Neuron-based explanations of neural networks sacrifice completeness and interpretability

Nolan Dey *nolan@cerebras.net*
Cerebras Systems, University of Waterloo, Vector Institute

Eric Taylor *eric.taylor@vectorinstitute.ai*
Borealis AI, Vector Institute

Alexander Wong *alexander.wong@uwaterloo.ca*
University of Waterloo, Apple

Bryan Tripp *bptripp@uwaterloo.ca*
University of Waterloo

Graham W. Taylor *gwtaylor@uoguelph.ca*
University of Guelph, Vector Institute

Reviewed on OpenReview: *https://openreview.net/forum?id=UWNa9Pv6qA*

TL;DR: The most important principal components provide more complete and interpretable explanations than the most important neurons.

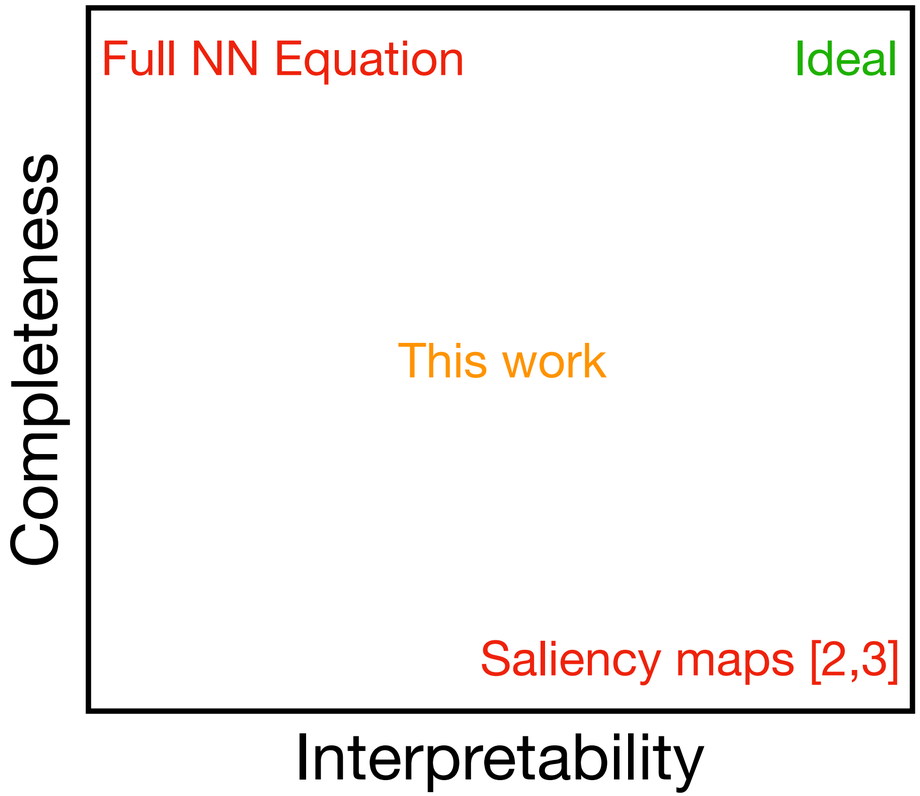
The urgency of interpretability

“Many of the risks and worries associated with generative AI are ultimately consequences of this opacity, and would be much easier to address if the models were interpretable.”

–Dario Amodei (https://www.darioamodei.com/post/the-urgency-of-interpretability)

Completeness and interpretability

- High quality explanations should be complete and interpretable [1]
- Completeness = accurately reflect a NN's function
- Interpretability = understandable to humans
- Popular NN explanation methods make choices that increase interpretability at the expense of completeness.



[1] Leilani H Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), pp. 80–89, 2018.

[2] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR, abs/1312.6034, 2013.

