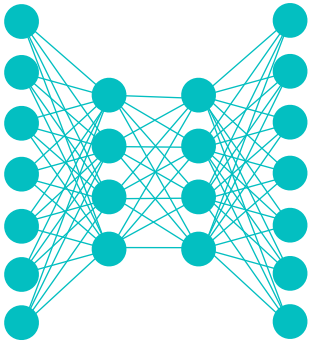
Lecture Notes for

Neural Networks and Machine Learning

More CNN Circuits

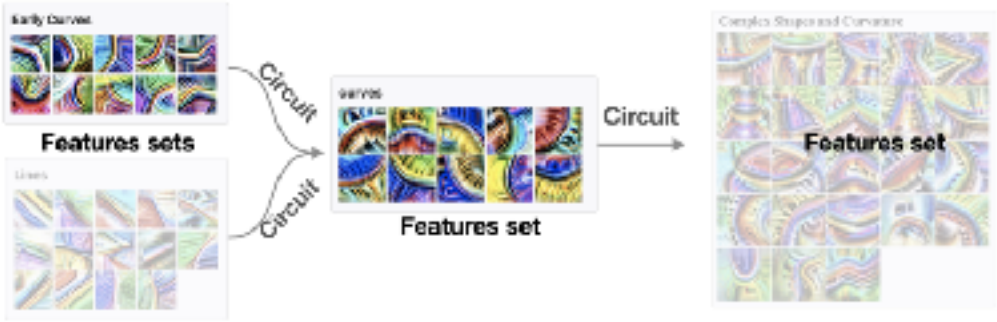


Logistics and Agenda

- Logistics
  - Grading Update
- Agenda
  - Last Time: Visualization + Circuits in CNNs
  - More Circuits
  - Continued Town Hall
  - Branch Specialization and Universality
  - Next Time:
    - Student Paper Presentation
    - Fully Convolutional Networks

Last Time

- *Features are connected by weights, forming circuits*
- *"All neurons in our network are formed from linear combinations of neurons in the previous layer, followed by ReLU. If we can understand the features in both layers, shouldn't we also be able to understand the connections between them?"*
- *"Once you understand what features they're connecting together... You can literally read meaningful algorithms off of the weights."*

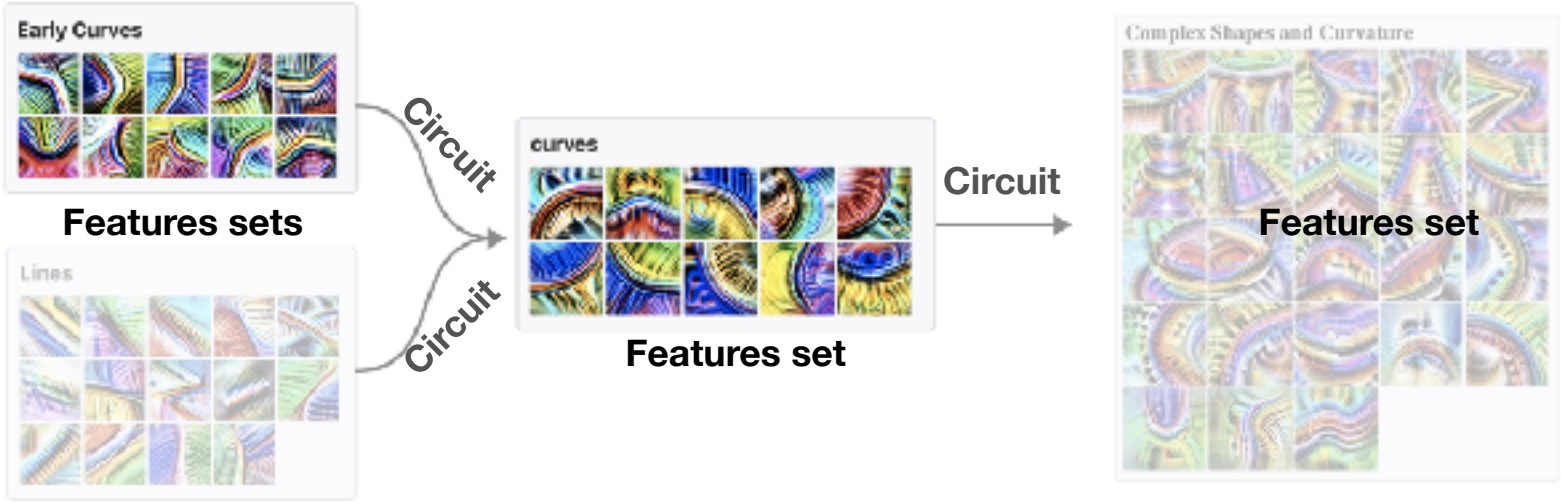


Neuron 4b:409

Dataset examples for neuron 4b:409

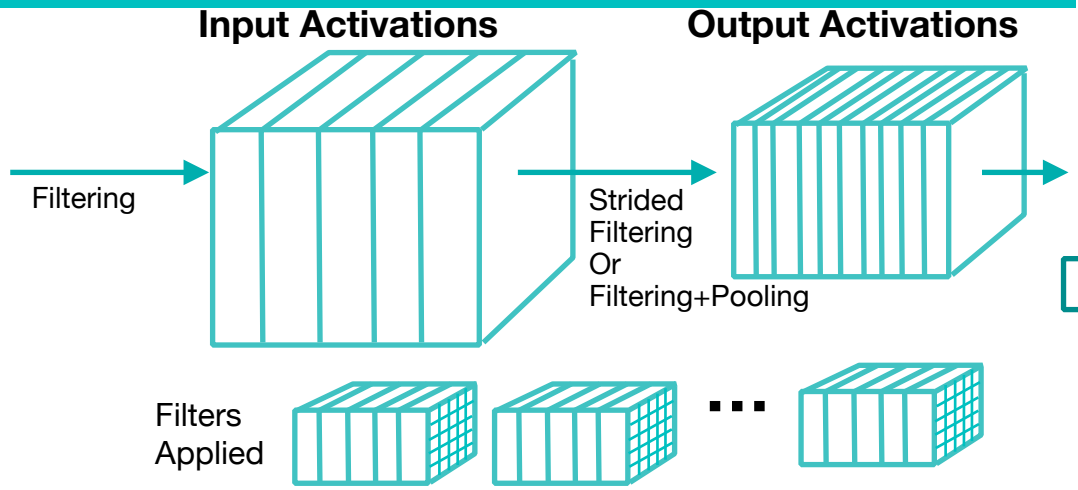
From Features to Circuits

- *Features are connected by weights, forming circuits*
- *"All neurons in our network are formed from linear combinations of neurons in the previous layer, followed by ReLU. If we can understand the features in both layers, shouldn't we also be able to understand the connections between them?"*
- *"Once you understand what features they're connecting together… You can literally read meaningful algorithms off of the weights."*



https://microscope.openai.com/models/inceptionv1/

Review of CNN Structure



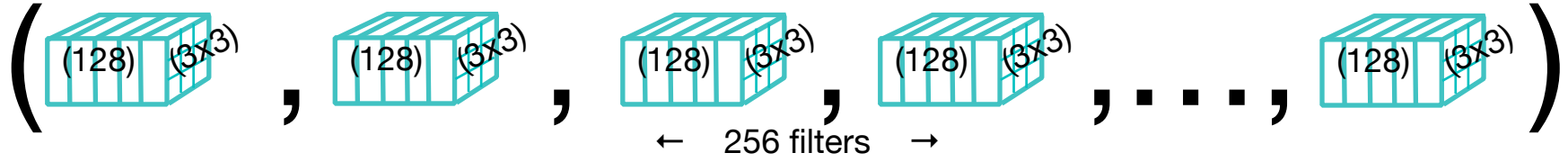
Model: "vgg16"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, None, None, 3)]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0

...

```
keras_layer = model.get_layer('block3_conv1')
layer_output = keras_layer.output
weights_list = keras_layer.get_weights() # list
filters = weights_list[0]
biases = weights_list[1]
```

```
block4_conv1 activation size is  (None, None, None, 256) (batch x H x W x filter)
block4_conv1 filters is of shape (3, 3, 128, 256) ...(k x k x channels x filters)
block4_conv1 biases is of shape (256,)
one filter in block4_conv1 is  (3, 3, 128)
```



