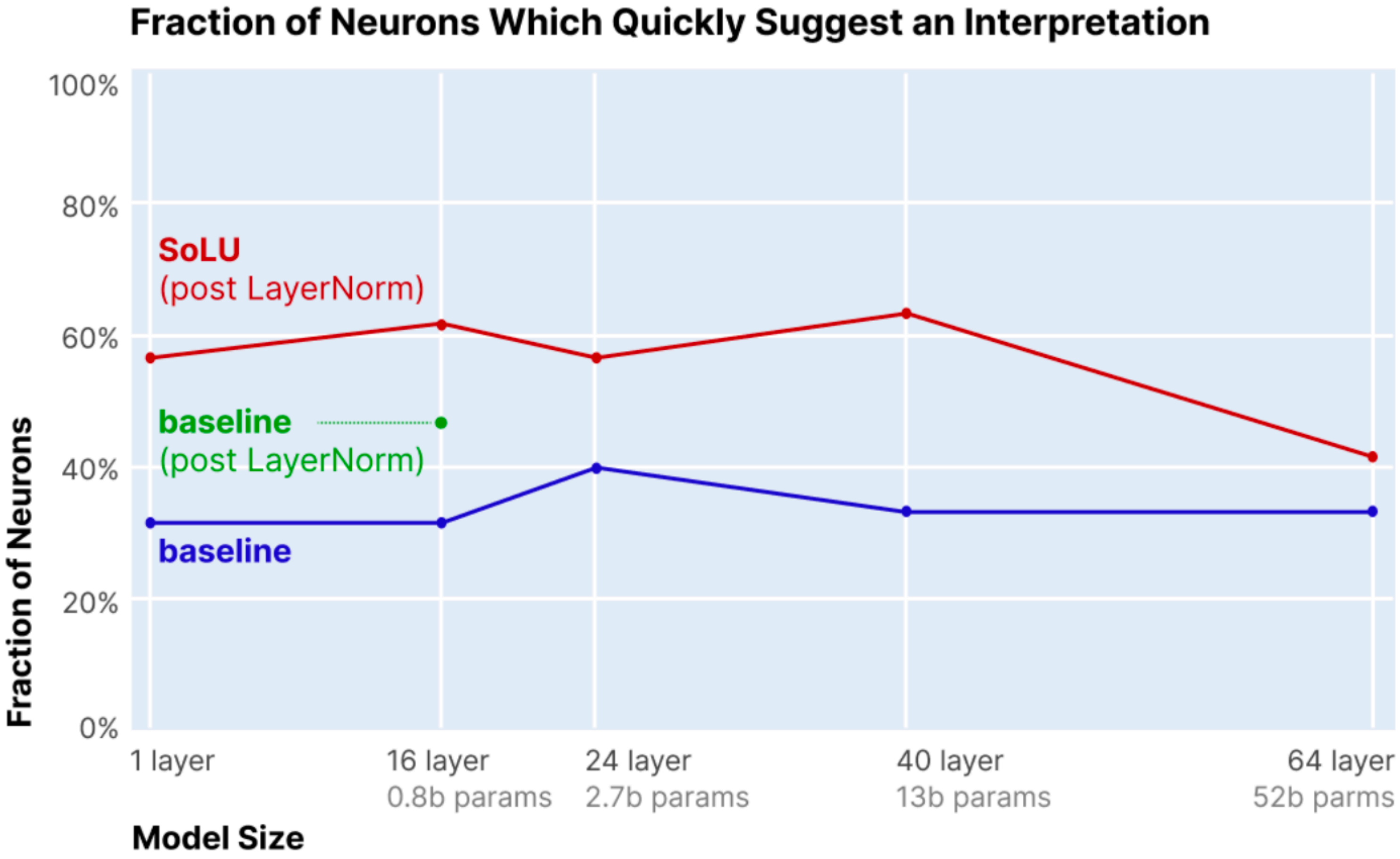
- Clearly superposition is deeply connected to the challenge of interpretability to make claims about AI safety

- Superposition also has the challenge that if features are always in superposition we may never know they are there, which is concerning especially if there are harmful features

- One can make more features interpretable in a model, for example using an activation function in a 1-layer transformer a SoLU activation function can generate more interpretable features that ReLU

- It seems like superposition is here to stay in some sense if we demand the best performance a model can give

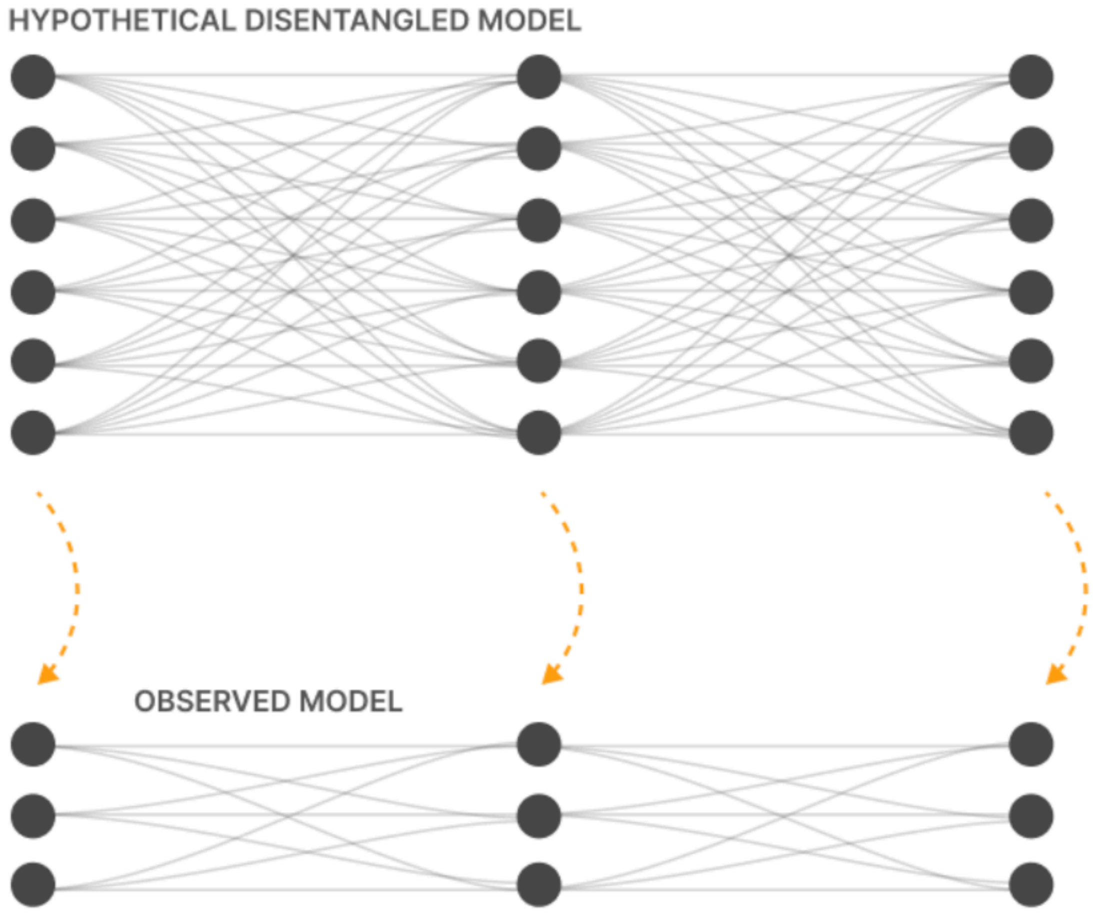
$$\textbf{SoLU}(x) = x\ \textbf{softmax}(x)$$



[ToyModels]

The Superposition Hypothesis

- The Superposition Hypothesis states that given a model with features in superposition, there exists a much larger model where every feature is given a dedicated neuron



Under the superposition hypothesis, the neural networks we observe are **simulations of larger networks** where every neuron is a disentangled feature.

These idealized neurons are **projected** on to the actual network as “almost orthogonal” vectors over the neurons.

The network we observe is a **low-dimensional projection** of the larger network. From the perspective of individual neurons, this presents as polysemanticity.

[ToyModels]

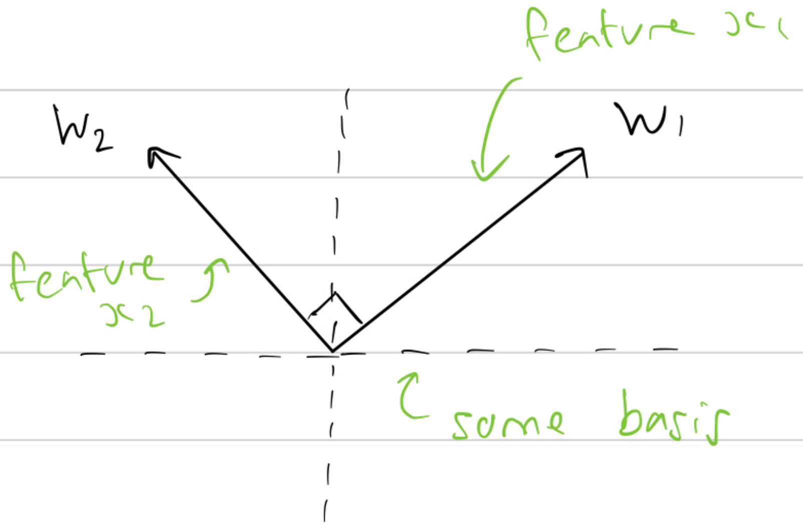
Features as Directions

- This is the hypothesis is that features are represented as as directions in activation space
- The intuition behind this is that one of the main tasks models do well is linear algebra
- Tokens certain in embedded vector space obey interpretable algebra such as V("king") - V("man") + V("woman") = V(“queen") (this is motivational only) without buying into the algebra too much, it motivates the orthogonality
- Let $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ be a vector of n features, if each x_i can be embedded into a vector space by multiplying x by a matrix $W \in \mathbb{R}^{n \times n}$ giving Wx. we can plot the columns of W as W_i corresponding to feature x_i
- For example $n = 2$, if we can plot the features as columns of W in $\mathbb{R}^2$

$$Wx = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ w_1 & w_2 & \cdots & w_n \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

column $w_i = (w_{1i}, w_{2i}, \ldots, w_{ni}) \in \mathbb{R}^n$

embedding n features into n dimensions

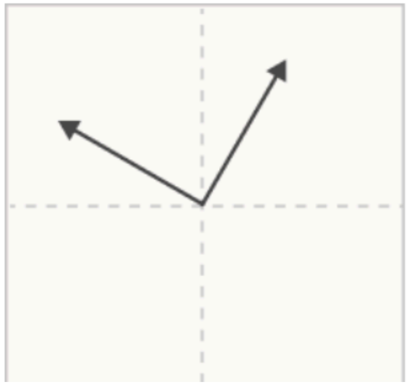


$n = 2$

2 features in $\mathbb{R}^2$

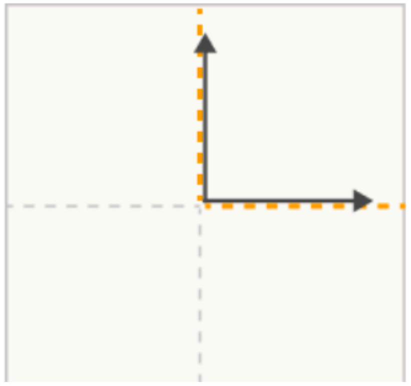
Do we care about the basis?

- Even if features are encoded as directions, a natural question to ask is which directions? In some cases, it seems useful to consider the basis directions, but in others it doesn't. Why is this?
- Superposition occurs when features have non-zero dot products
- From this perspective, it only makes sense to ask if a *neuron* is interpretable when it is in a privileged basis.
- Note that having a privileged basis doesn't guarantee that features will be basis-aligned – we'll see that they often aren't! But it's a minimal condition for the question to even make sense.



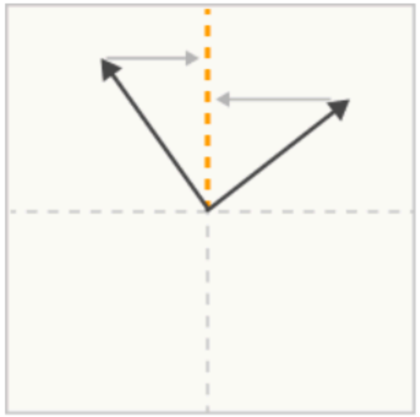
In a **non-privileged basis**, features can be embedded in any direction. There is no reason to expect basis dimensions to be special.

Examples: word embeddings, transformer residual stream

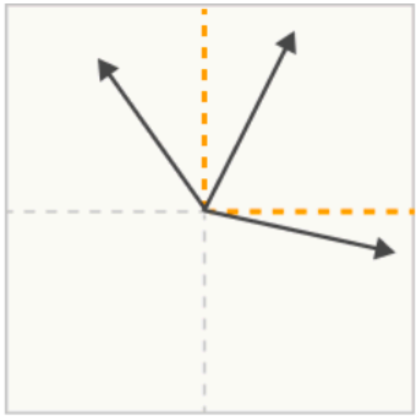


In a **privilged basis**, there is an incentive for features to align with basis dimensions. This doesn't necessarily mean they will.

Examples: conv net neurons, transformer MLPs



Polysemanticity is what we'd expect to observe if features were not aligned with a neuron, despite incentives to align with the privileged basis.



In the **superposition hypothesis**, features can't align with the basis because the model embeds more features than there are neurons. Polysemanticity is inevitable if this happens.

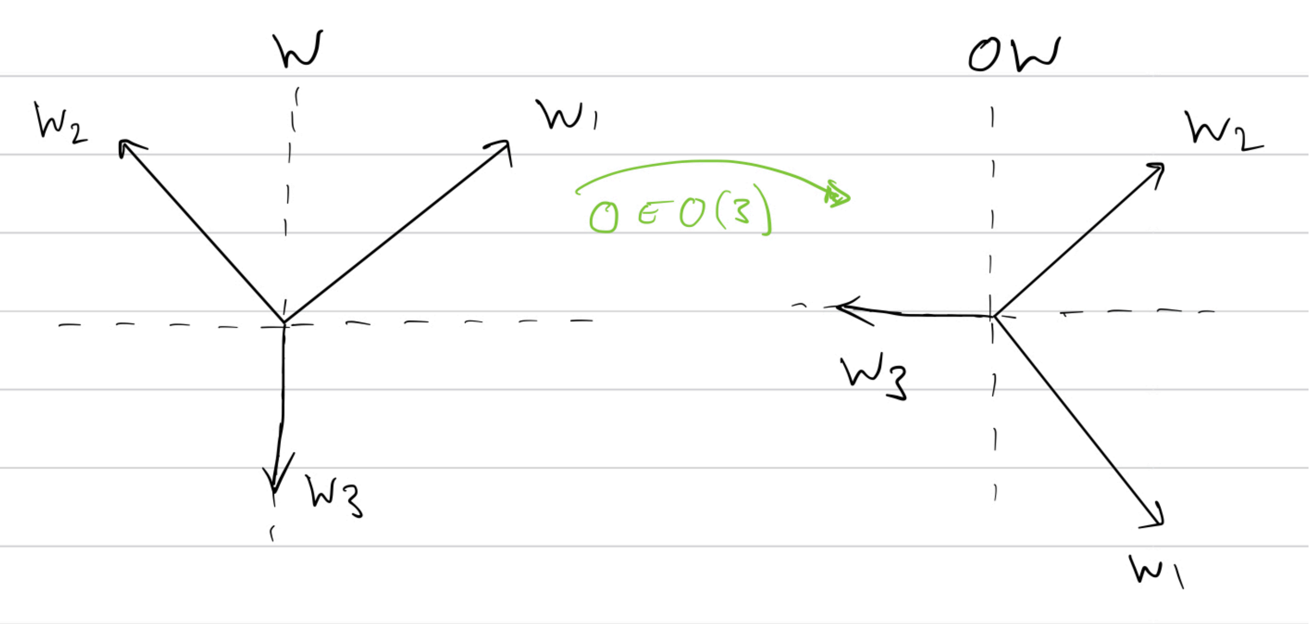
Toy Models of Superposition

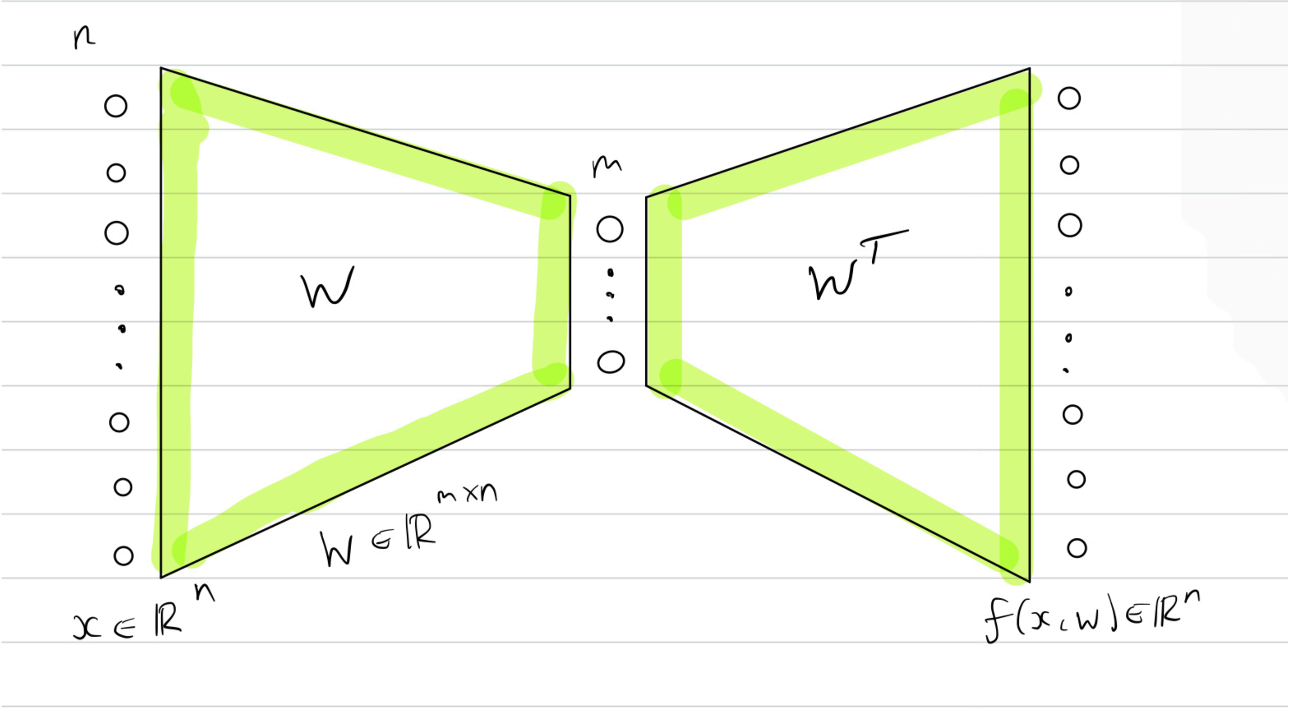
- So if we embed more features in than we have dimensions, then the model with HAVE to use to superposition! Therefore our Toy Model of Superposition will be embedding n features $(x_1, \dots, x_n) \in \mathbb{R}^n$ into an $m-$dimensional space where $n > m$

- A linear representation W exhibits superposition if $W^T W$ is not invertible and does not exhibit superposition if it is. For $x, b \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}$, we define $w = (W, b)$. So

$$x \in \mathbb{R}^n, \quad x_i \sim \textbf{U[0,1]} \qquad f(x, w) = \textbf{ReLU}(W^T W x + b)$$

- For example $n = 3$ features into $m = 2$ dimensional space we do not have a privilege basis due to rotation invariance





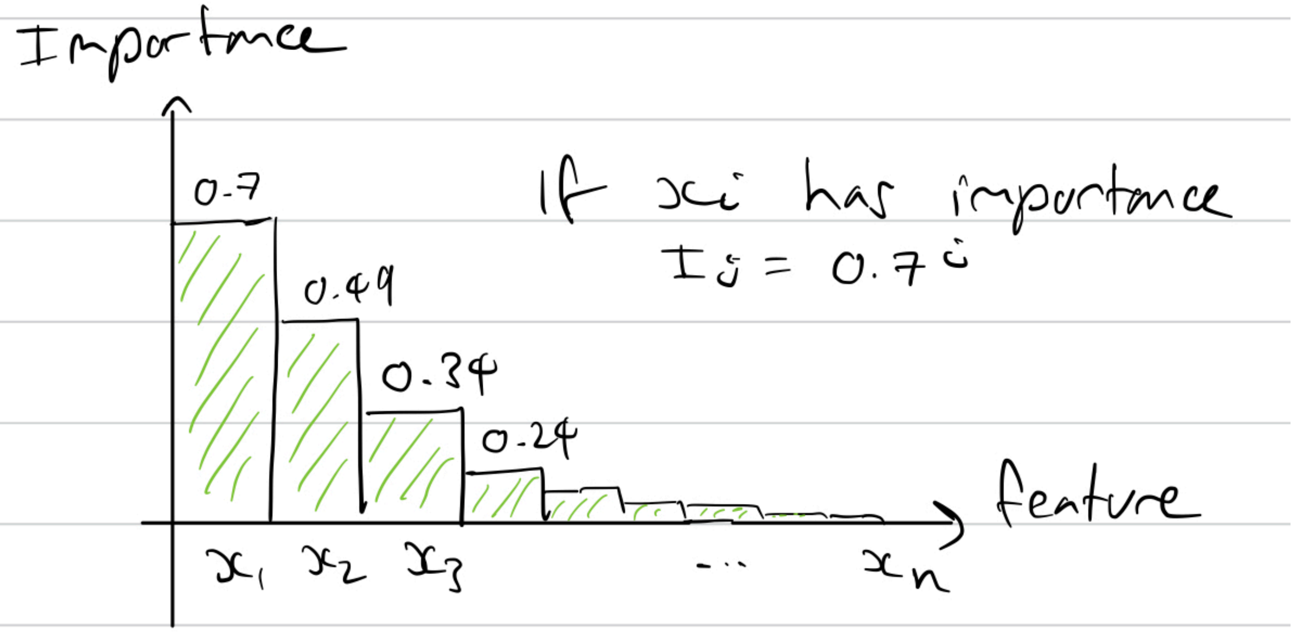
Features (Parameters) of Features

- We will see how the parameters **sparsity** and **importance** affect feature embedding

- Suppose $x_i = 0$ with probability s, otherwise $x_i \sim$ U[0,1] as before we call s the sparsity

- Each feature x_i will be given an importance I_i where more important features (larger I) will give a lower loss if embedded

$$x = (x_1, 0, \ldots, 0, x_n)$$



If x_i has importance $I_i = 0.7^i$

[ToyModels]

Loss of the Toy Models of Superposition

The loss we wish to minimise is

$$K(w) = \int_{\mathbb{R}^n} q(x,s)\,||I(x - f(x,w))||^2 dx$$
$$= \sum_{i=1}^{n} \int_{\mathbb{R}^n} q(x,s) I_i (x_i - f(x,w)_i)^2 dx$$

where I is the importance matrix, s is the sparsity and

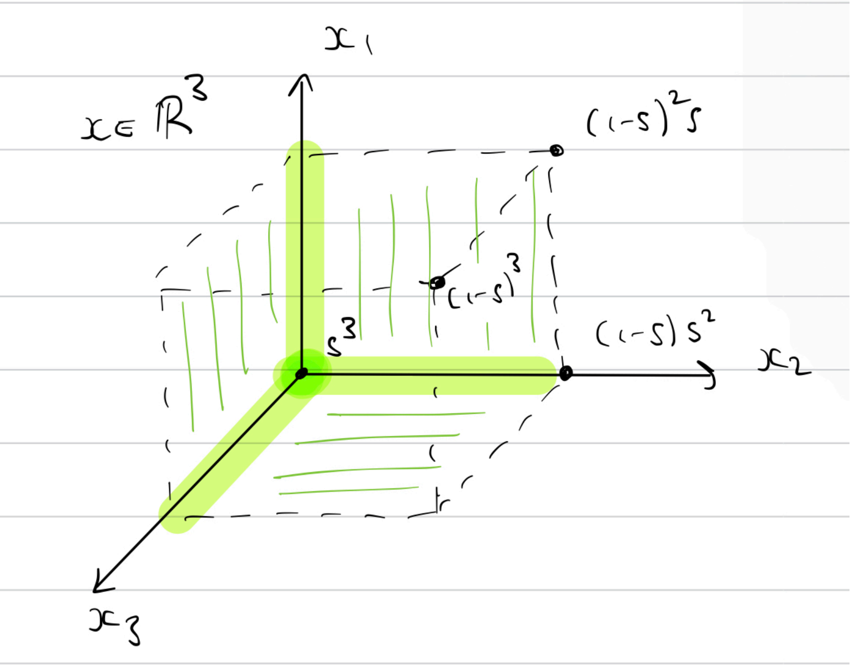
$$q(x,s) = \sum_T \binom{n}{|T|}^{-1} s^{|T|}(1-s)^{n-|T|}\delta_{x\in[0,1]^T}$$