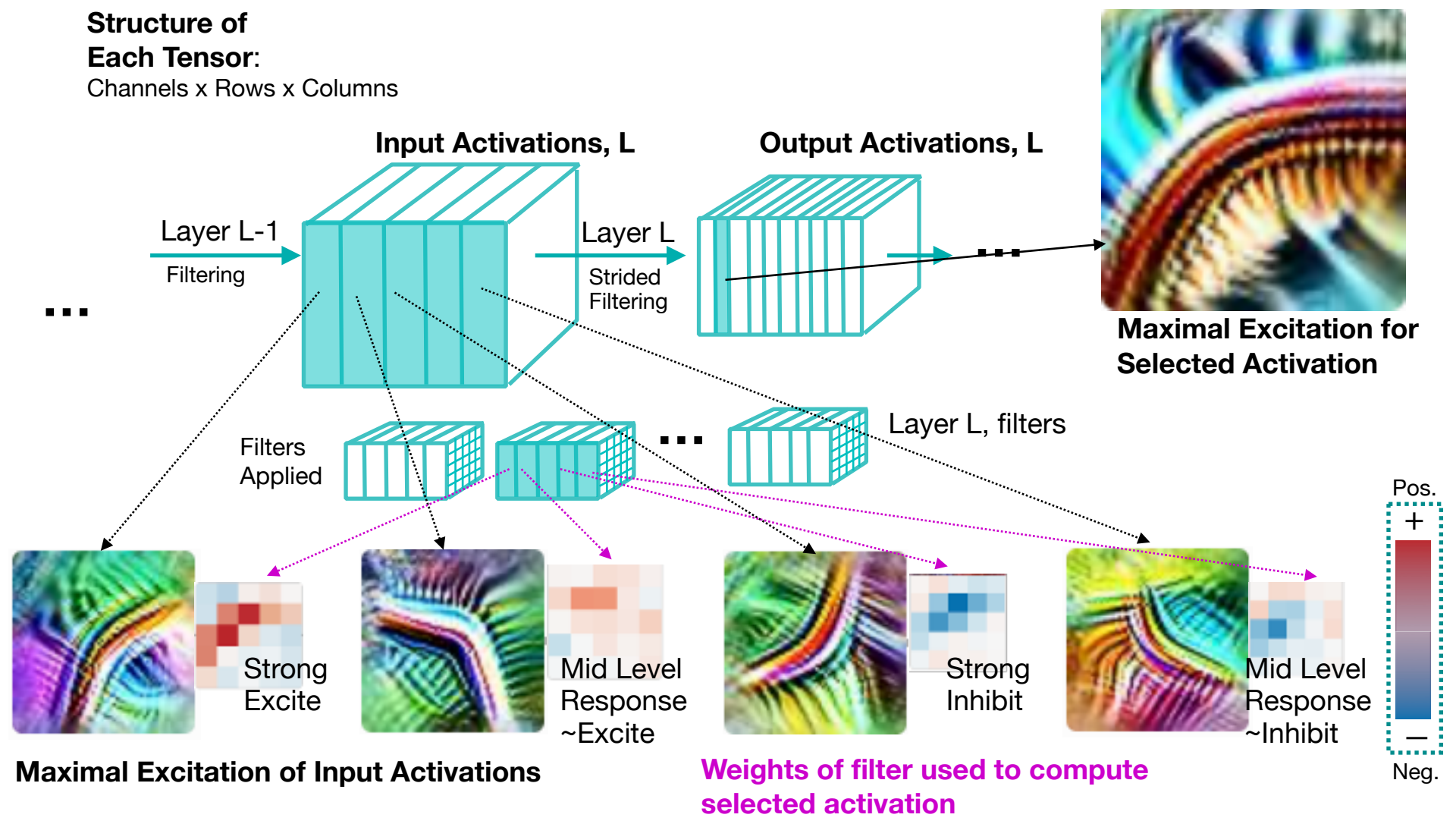
```
#  lets look at the shapes of some of the filters above
keras_layer = model.get_layer('block3_conv1')
layer_output = keras_layer.output
weights_list = keras_layer.get_weights() # list of filter, the biases
filters = weights_list[0]
biases = weights_list[1]

# print out some specifics of how the filter is saved
print('block4_conv1 activation size is ', layer_output.get_shape(), '(batch x H x W x
print('block4_conv1 filters is of shape',filters.shape, '...(k x k x channels x filter
print('block4_conv1 biases is of shape',biases.shape)

idx = 32
print('one filter in block4_conv1 is ', filters[:,:,:,idx].shape )
channel = 2
print('one channel in the the filter is', filters[:,:,channel,idx].shape)
print('The weights of that channel in the filter are:\n', filters[:,:,channel,idx])
print('The bias of the filter is:',biases[idx])
```

```
block4_conv1 activation size is  (None, None, None, 256) (batch x H x W x filter)
block4_conv1 filters is of shape (3, 3, 128, 256) ...(k x k x channels x filters)
block4_conv1 biases is of shape (256,)
one filter in block4_conv1 is  (3, 3, 128)
one channel in the the filter is (3, 3)
The weights of that channel in the filter are:
 [[-0.03330493  0.01174345  0.03184387]
 [-0.04050588 -0.02253938  0.02304637]
 [-0.00191393 -0.01501364  0.02783429]]
The bias of the filter is: 0.030420048
```

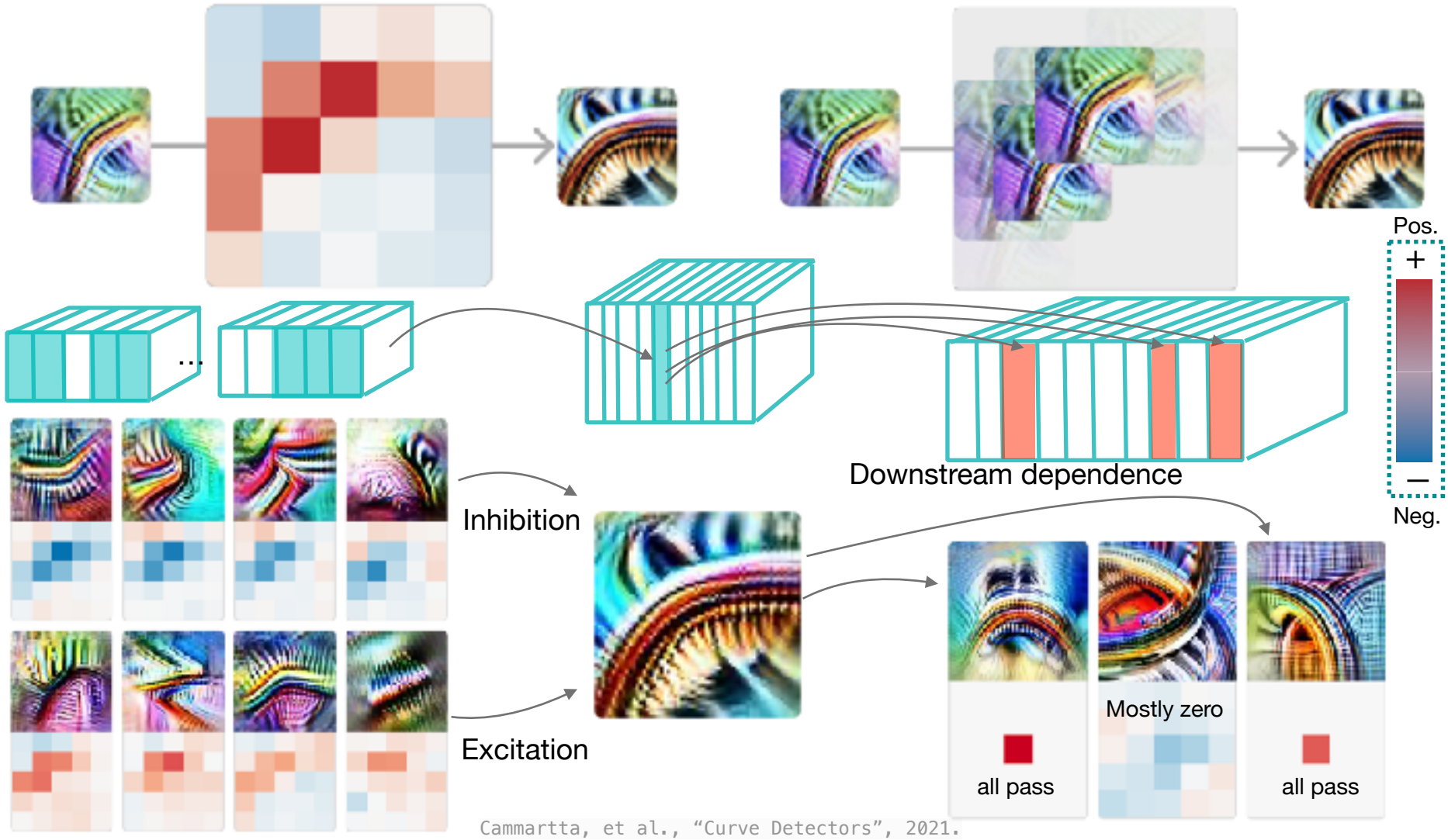

What weights comprise a circuit?