Here $T = \{1,\ldots,n\}$. The model once again is

$$f(x,w) = \textbf{ReLU}(W^T W x + b)$$

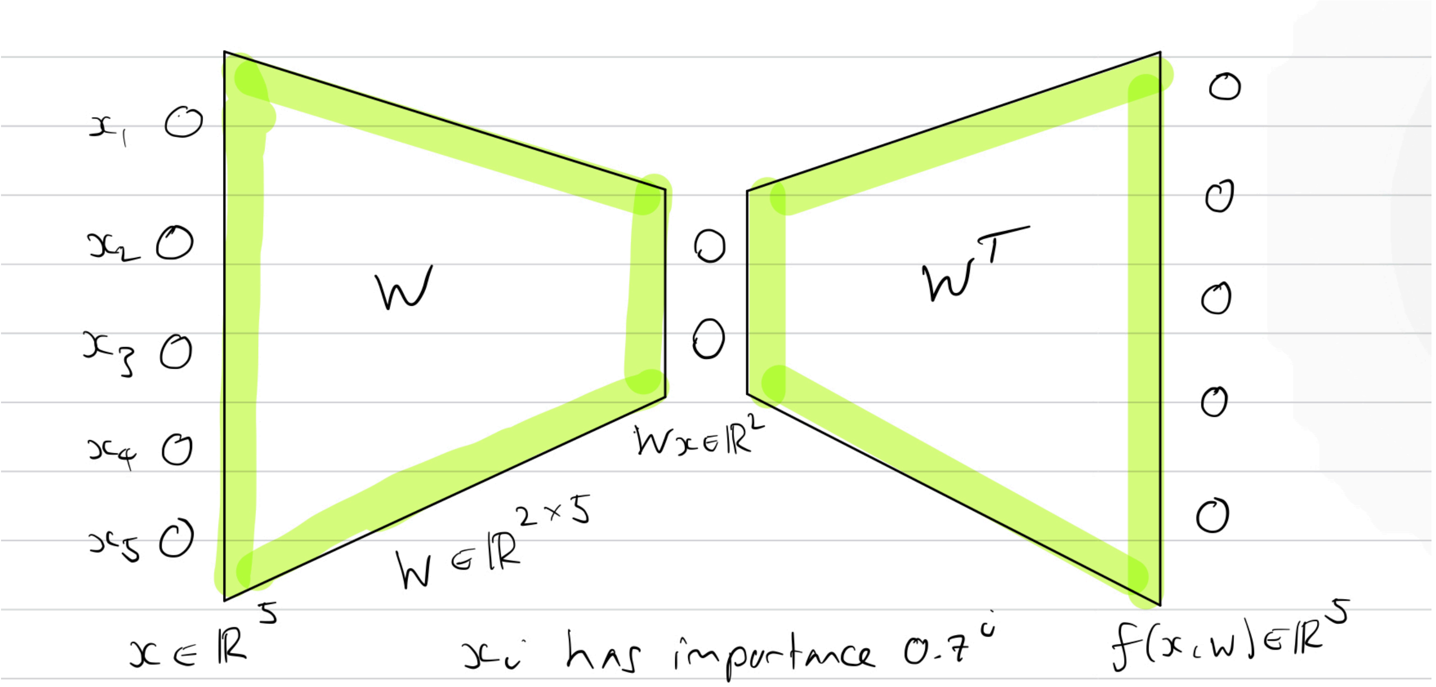
The paper gives the loss as $L = \int_{\mathbb{R}^n} \sum_{i=1}^{n} I_i (x_i - f(x,w)_i)^2 dx$

$x \in \mathbb{R}^3, x_i = 0$ with probability s

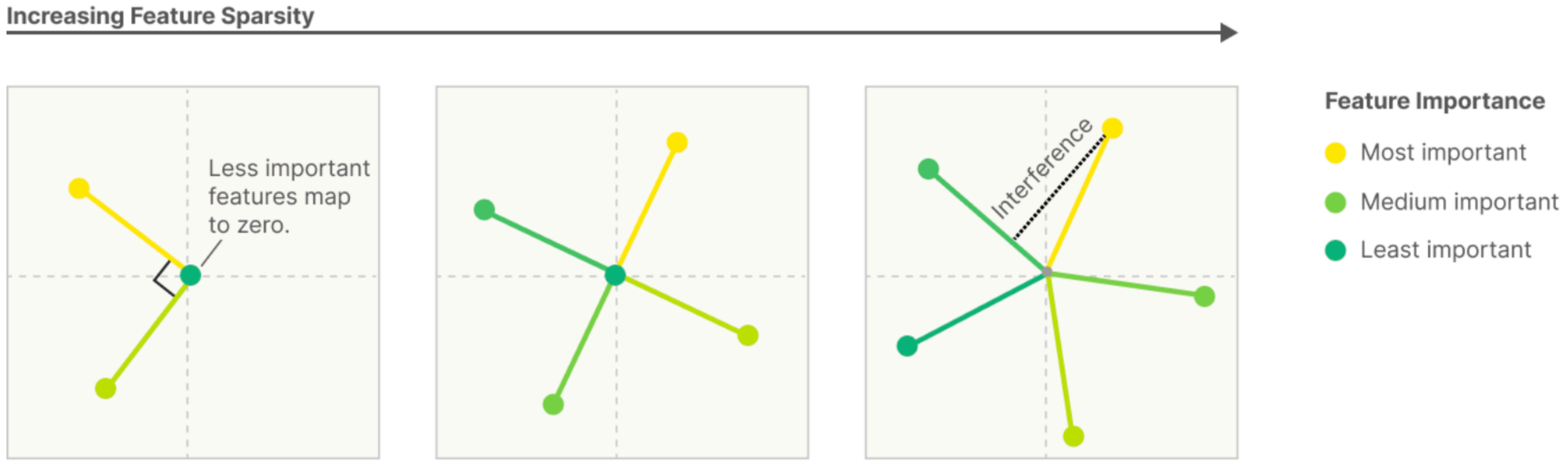


[ToyModels]

5 Features in 2 Dimensions


x_1
x_2
x_3
x_4
x_5
W
W^T
$Wx \in \mathbb{R}^2$
$W \in \mathbb{R}^{2 \times 5}$
$x \in \mathbb{R}^5$
x_i has importance 0.7^i
$f(x, W) \in \mathbb{R}^5$

As Sparsity Increases, Models Use “Superposition” To Represent More Features Than Dimensions



0% Sparsity

The two most important features are given **dedicated orthogonal dimensions**, while other features are **not embedded**.

80% Sparsity

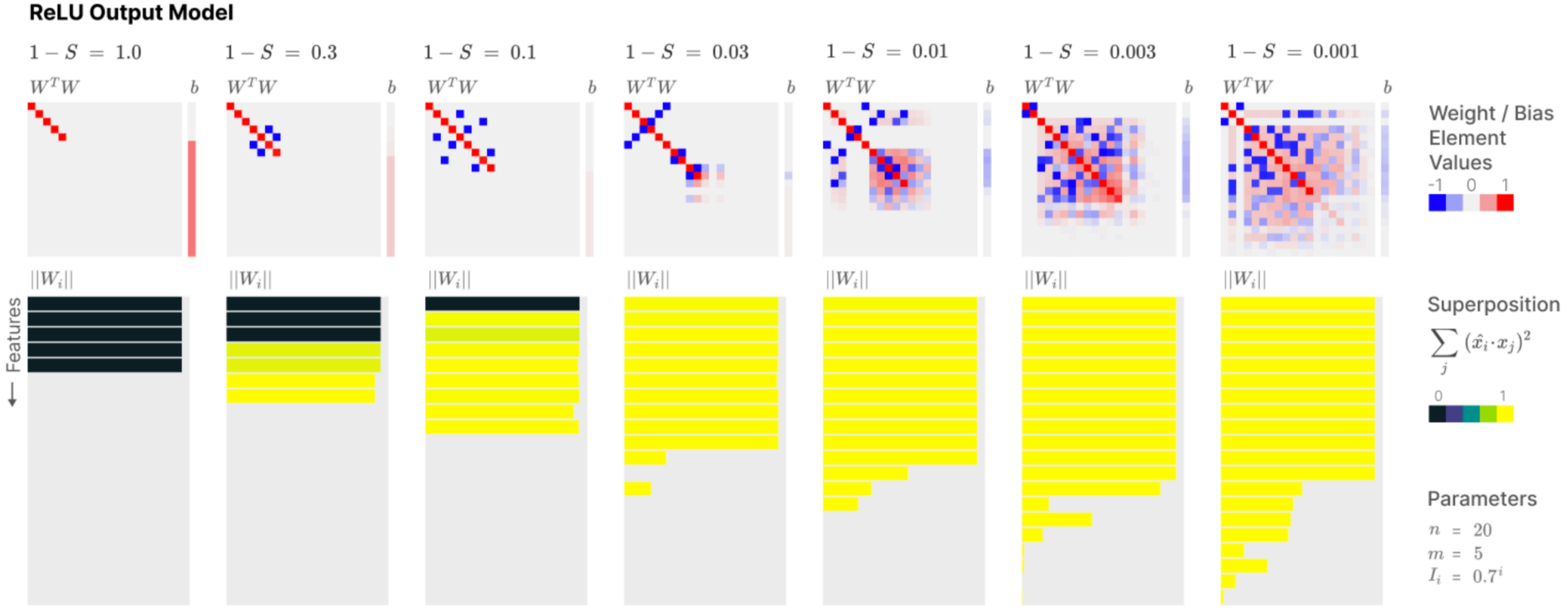
The four most important features are represented as **antipodal pairs**. The least important features are **not embedded**.

90% Sparsity

All five features are embedded **as a pentagon**, but there is now “positive interference.”

[ToyModels]

Higher Sparsity Embeds More Features



In the **dense** regime, ReLU output models also learn the top m features.

As **sparsity increases**, superposition allows models to represent more features. The most important features are initially untouched. This early superposition is organized in antipodal pairs (more on this later).

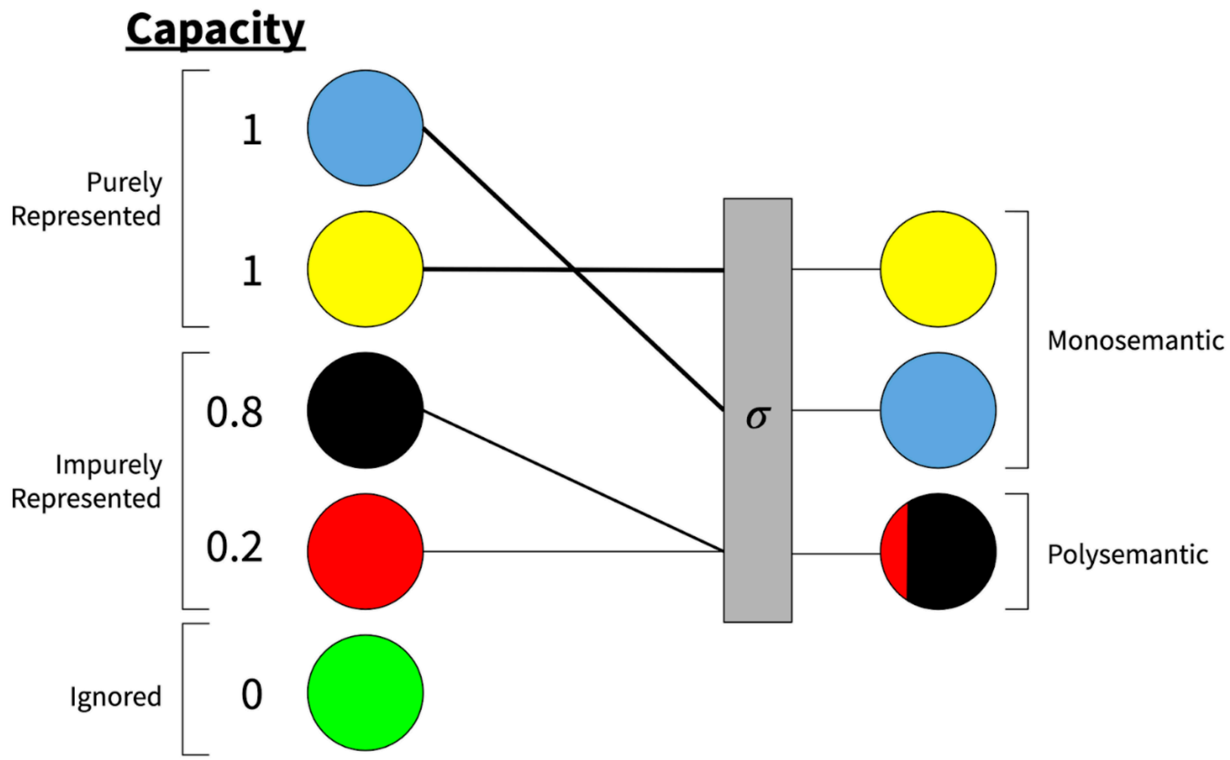
In the **high sparsity** regime, models put all features in superposition, and continue packing more. Note that at this point we begin to see positive interference and negative biases. We'll talk about this more later.