https://distill.pub/2020/circuits/curve-circuits/

Example: Circuit for Better Curve Detection

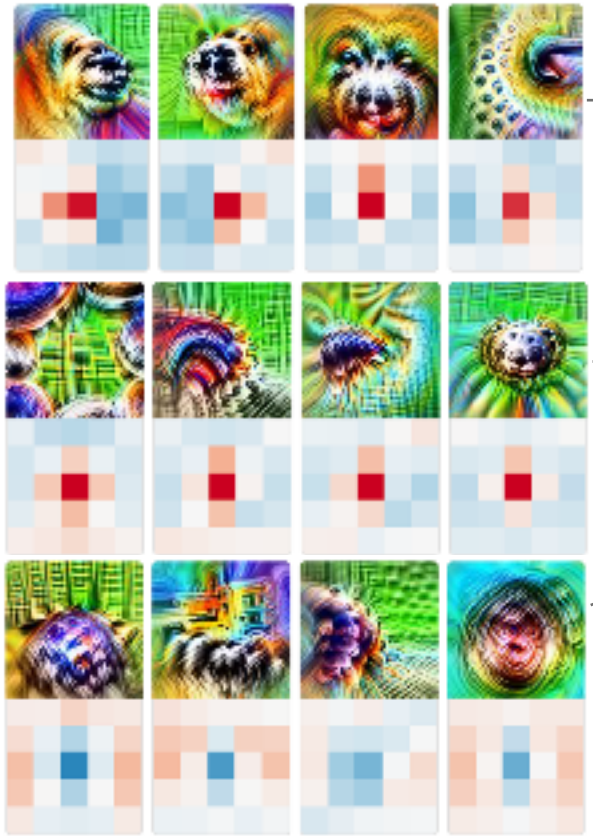
If we visualize the 5x5 Conv Filter, we can see that this becomes a Superposition of Early Curves

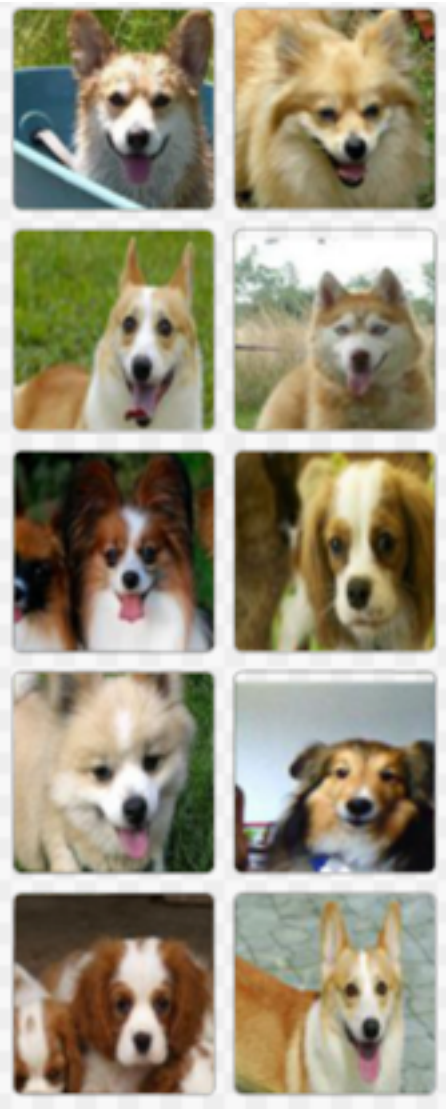


Cammartta, et al., "Curve Detectors", 2021.

Another Example: Dog head

Compact Circuit Visualization





This example is also **polysemantic** due to the "**espresso maker**" class also being excited by this…

Olah, et al., "Naturally Occurring Equivariance in NN", 2021.

Equivariant Circuits

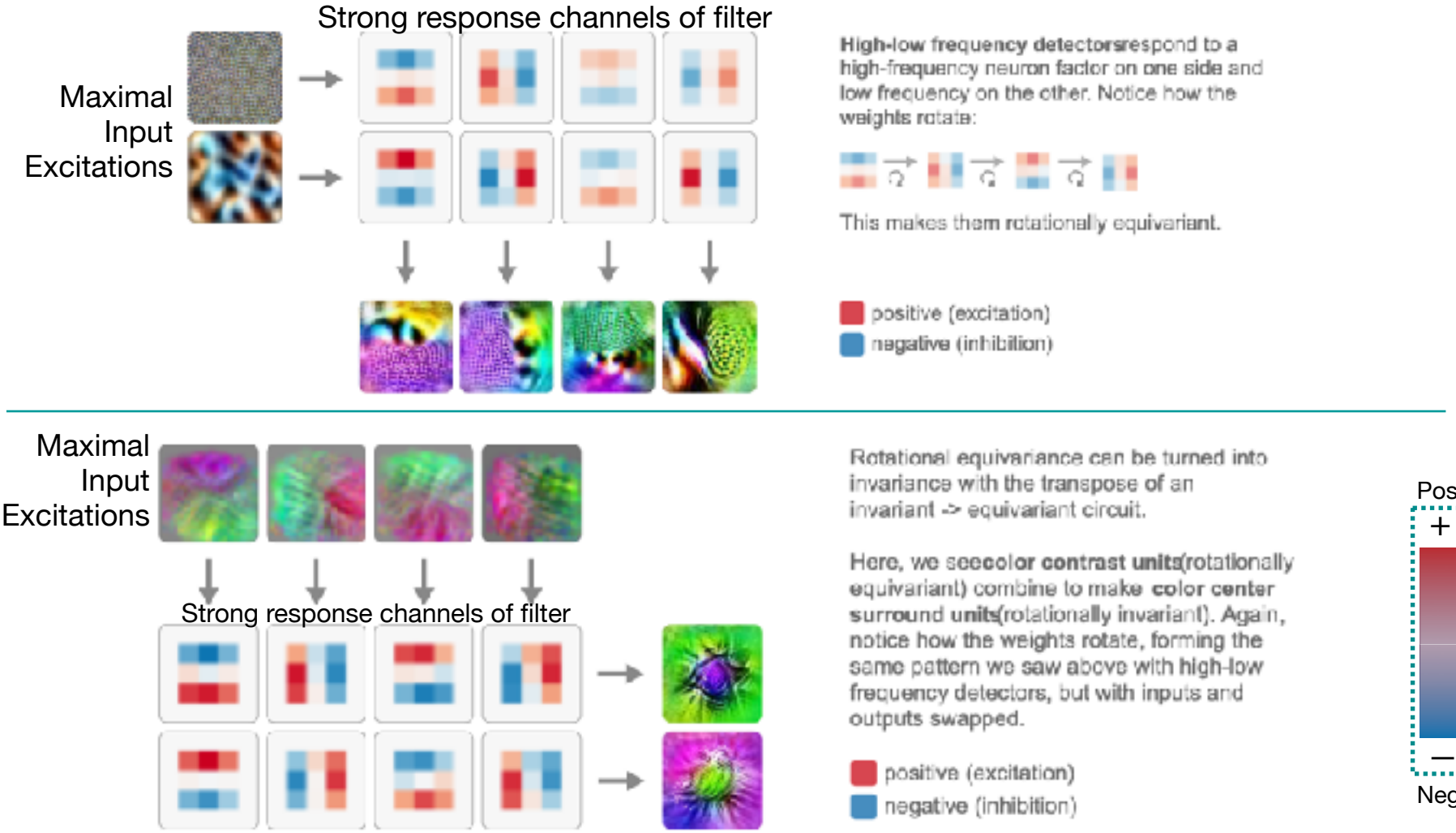
- Many features that are part of a circuit are clearly designed for rotation, hue, and other invariance

Olah, et al., "Naturally Occurring Equivariance in NN", 2021.

https://distill.pub/2020/circuits/equivariance/

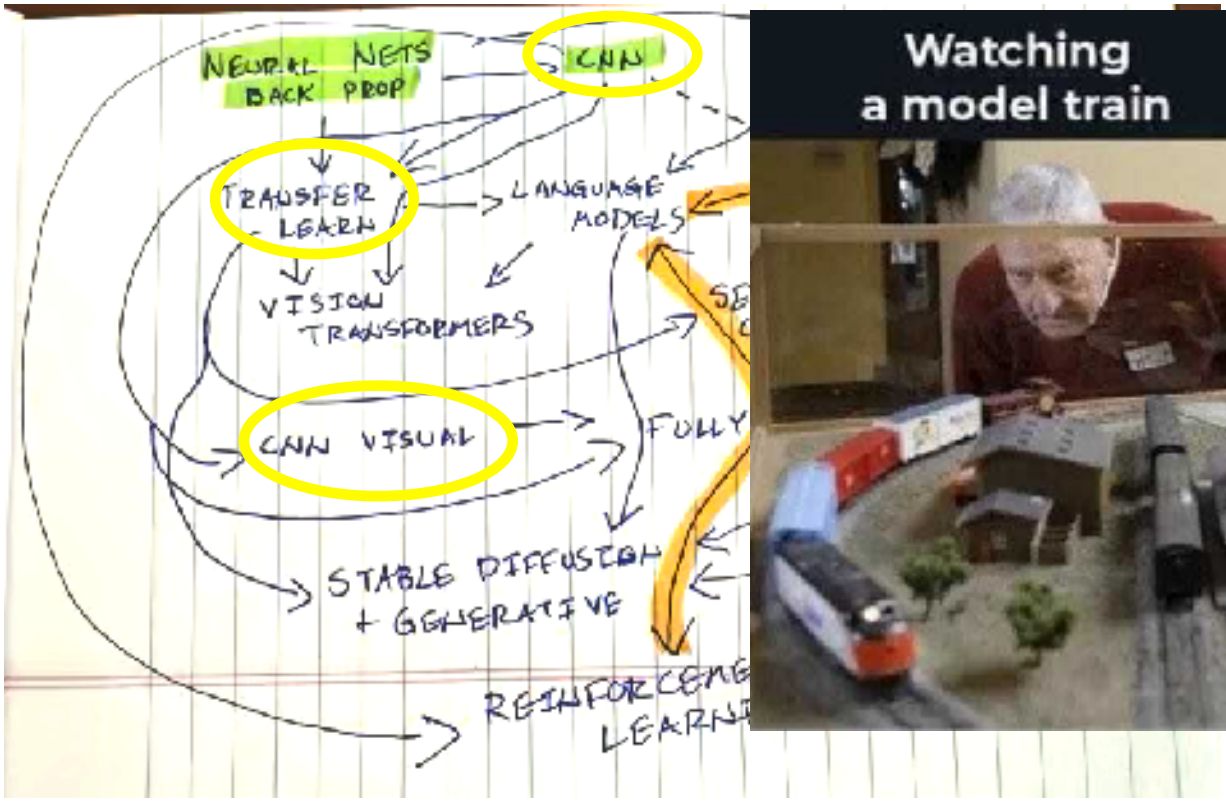
Equivariant circuits: a Motif

- Possible to reveal patterns of circuits via sets of weights



Olah, et al., "Naturally Occurring Equivariance in NN", 2021.

Circuits Town Hall



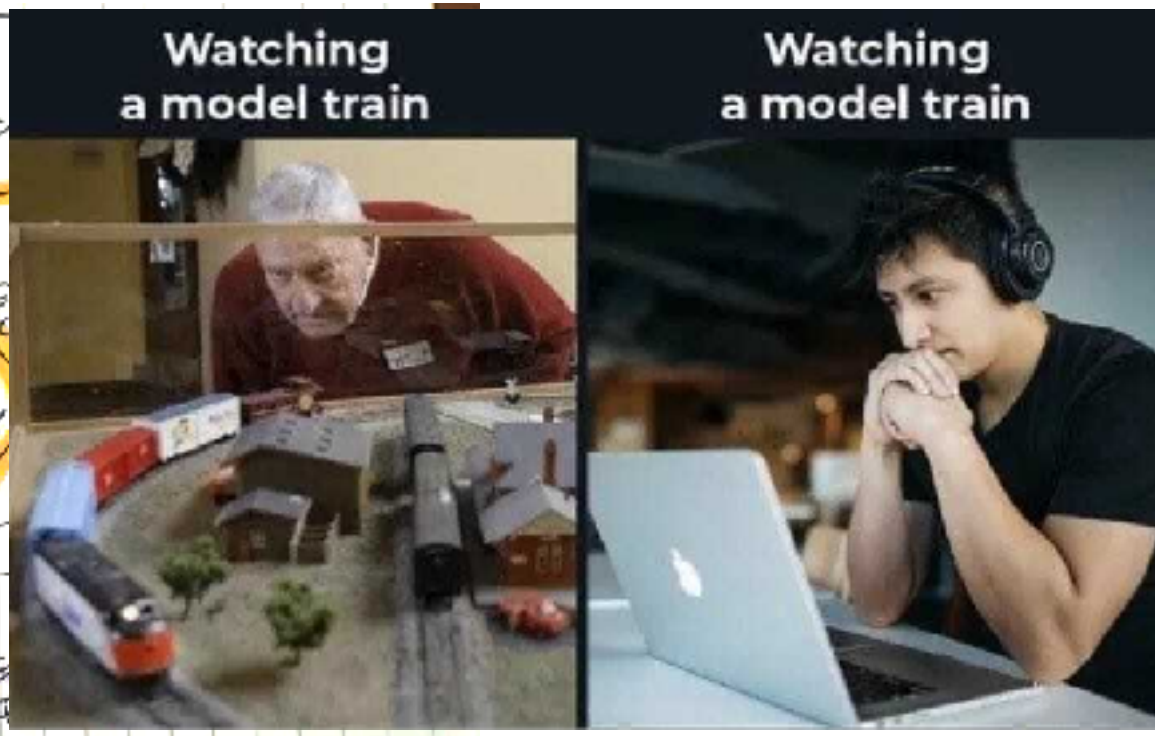


Figure for Circuits Lab

1. Choose a network and middle layer to analyze. This example shows activations from a VGG layer with 512 channels.

2. Choose Output Activation of Interest

Structure of Each Tensor:
Channels x Rows x Columns

Input Activations

Output Activations

Channels

Columns

Rows

Filtering

Strided Filtering

...

...

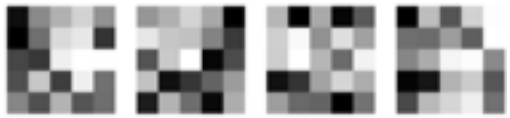
5. Look at the input activations for each channel of the multi-channel filter, see which are the most influential.

3. Find multi-channel filter that is responsible for selected activation

Multi-channel Filters

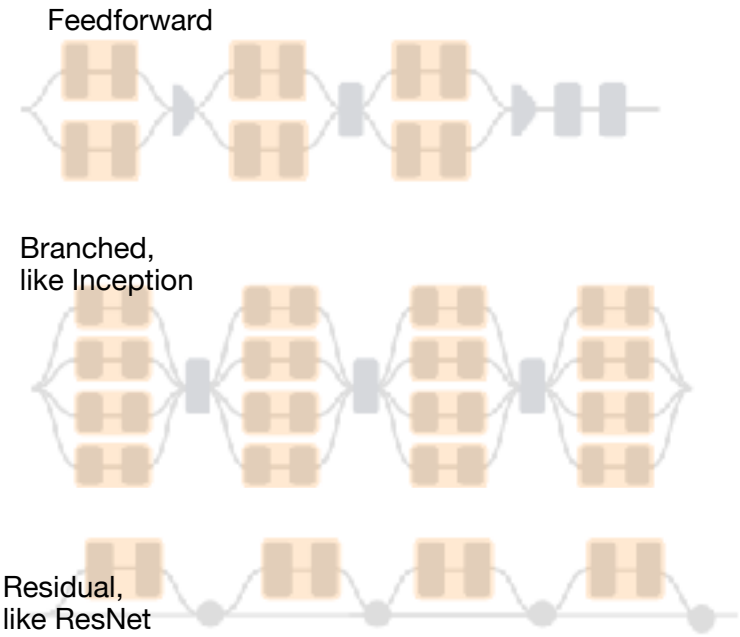
...