[NeuronCapacity]

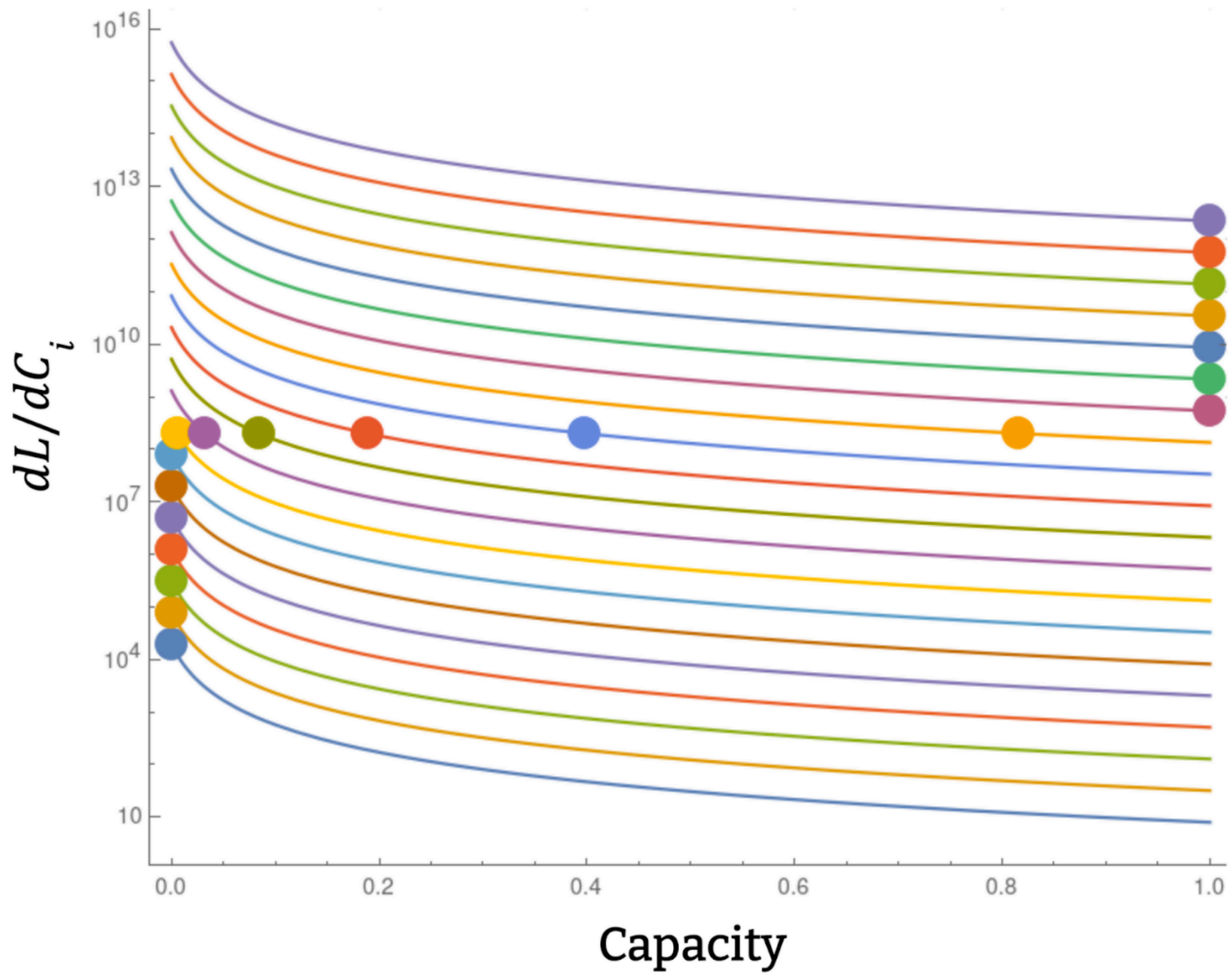
Tangent - Back to Neuron Capacity

- We can think of this Toy Model's neurons as having fixed capacity to store features

- Lets say we have a loss L and each neuron or feature x_i has a capacity C_i between 0 and 1 for how activated it is

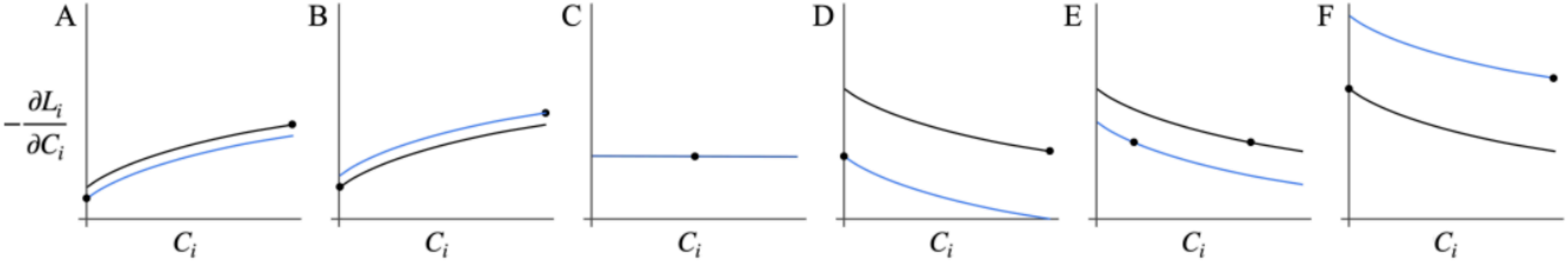


In the case where we have many features with diminishing marginal returns, capacity will in general be allocated like this:



[NeuronCapacity]

Neuron Capacity



- These graphs show a variety of different possible marginal benefit curves.

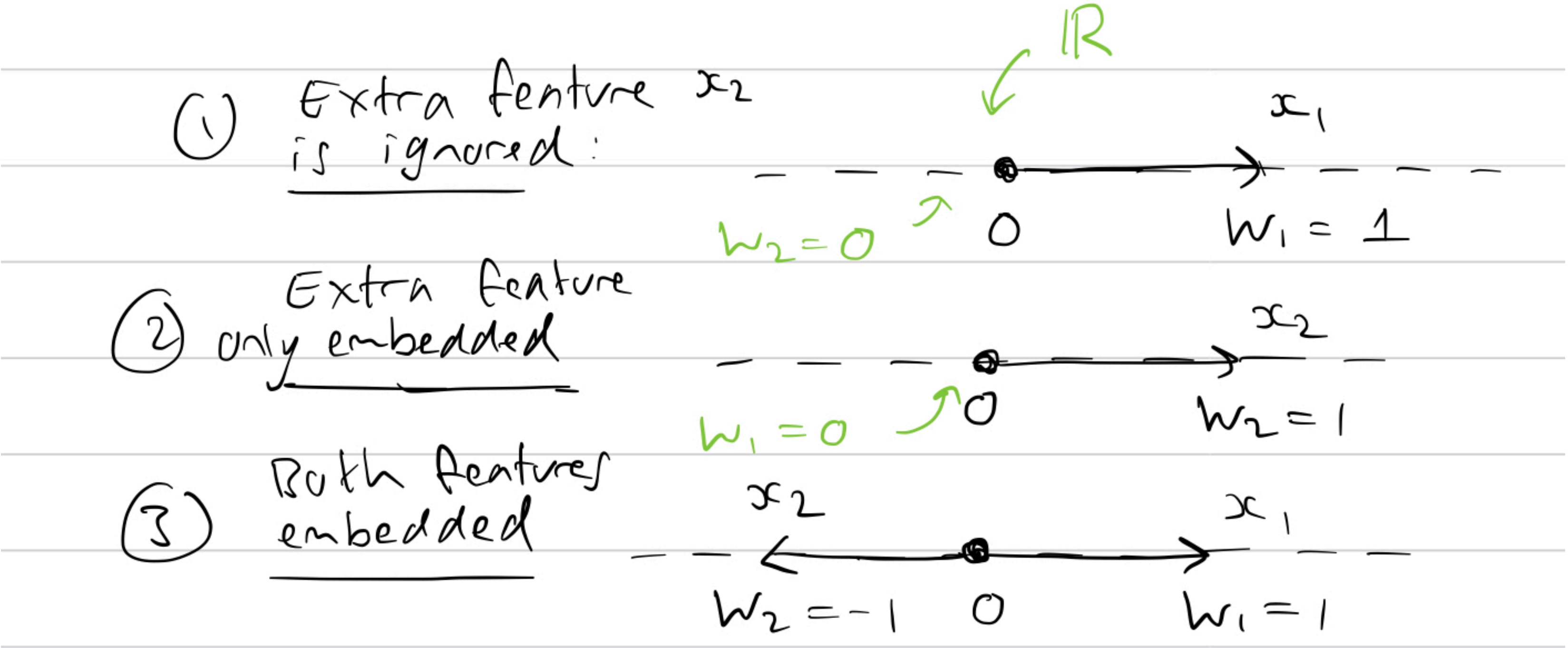
- In A and B, the marginal returns are increasing–the more you allocate capacity to a feature, the more strongly you want to allocate more capacity to it.

- In C, the marginal returns are constant (and in this graph they happen to be equal for the two features, but there’s no reason why constant marginal returns imply equal returns in general).

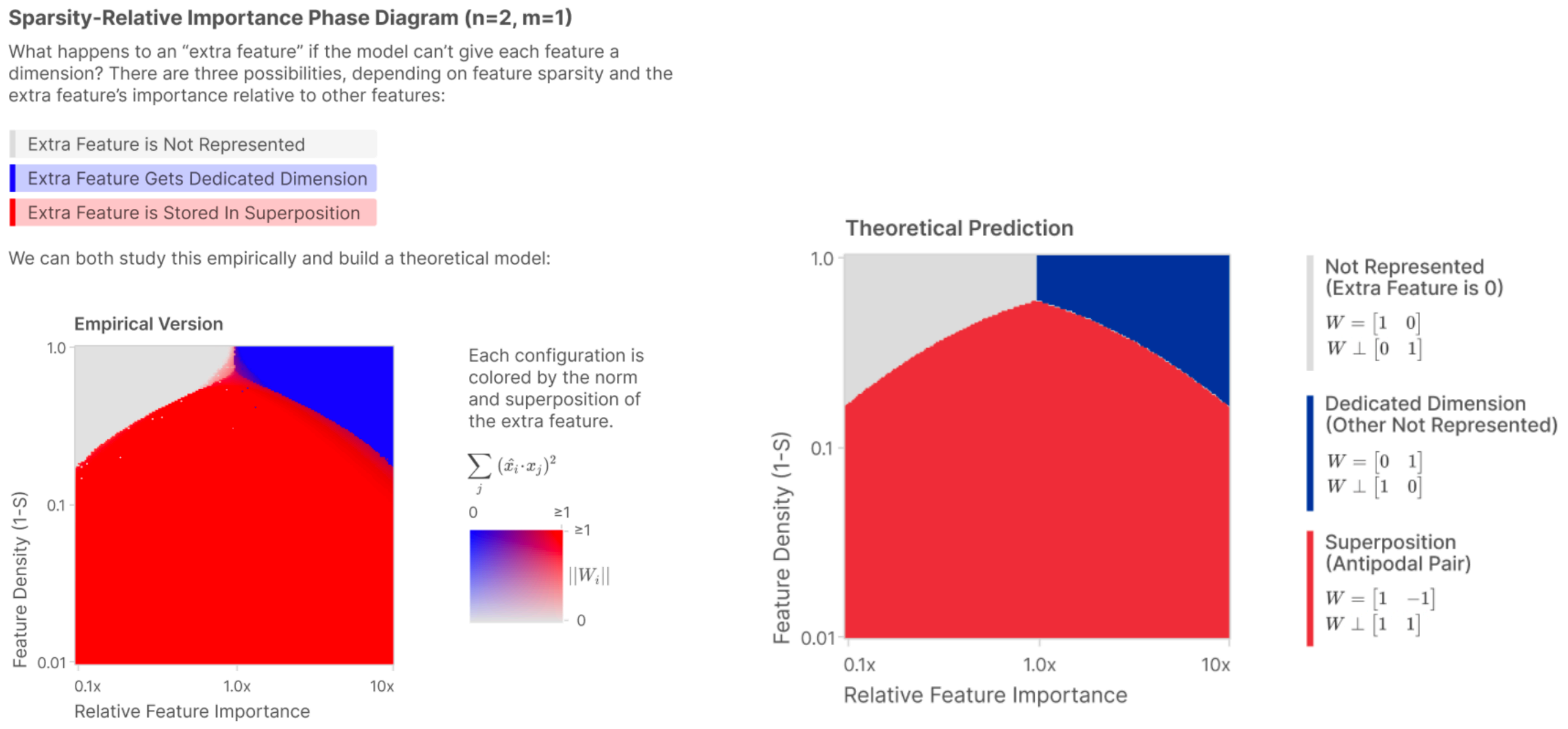
- In D, E, and F, there are diminishing marginal returns.