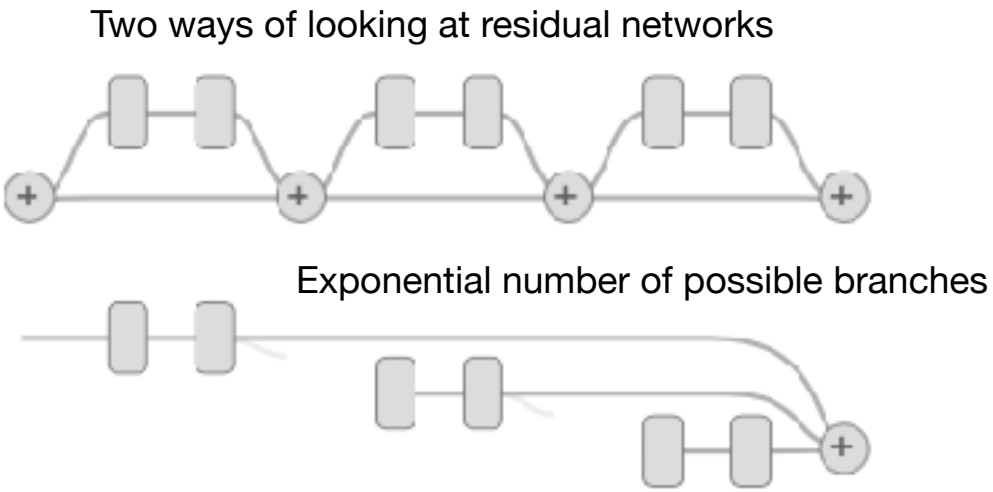
...



4. Look at each channel of the multi-channel filter. Each channel can be thought of as a filter applied to the activations in the previous layer.

Branch Specialization



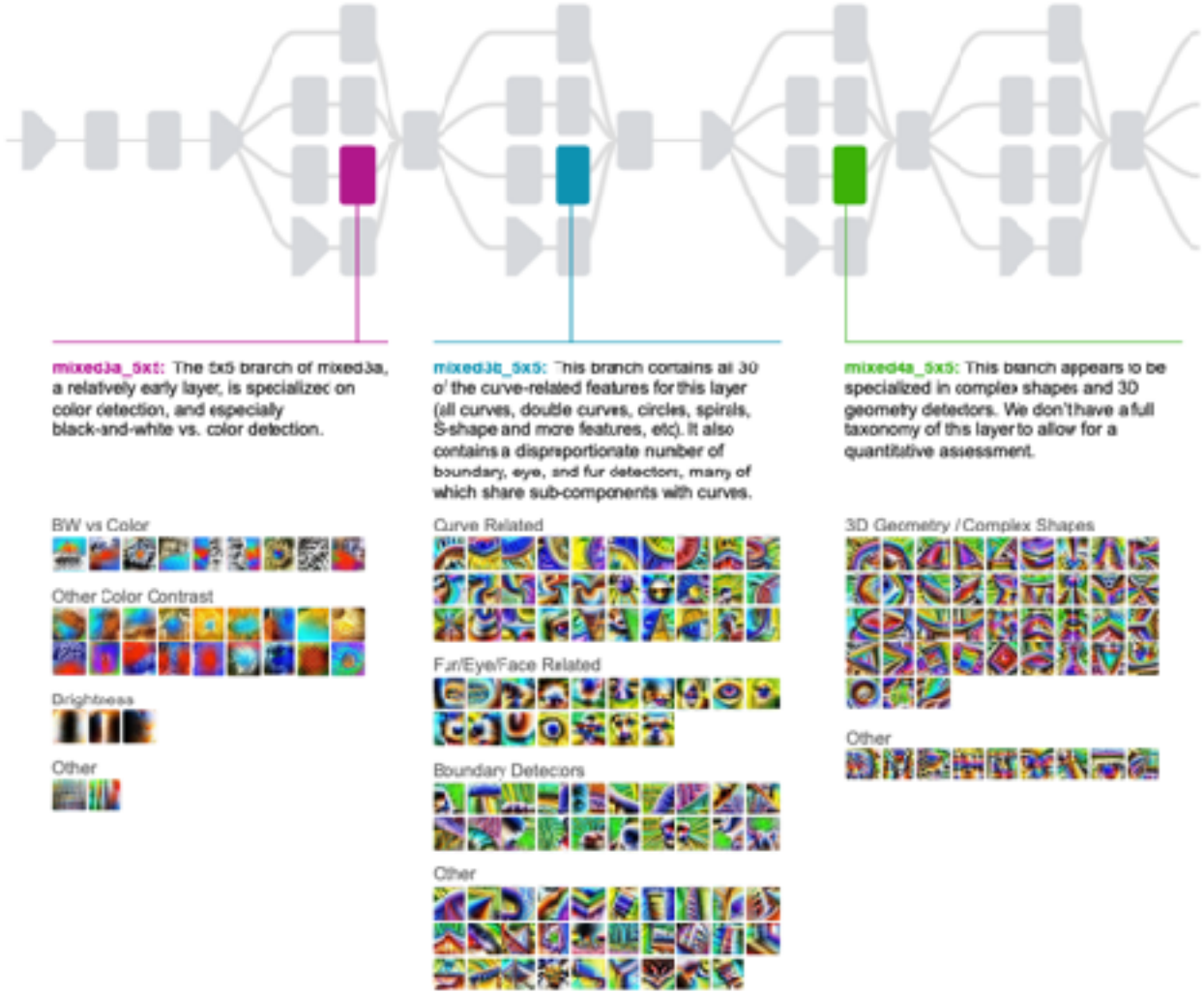


- Specialized branches are consistent across many architectures, support the idea of an interconnected graph of operations

https://distill.pub/2020/circuits/branch-specialization/

Voss, et al., "Branch Specialization", Distill, 2021.

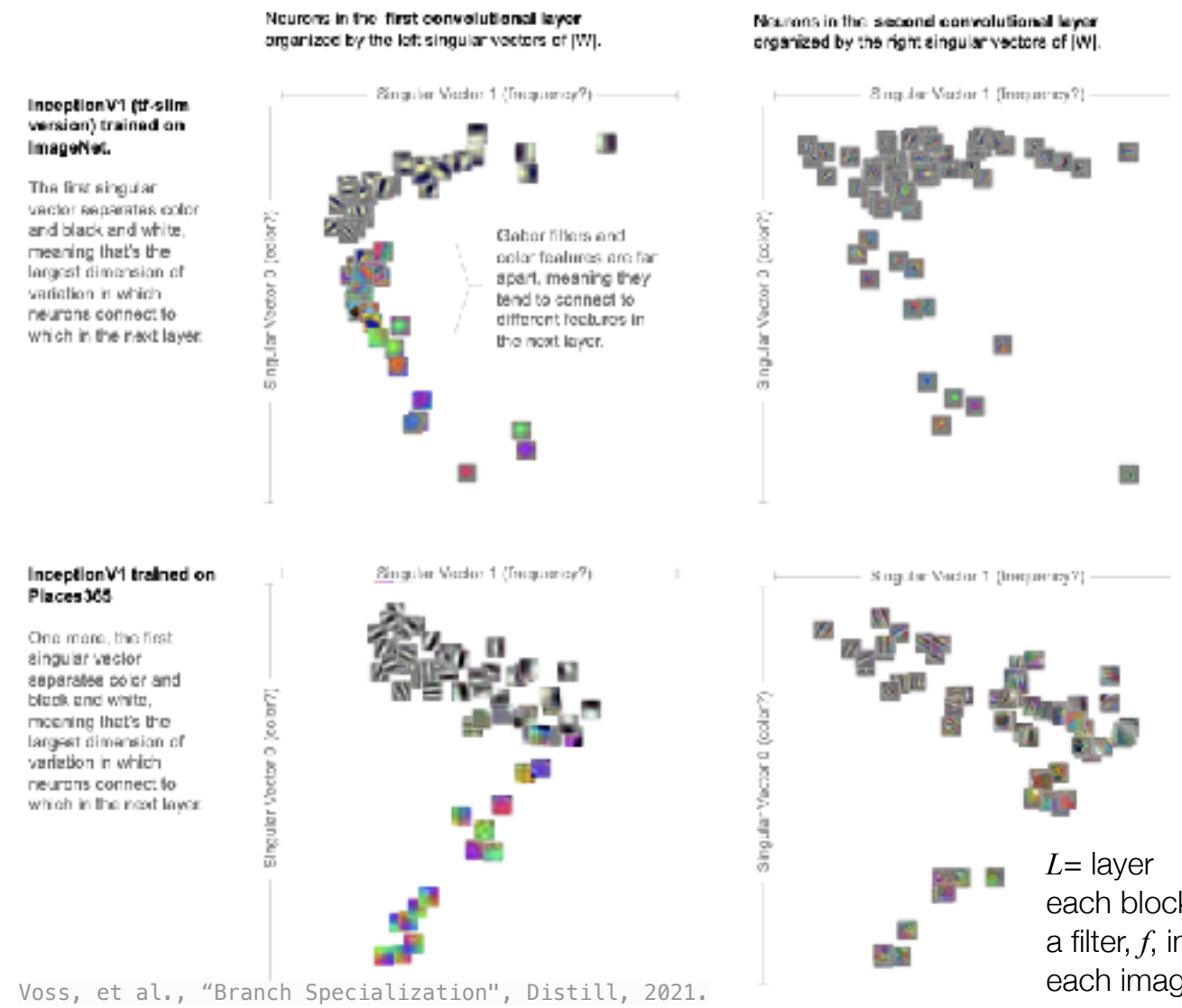
Branch Specialization



Motifs appear in branches. Similar clusters of operations can be found across different architectures

Voss, et al., “Branch Specialization", Distill, 2021.