Phase Transition in Embedding Features

- We will now try to cram 2 features into 1 dimension.
- Let $x = (x_1, x_2) \in \mathbb{R}^2$ and $W = [w_1 \quad w_2]$.
- Feature x_1 will have importance 1 and x_2 importance I
- We will have 3 cases for embedding the features



Phase Transition in feature embedding



Sparsity-Relative Importance Phase Diagram (n=2, m=1)

What happens to an “extra feature” if the model can’t give each feature a dimension? There are three possibilities, depending on feature sparsity and the extra feature’s importance relative to other features:

- Extra Feature is Not Represented
- Extra Feature Gets Dedicated Dimension
- Extra Feature is Stored In Superposition

We can both study this empirically and build a theoretical model:

Each configuration is colored by the norm and superposition of the extra feature.

[ToyModels]

The Geometry of Superposition

- We've seen that superposition can allow a model to represent extra features, and that the number of extra features increases as we increase sparsity.

- We will see that features seem to organise themselves into geometric structures such as pentagons and digons

- There's a good chance it's at least partly due to to the toy model we're investigating. But it seems worth investigating because if anything about this generalise to real models, and turns out to be a great test bed for SLT tools

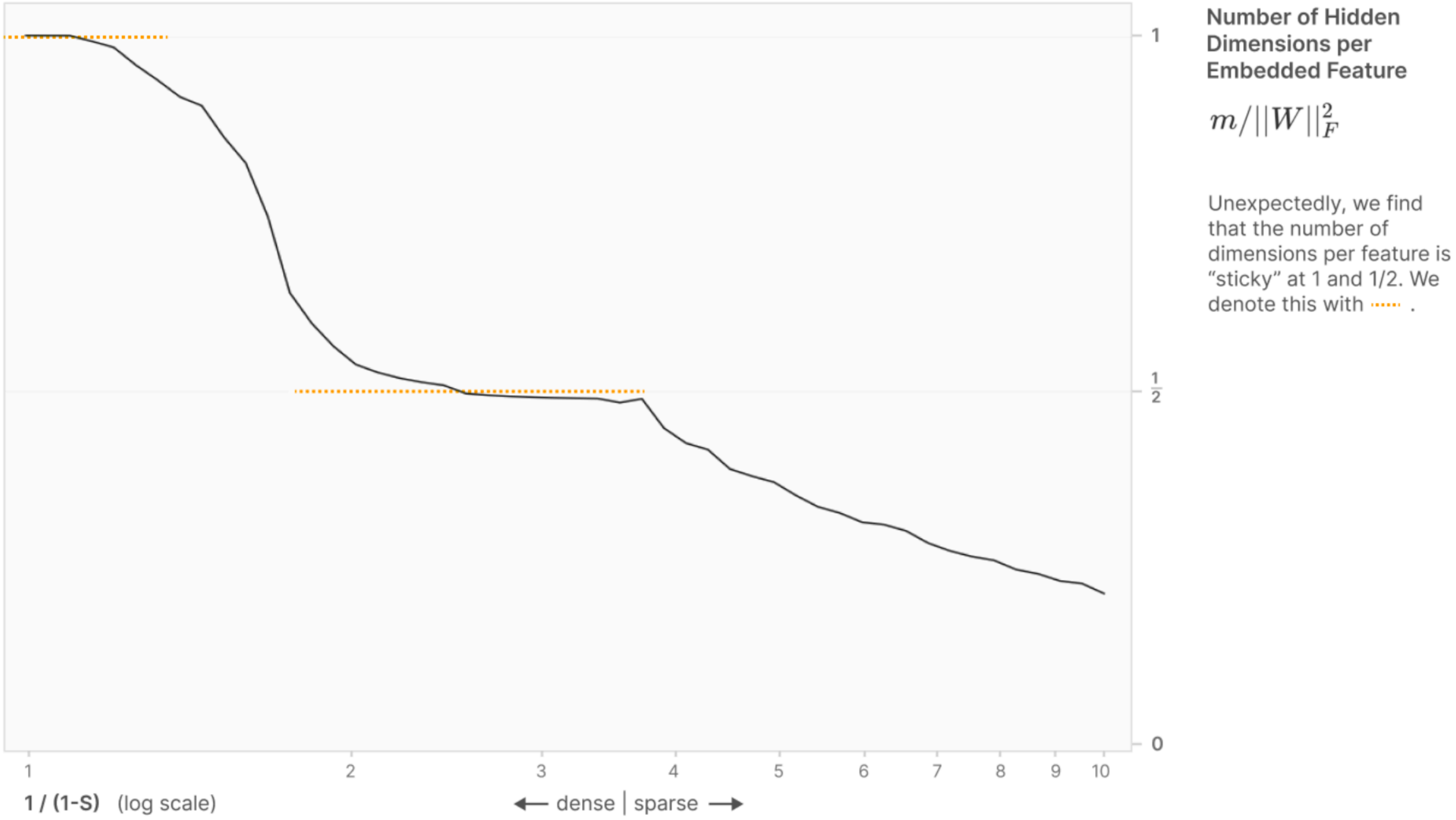
- How can we measure the number of features a model has learned? A natural idea is to look at the Frobenius Norm $||W||_F^2 = \sum_{i=1}^{n} ||W_i||^2$ where W_i is a column of W.

- Counts learned features since $||W_i||^2 \approx 1$ if a feature is learned and $||W_i||^2 \approx 0$ if it is not

[ToyModels]

Dimensions per features

- We can then plot dimensions per feature as $D^* = m/||W||_F^2 = m/\sum_{i=1}^{n} ||W_i||^2$



Number of Hidden Dimensions per Embedded Feature

$m/||W||_F^2$

Unexpectedly, we find that the number of dimensions per feature is "sticky" at 1 and 1/2. We denote this with ┈┈ .

[ToyModels]

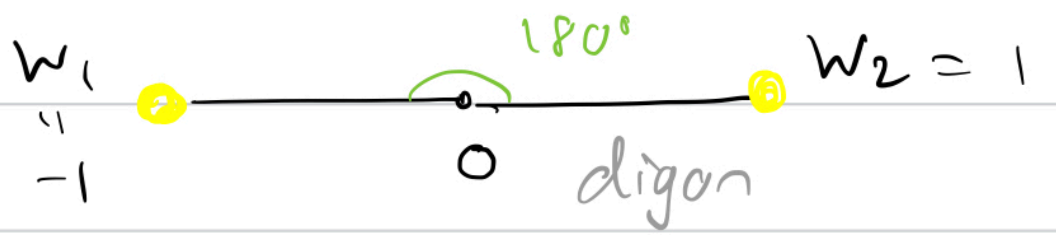
Toy Models of Superposition - Dimensionality

- How can we measure how many dimensions a given feature is embedded in?
- For this use the quantity **dimensionality,** a feature x_i will have dimensionality

$$D_i = \frac{||W_i||^2}{\sum_{j=1}^{n} (\hat{W}_i \cdot W_j)^2}$$

- The numerator represents how much a feature is represented, while the denominator is how many features share the dimension it's embedded in

2 features in 1-Dimension, $D_1 = \frac{1}{1^2+1^2} = 1/2$

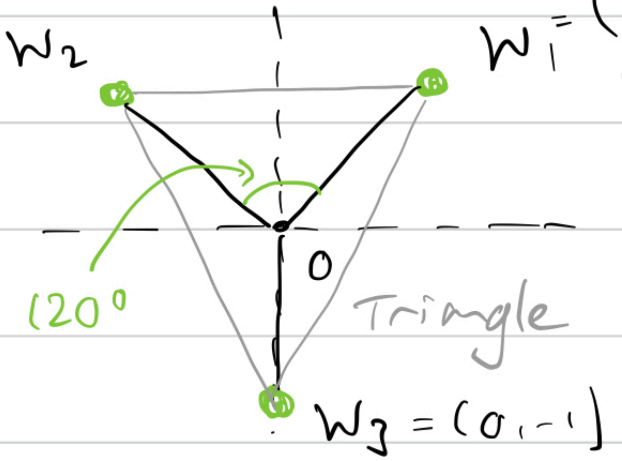


More Dimensionality Examples

$$D_i = \frac{||W_i||^2}{\sum_{j=1}^{n} (\hat{W}_i \cdot W_j)^2}$$

- Even though we only computed D_1 in all these, the dimensionality of each feature inside the polytope will be the same
- Why don't we embed these polytopes into a much larger dimensional space?

3 features in 2 dimensions

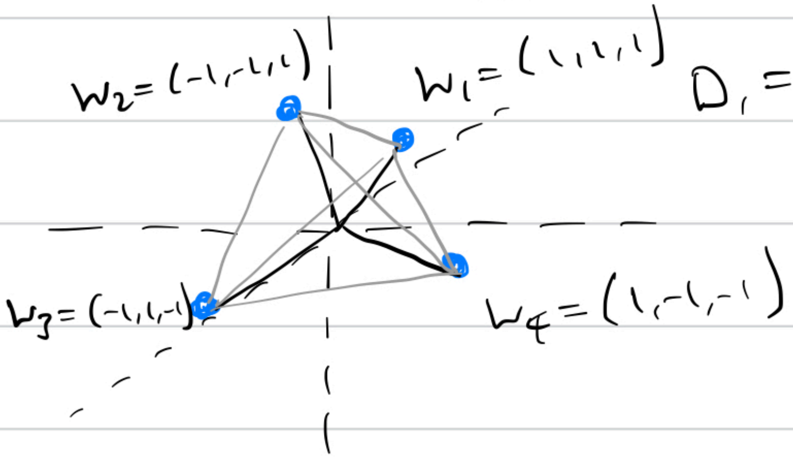


$W_1 = \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$

$$D_1 = \frac{1}{1^2 + (\frac{1}{2})^2 + (\frac{1}{2})^2} = \frac{1}{3/2} = \frac{2}{3}$$