Investigating Connection Clusters via SVD



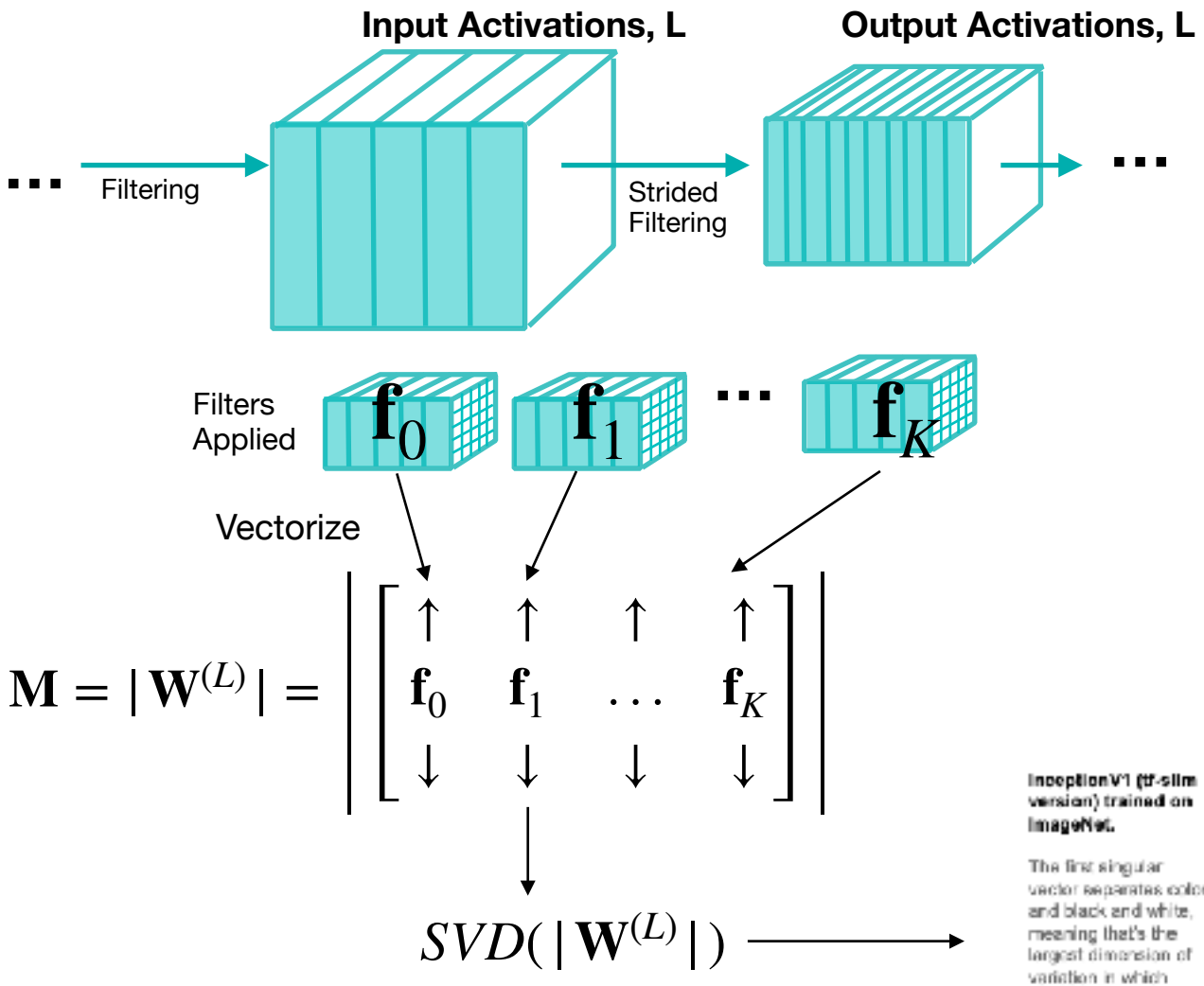
- Singular Value Decomposition (SVD) decomposes a matrix into three elements
  - $M = U\Sigma V^T$
  - U is eig-vec of MM^T
    V is eig-vec of M^TM
  - U and V are orthogonal such that
    $UU^T=I \quad VV^T=I$
  - Σ is a diagonal matrix of the singular values
  - These values characterize the variability in a matrix

L= layer
each block is a filter, f, in layer
each image is the optimized excitation

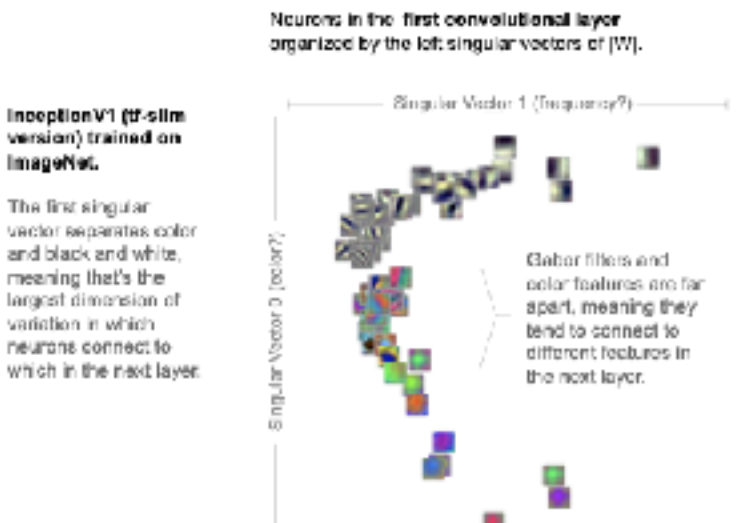
$$SVD(|\mathbf{W}^{(L)}|)$$

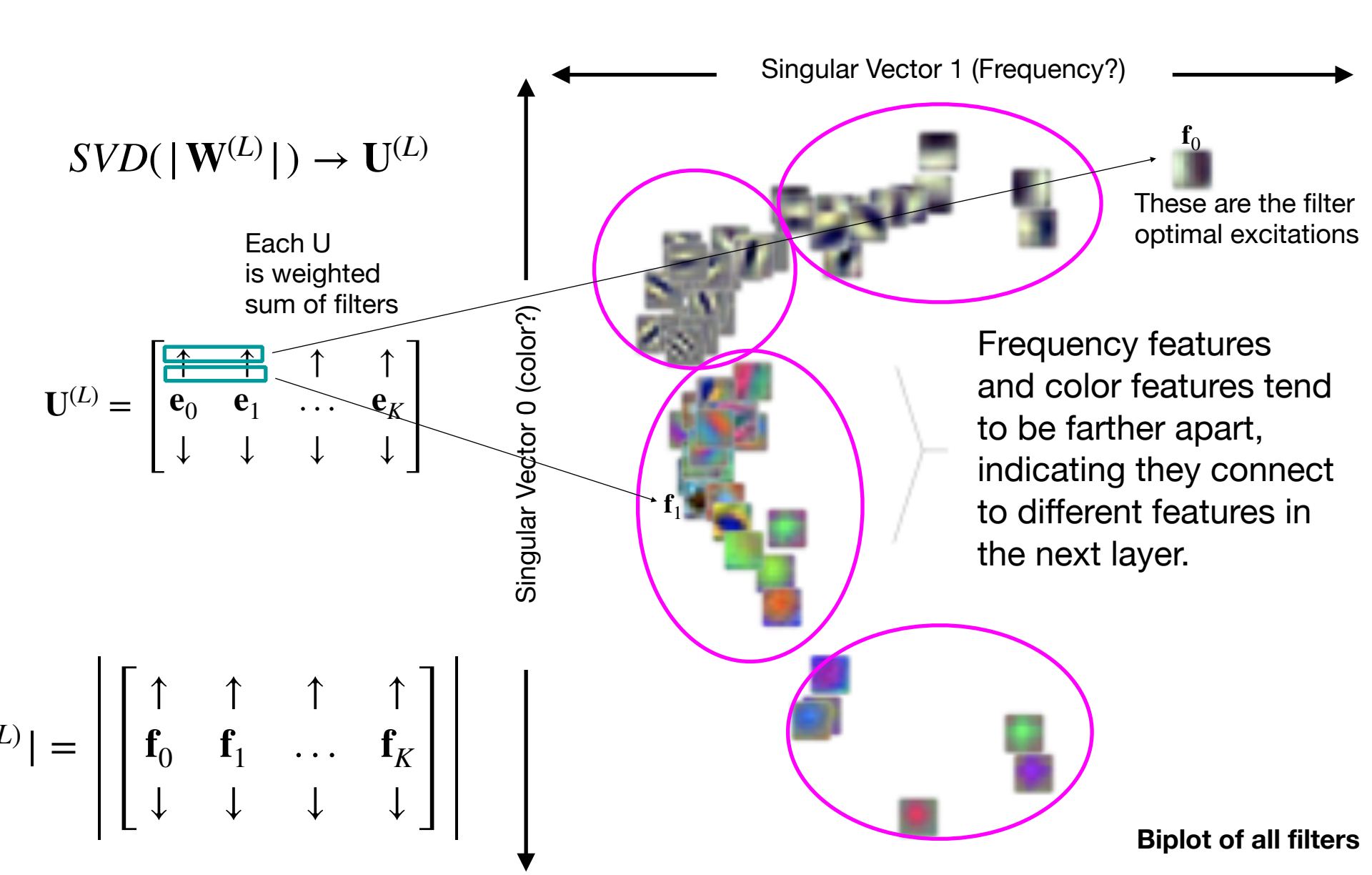
Voss, et al., "Branch Specialization", Distill, 2021.

Structure of Each Tensor:
Channels x Rows x Columns

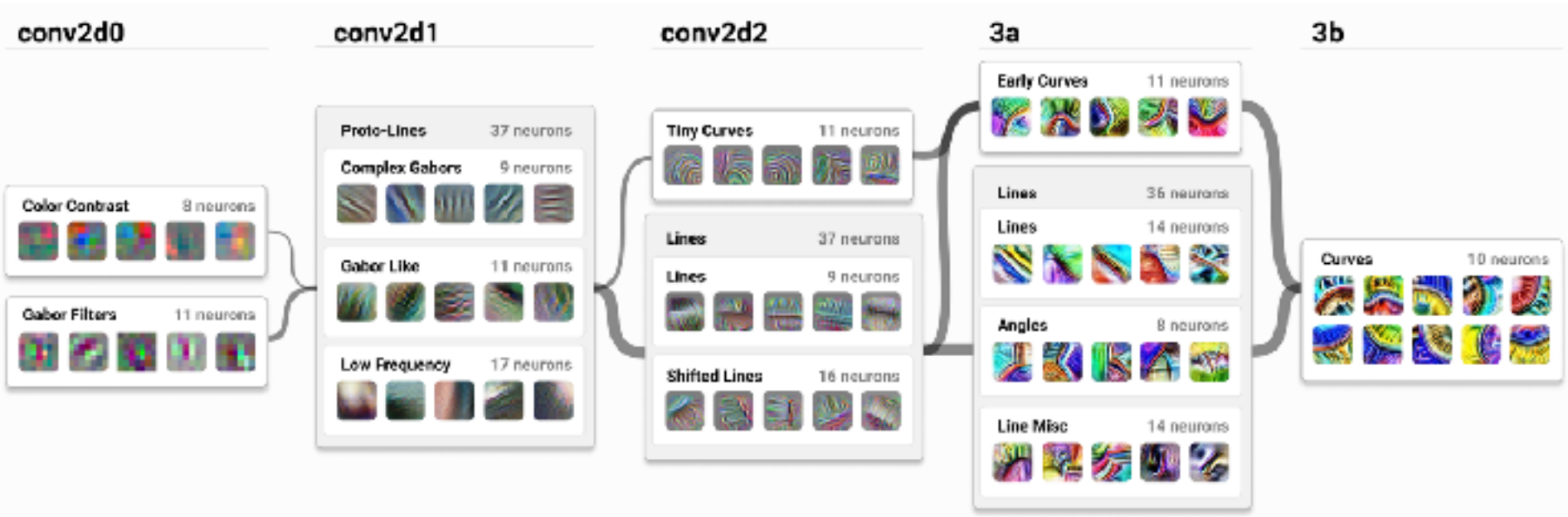


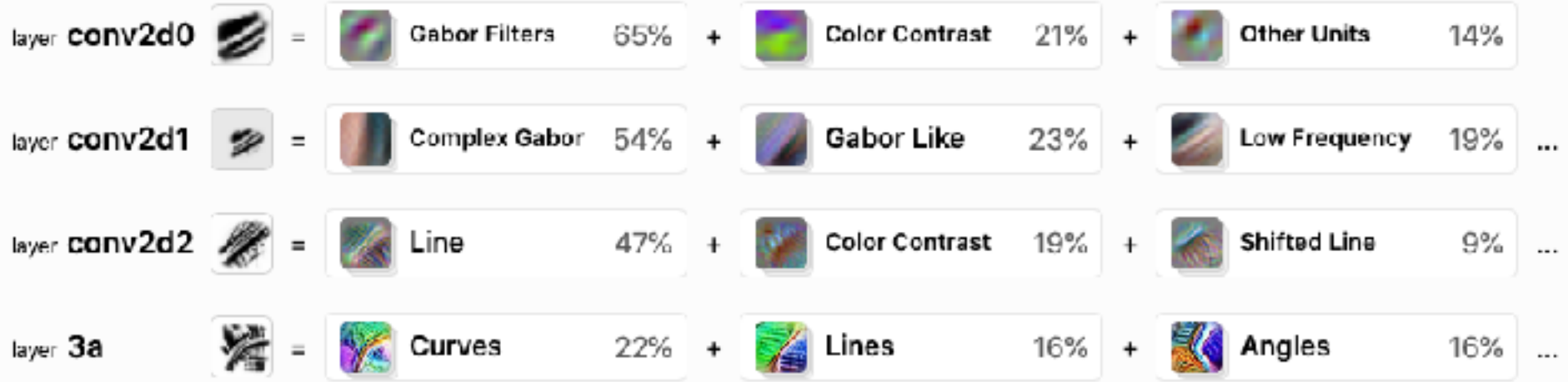
- Singular Value Decomposition (SVD) decomposes a matrix into three elements
  - $M = U\Sigma V^T$
  - U is eig-vec of MM^T
    V is eig-vec of M^TM
  - U and V are orthogonal such that
    $UU^T=I \quad VV^T=I$
  - Σ is a diagonal matrix of the singular values
  - These values characterize the variability in a matrix




$SVD(|\mathbf{W}^{(L)}|) \rightarrow \mathbf{U}^{(L)}$
Each U
is weighted
sum of filters
$\mathbf{U}^{(L)} = \begin{bmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \mathbf{e}_0 & \mathbf{e}_1 & \cdots & \mathbf{e}_K \\ \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix}$
$|\mathbf{U}^{(L)}| = \left| \begin{bmatrix} \uparrow & \uparrow & \uparrow & \uparrow \\ \mathbf{f}_0 & \mathbf{f}_1 & \cdots & \mathbf{f}_K \\ \downarrow & \downarrow & \downarrow & \downarrow \end{bmatrix} \right|$
Singular Vector 1 (Frequency?)
Singular Vector 0 (color?)
$\mathbf{f}_0$
$\mathbf{f}_1$
These are the filter
optimal excitations
Frequency features
and color features tend
to be farther apart,
indicating they connect
to different features in
the next layer.
Biplot of all filters
Clusters tend to be motifs, and
separation of clusters reveals connections between layers (like edges in a graph)