4 features in 3 dimensions

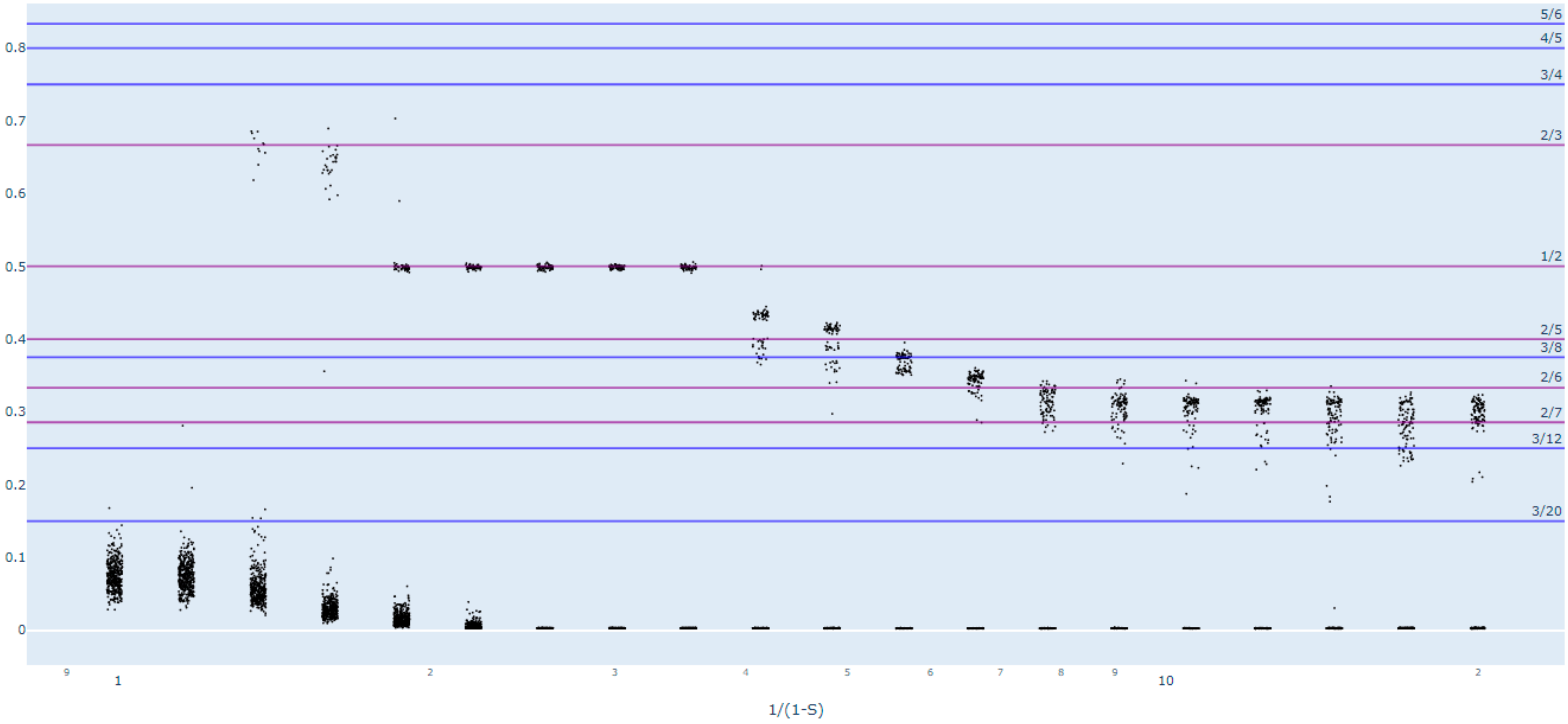
Tetrahedron



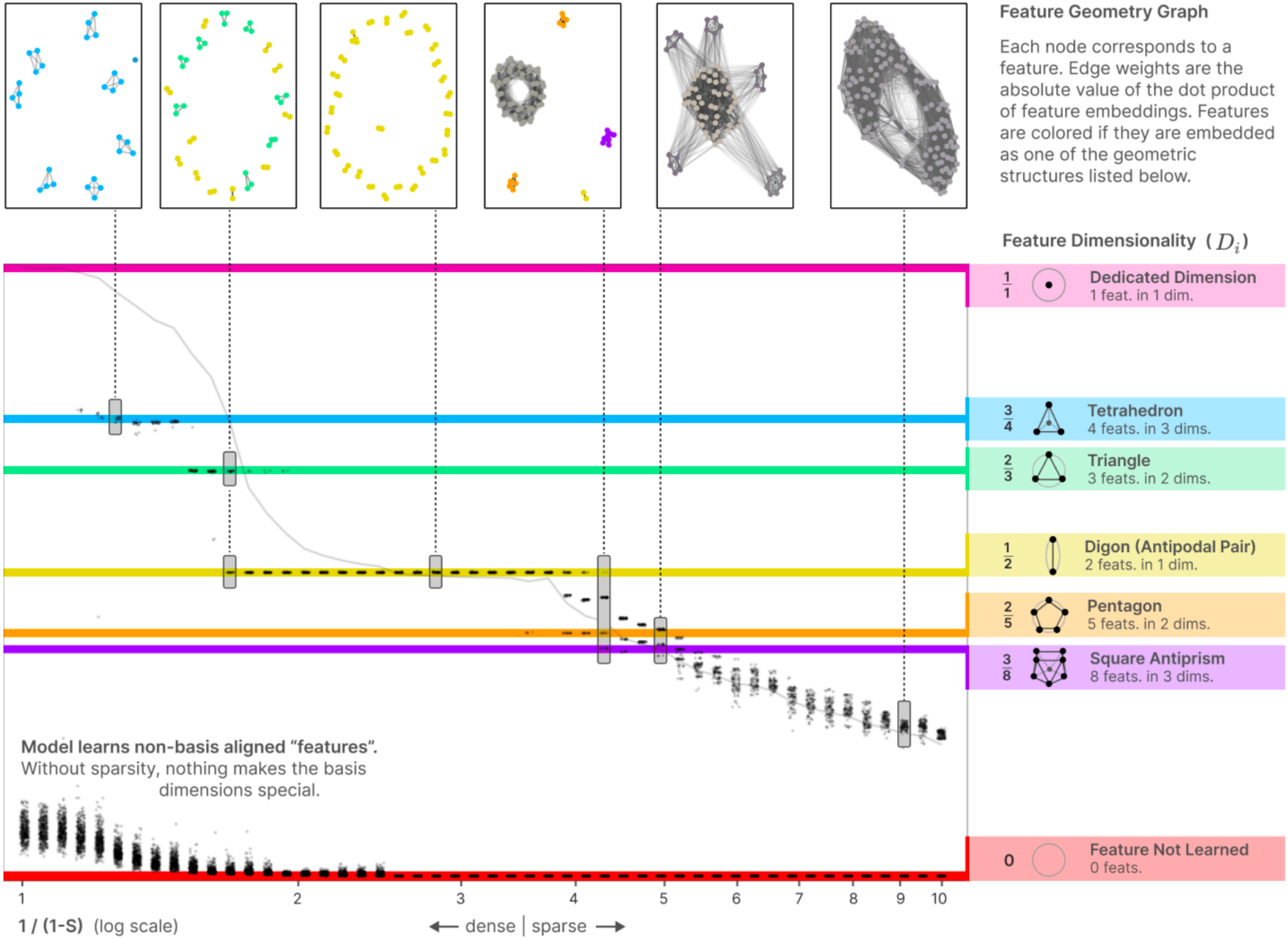
$$D_1 = \frac{1}{1 + (\frac{1}{3})^2 + (\frac{1}{3})^2 + (\frac{1}{3})^2} = 3/4$$

* divide each coordinate by $\sqrt{3}$ to normalise so $W_i = 1$.

Dimensionality Plot of 400 features into 30 dimensions



Phase Transitions in Polytopes



Feature Geometry Graph

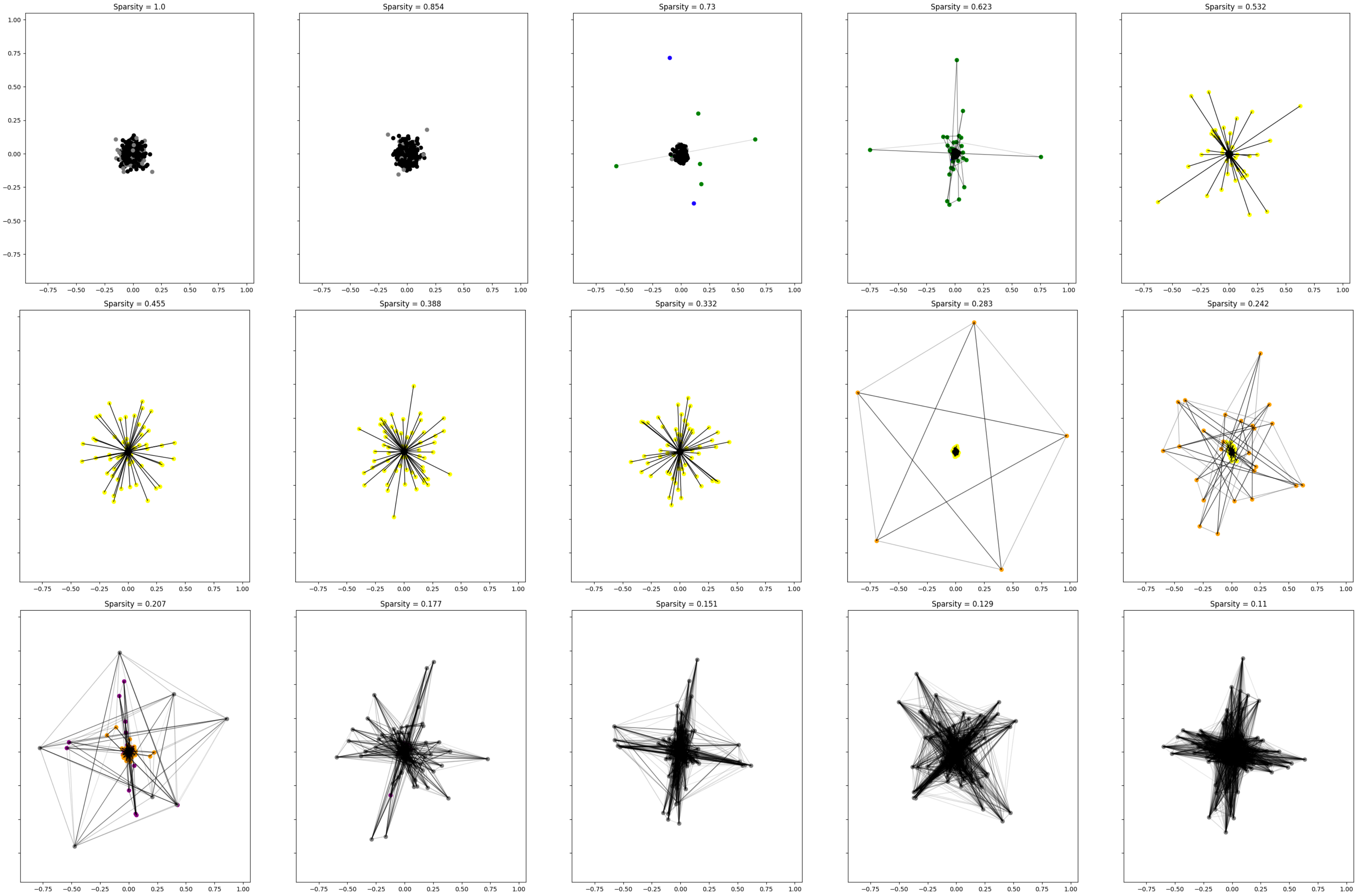
Each node corresponds to a feature. Edge weights are the absolute value of the dot product of feature embeddings. Features are colored if they are embedded as one of the geometric structures listed below.

Feature Dimensionality (D_i)

$\frac{1}{1}$	⊙	**Dedicated Dimension** 1 feat. in 1 dim.
$\frac{3}{4}$		**Tetrahedron** 4 feats. in 3 dims.
$\frac{2}{3}$		**Triangle** 3 feats. in 2 dims.
$\frac{1}{2}$		**Digon (Antipodal Pair)** 2 feats. in 1 dim.
$\frac{2}{5}$		**Pentagon** 5 feats. in 2 dims.
$\frac{3}{8}$		**Square Antiprism** 8 feats. in 3 dims.
0	○	**Feature Not Learned** 0 feats.

Model learns non-basis aligned "features". Without sparsity, nothing makes the basis dimensions special.

[ToyModels]



Sparsity = 1.0

Sparsity = 0.854

Sparsity = 0.73

Sparsity = 0.623

Sparsity = 0.532

Sparsity = 0.455

Sparsity = 0.388

Sparsity = 0.332

Sparsity = 0.283

Sparsity = 0.242

Sparsity = 0.207

Sparsity = 0.177

Sparsity = 0.151

Sparsity = 0.129

Sparsity = 0.11

- Note Sparsity here means 1- s

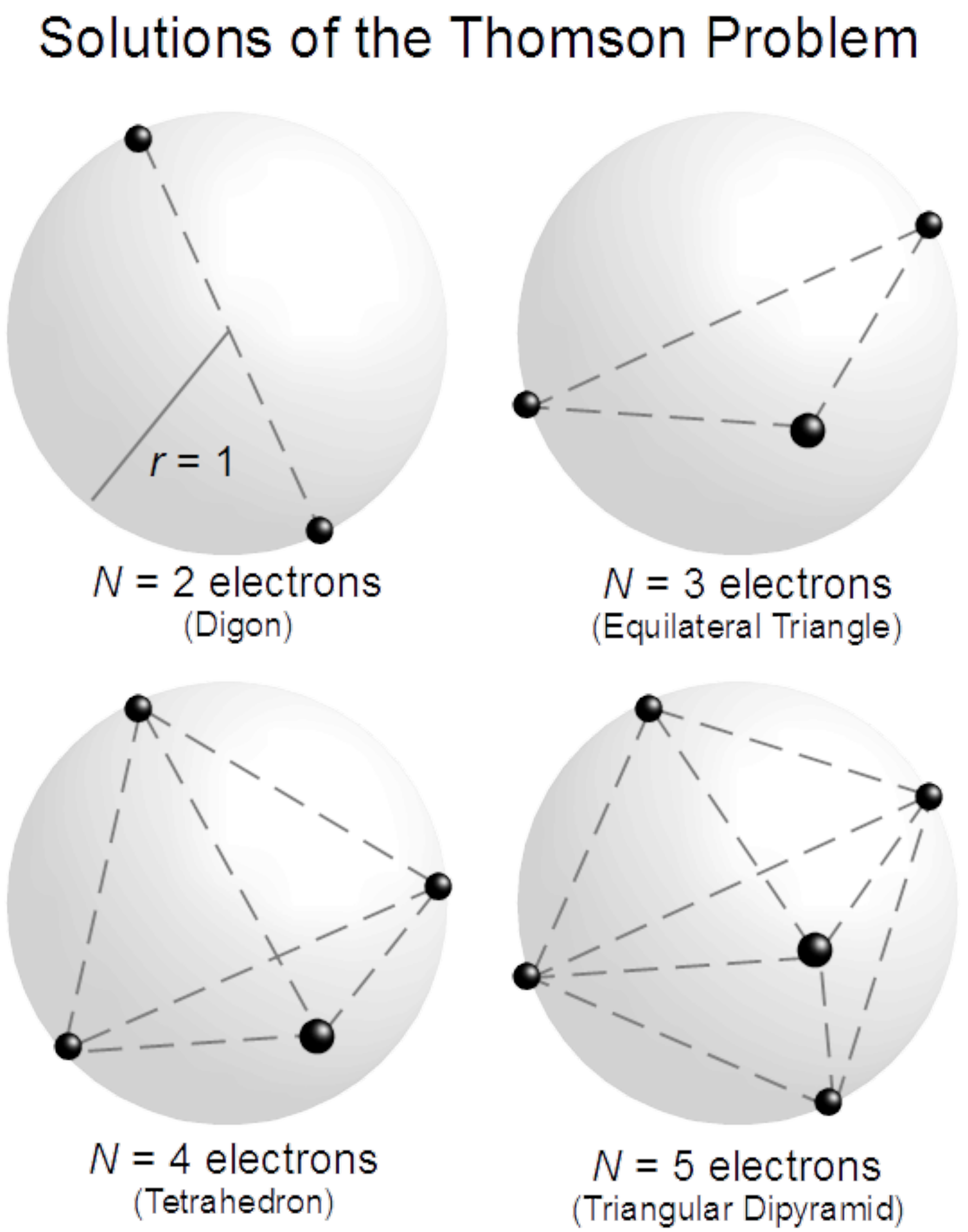
[ToyModels]

Superposition - Why these shapes?

- The reason the model likes antipodal pairs for many sparsity values is because if at most one of either feature in the pair is present, but doesn’t if both are.

- Say both features are sparse and they’re present 25% of the time, then this is good 50% of the time and bad 6.25% of the time

- Similar to the Thompson Problem (but don’t read too much into this)



Solutions of the Thomson Problem

N = 2 electrons
(Digon)

N = 3 electrons
(Equilateral Triangle)

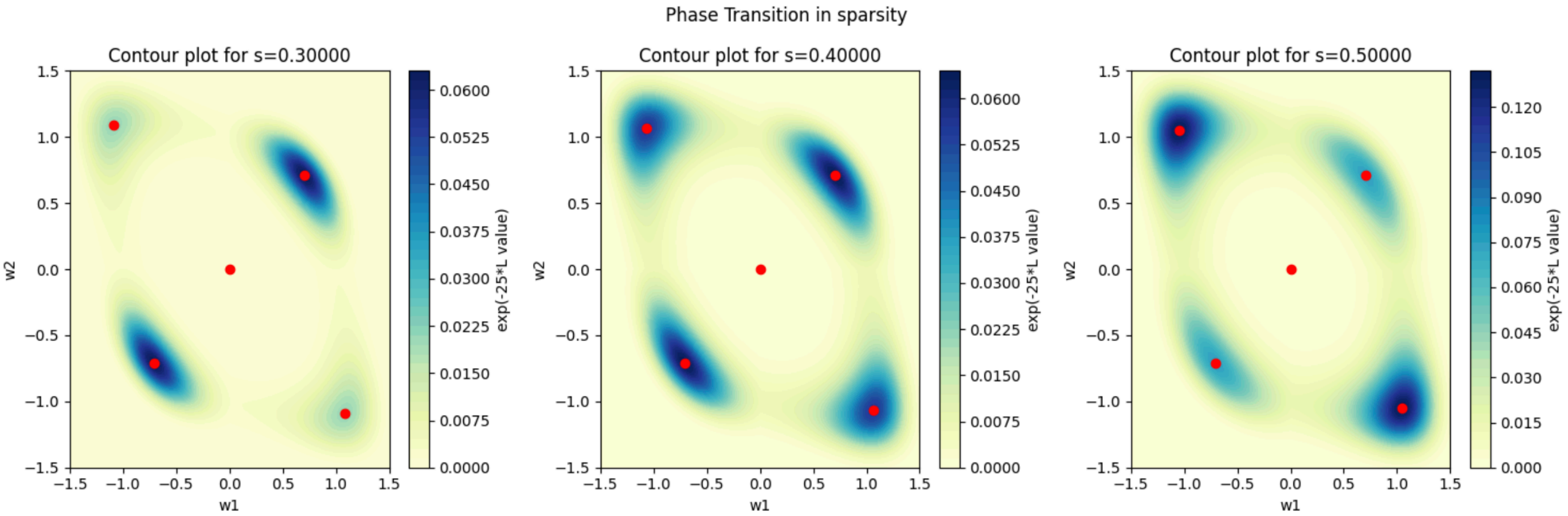
N = 4 electrons
(Tetrahedron)

N = 5 electrons
(Triangular Dipyramid)

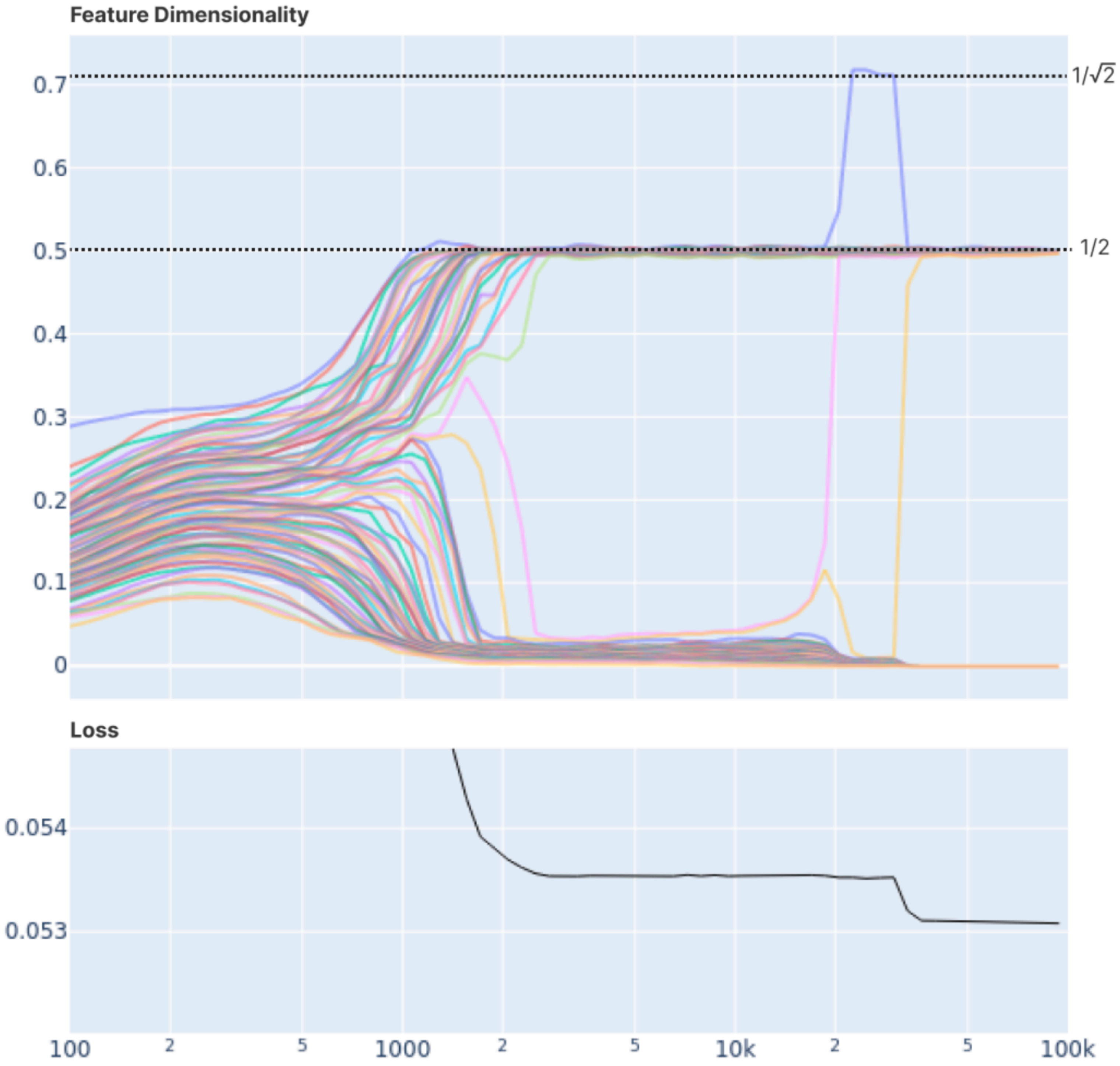
Toy Models of Superposition - n2m1 in Biasless Model

- It is possible to exactly solve the expected loss for 2 features in 1 dimension with a biasless model

$$K(w) = \int_{\mathbb{R}^2} q(x,s)\,||x - \textbf{ReLU}(W^T W x)||^2\,dx$$

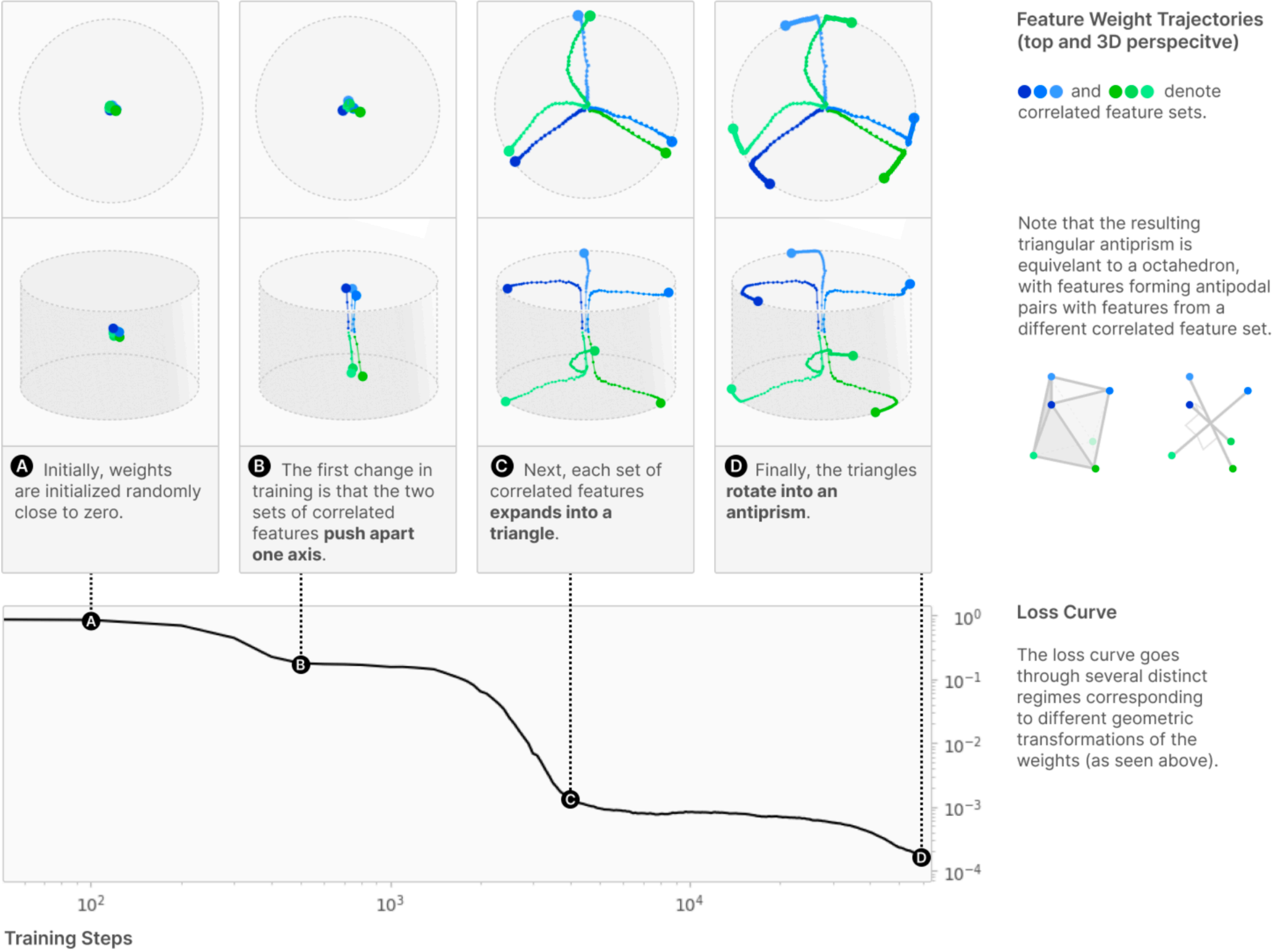


Dimensionality Phase Transitions in Training



[ToyModels]

Phase Transition in Training



A Initially, weights are initialized randomly close to zero.

B The first change in training is that the two sets of correlated features **push apart one axis**.

C Next, each set of correlated features **expands into a triangle**.

D Finally, the triangles **rotate into an antiprism**.