Neural Nets: Directed Graph of Circuits

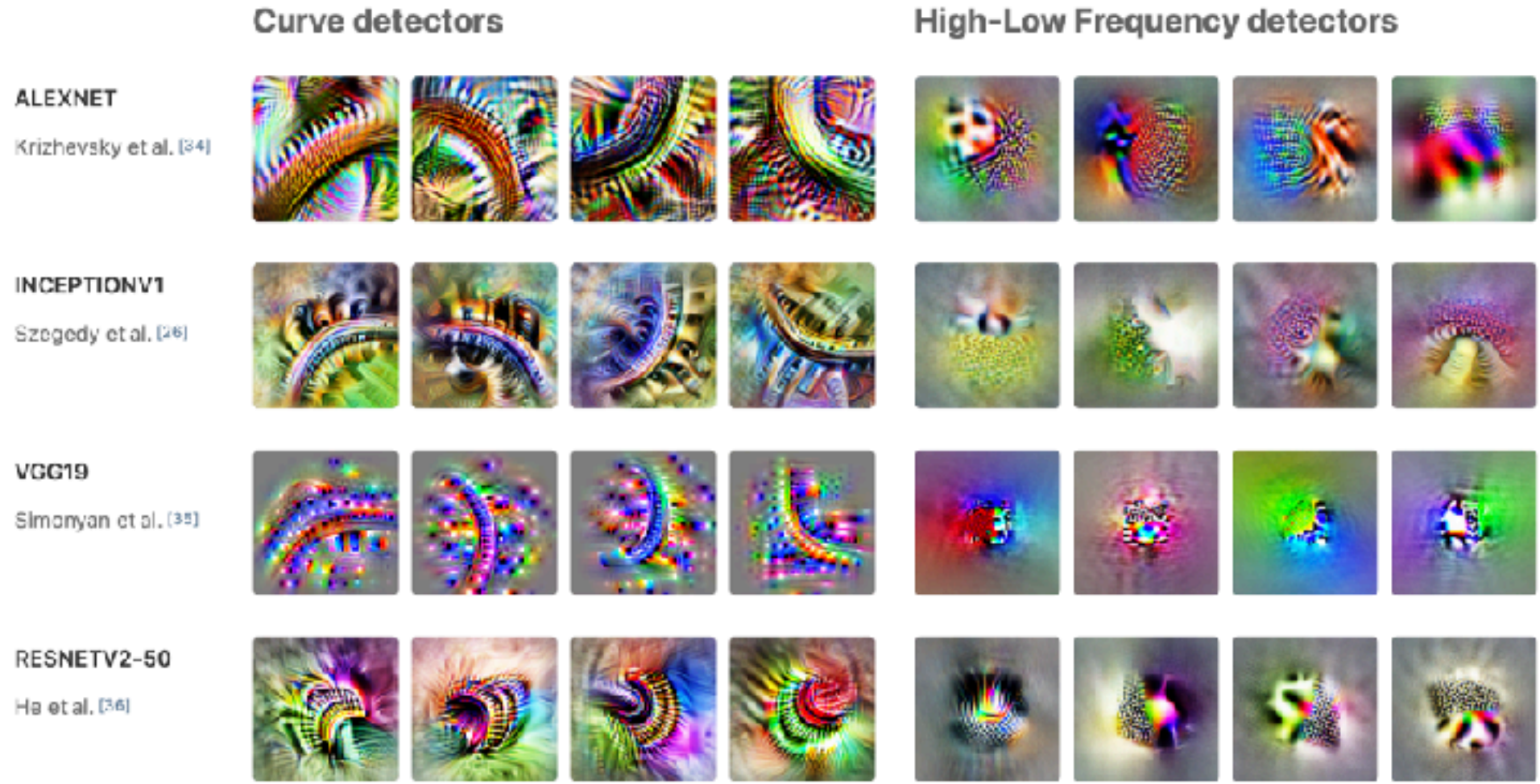




Voss, et al., "Branch Specialization", Distill, 2021.

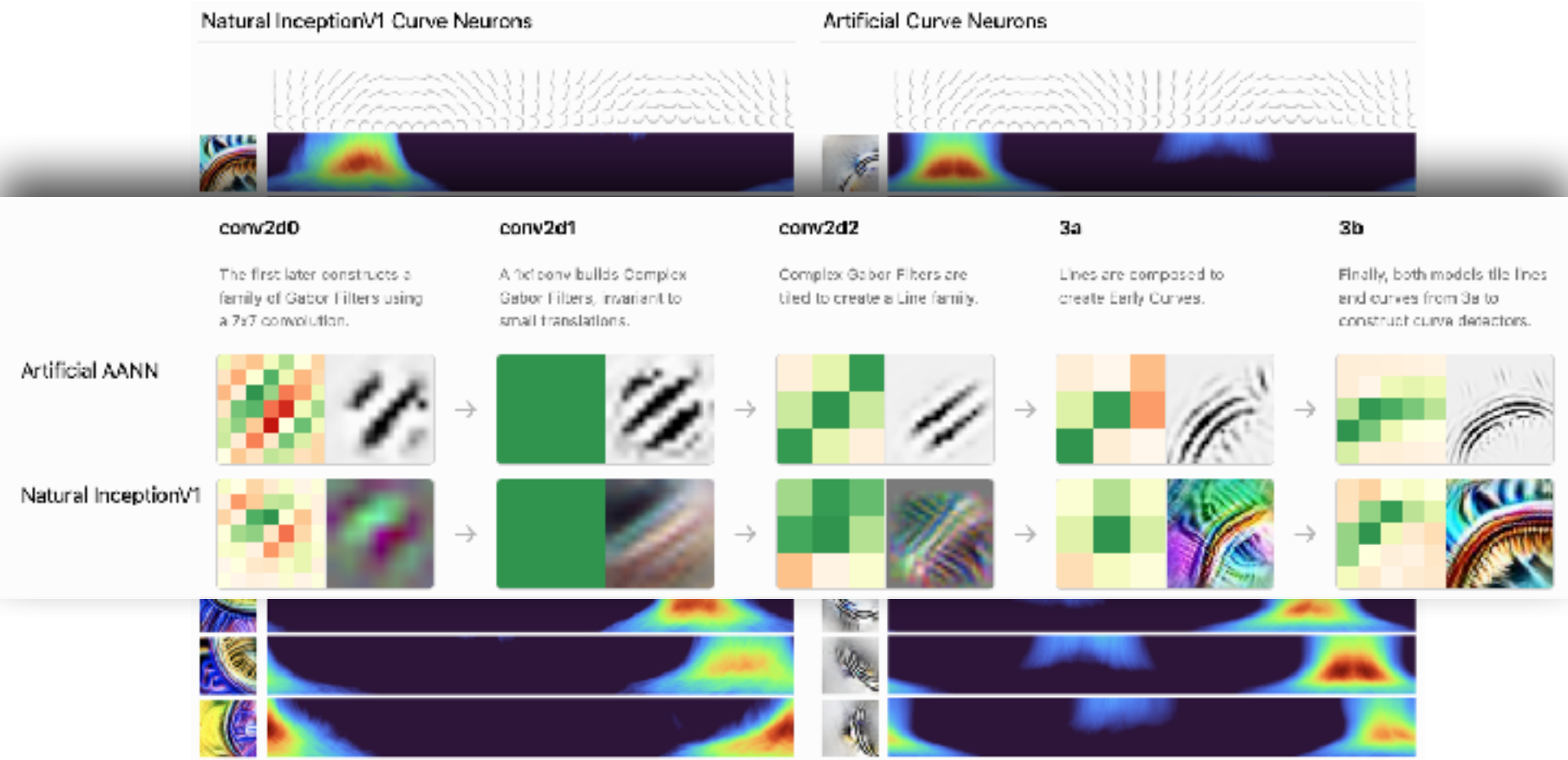
Universality of Circuits

- Analogous features and circuits form across models and tasks



Reverse Engineering a Circuit

- With assumption of what feature is, a circuit can be implemented by hand that nearly identically follows the assumed functionality



Closing Thoughts from OpenAI Researchers

Closing Thoughts

We take it for granted that the microscope is an important scientific instrument. It's practically a symbol of science. But this wasn't always the case, and microscopes didn't initially take off as a scientific tool. In fact, they seem to have languished for around fifty years. The turning point was when Robert Hooke published Micrographia [1], a collection of drawings of things he'd seen using a microscope, including the first picture of a cell.

Our impression is that there is some anxiety in the interpretability community that we aren't taken very seriously. That this research is too qualitative. That it isn't scientific. But the lesson of the microscope and cellular biology is that perhaps this is expected. The discovery of cells was a qualitative research result. That didn't stop it from changing the world.

https://distill.pub/2020/circuits/zoom-in/

Lecture Notes for

Neural Networks and Machine Learning

CNN Circuits

Next Time:

Fully Convolutional Learning

Reading: Chollet 5.4