Feature Weight Trajectories (top and 3D perspecitve)

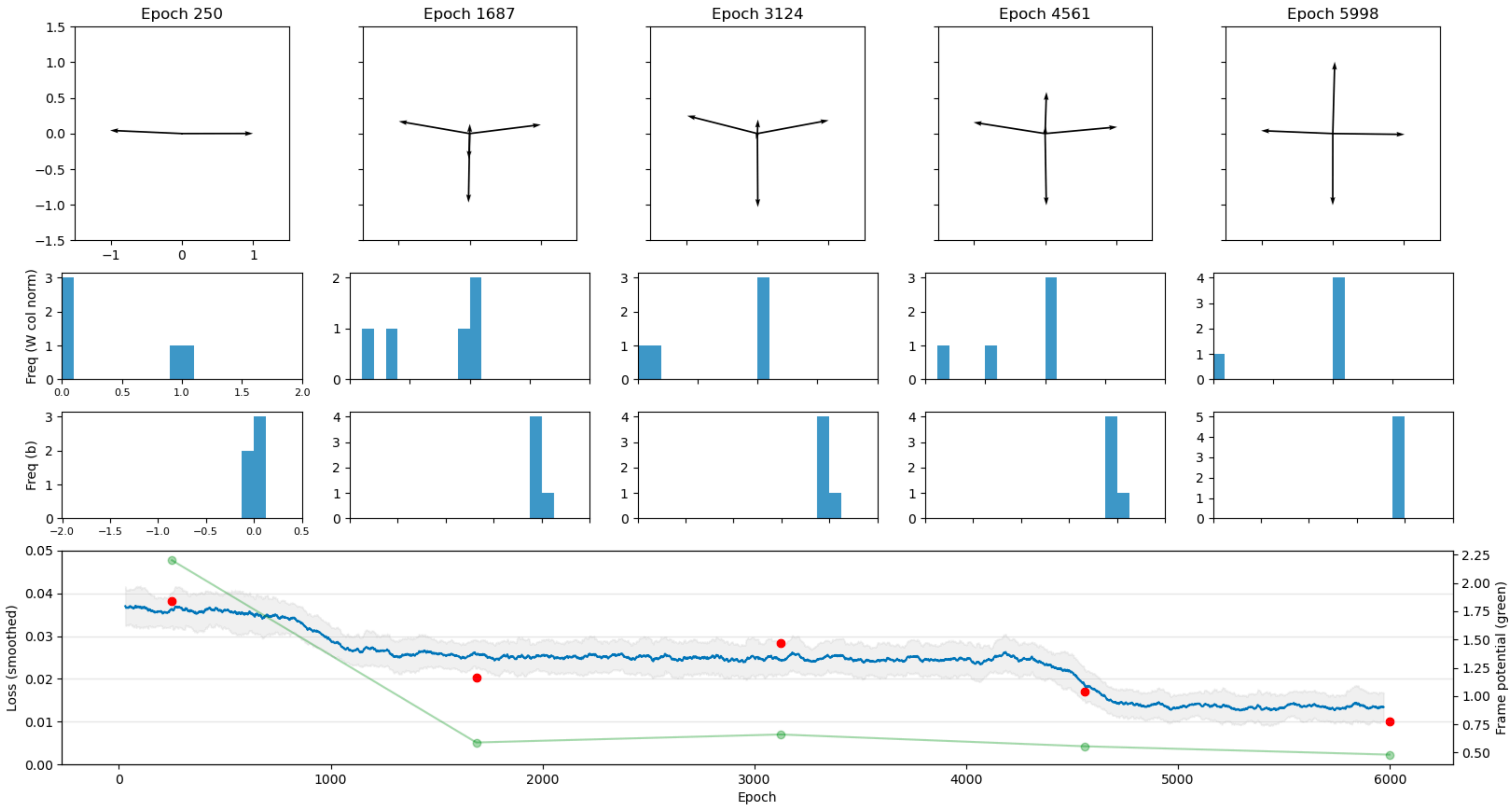
●●● and ●●● denote correlated feature sets.

Note that the resulting triangular antiprism is equivelant to a octahedron, with features forming antipodal pairs with features from a different correlated feature set.

Loss Curve

The loss curve goes through several distinct regimes corresponding to different geometric transformations of the weights (as seen above).

Phase Transition in Loss

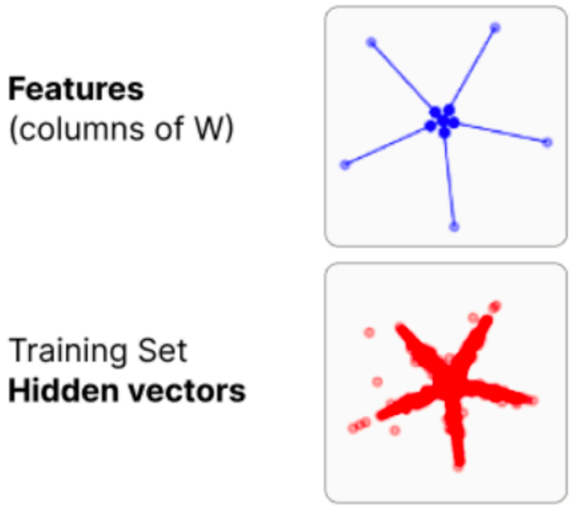


Toy models (n=5, m=2, batch_size=128, lr_init=0.001, lr_final=0.00051, decay_factor=0.8, decay_interval=2874, final_loss=0.01827)

[ToyModels2]

Bonus Content: Training Data v Features

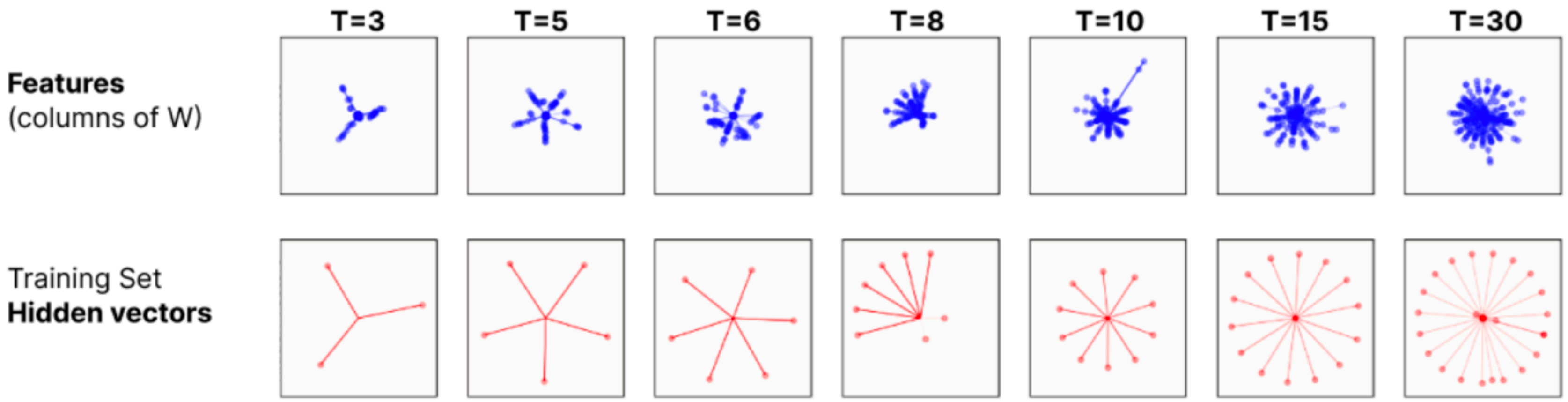
- What if instead we have a finite training data set T
- Here the sparsity $s = 0.999$ is fixed and features are uniformly distribution U$[0,1]$ but rescaled so $||x||^2 = 1$
- Keeping track of training sets T with $X \in \mathbb{R}^{n\times T}$ representing training data
- Hidden vectors are $h_i = WX_i$
- We see data points, rather than features are being represented as direction
- This makes sense when $T < n$ since it's easier to represent the T samples as oppose n features in the $m-$ dimensional space



As we expect from the original toy models paper, the feature embedding vectors (W_i) for sparse features correspond to points on a pentagon. Five points are represented, and the rest are mapped to approximately zero.

In addition to looking at the features, we can ask how individual data points are represented in the hidden space. For this model, we see that most data points only activate zero or one of the five features. The outliers are rare cases where >2 features are activated.

$$L = \frac{1}{T}\sum_x \sum_i I_i(x_i - x_i')^2$$



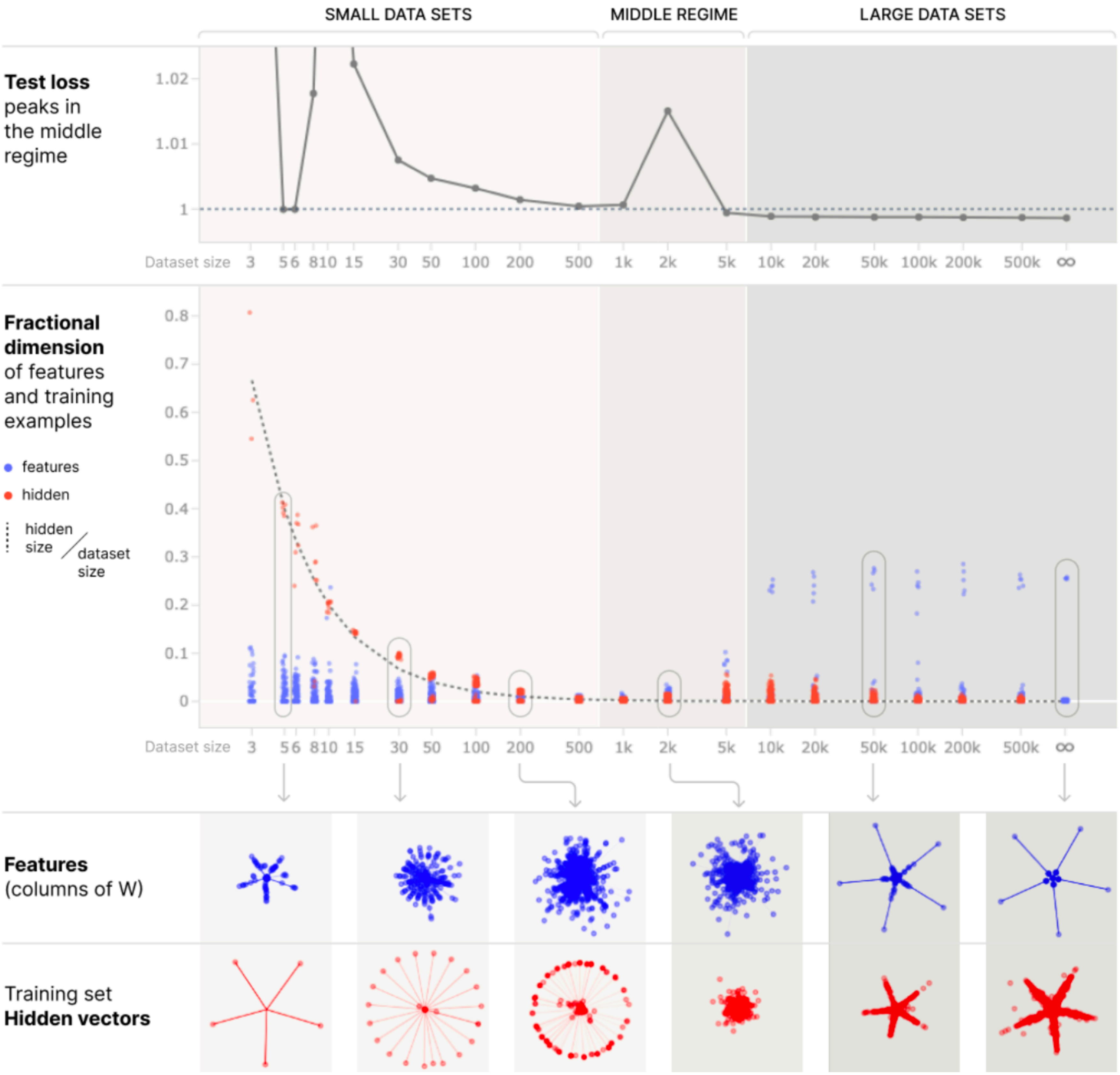
[ToyModels2]

Memorisation v Generalisation

- Here we also introduce a dimensionality in the hidden training examples

$$D_{X_i} = \frac{||h_i||^2}{\sum_{j=1}^{n} (\hat{h}_i \cdot h_j)^2}$$

- Where $h_i = WX_i$ and we can now visualise the geometry of features and data points as we vary the dataset size T
- We expect each example to have dimensionality m/T (here $m = 2$)



Test loss peaks in the middle regime

Fractional dimension of features and training examples

Features (columns of W)

Training set **Hidden vectors**

Key Takeaways

- Mechanistic interpretability uses features and circuits to help us understand what’s going on inside these models
- Superposition appears to be one of the largest hurdles Mechanistic Interpretability has to struggle with, as what makes it so powerful at reducing loss also makes it hard to interpret
- It appears that sparse features align themselves into geometric polytopes
- A promising tool would be the ability to *identify and enumerate over all features*. The ability to have a universal quantifier over the fundamental units of neural network computation is a significant step towards saying that certain types of circuits don't exist.
- We plan to use the Toy Models given to us by Anthropic as a test bed for SLT tools and to better understand what superposition is