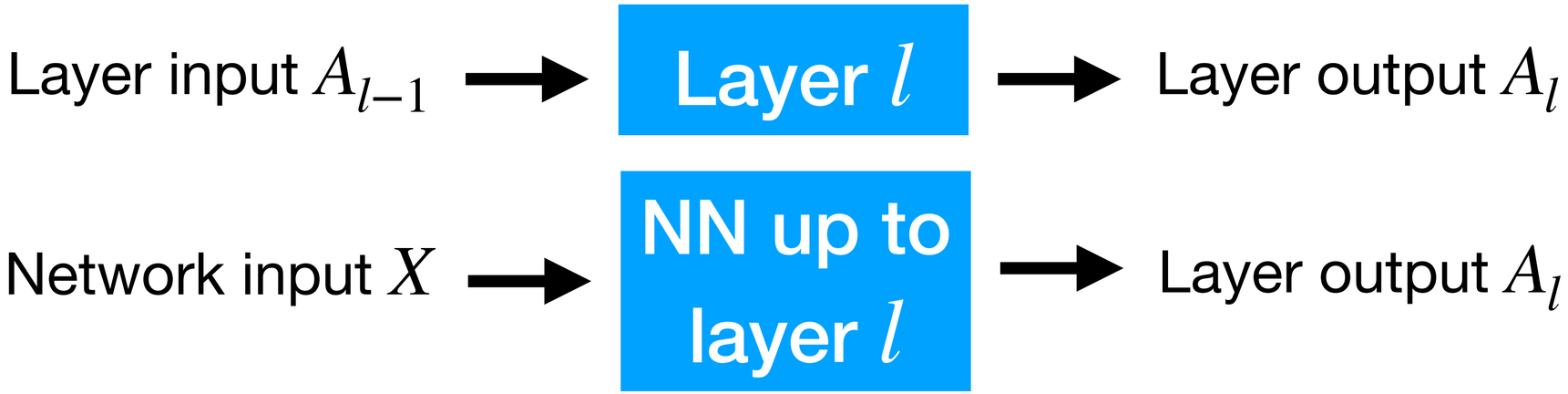
[3] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods, 2017.

Explainability philosophy

- Each NN layer outputs a nonlinear transformation of its input
- **Goal:** Understand each layer's nonlinear transformation by explaining how output differs from input
- **Problem:** Humans cannot naturally understand NN latent space

Explainability philosophy

- Each NN layer outputs a nonlinear transformation of its input
- **Ideal Goal:** Understand each layer's nonlinear transformation by explaining how output differs from input
- **Problem:** Humans cannot naturally understand NN latent space, only input space X
- **Refined goal:** Understand how NN input X is transformed to produce A_l



Method overview

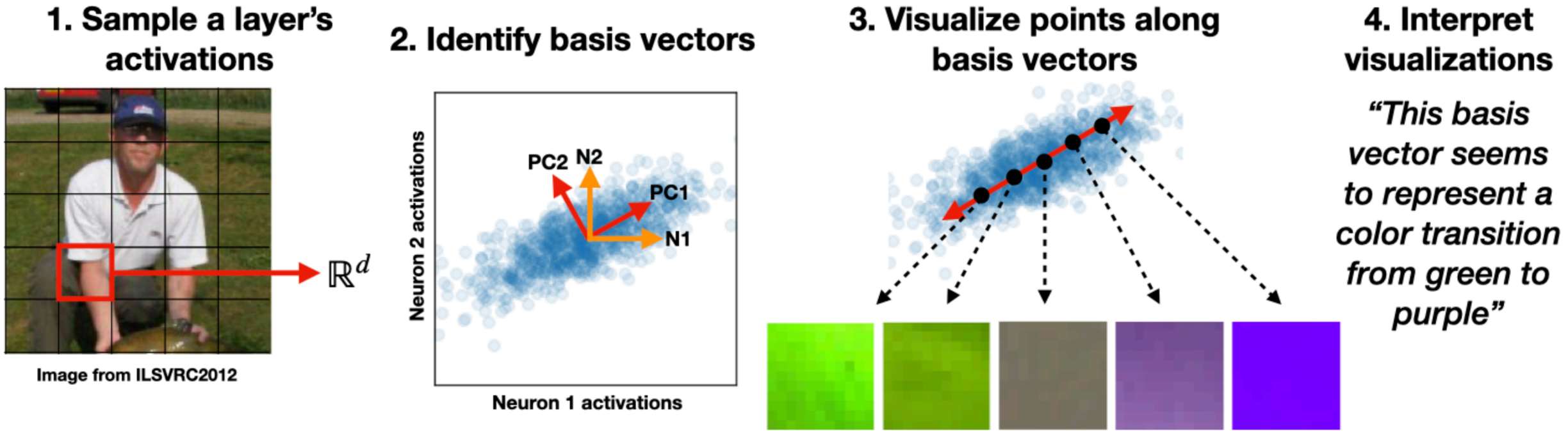
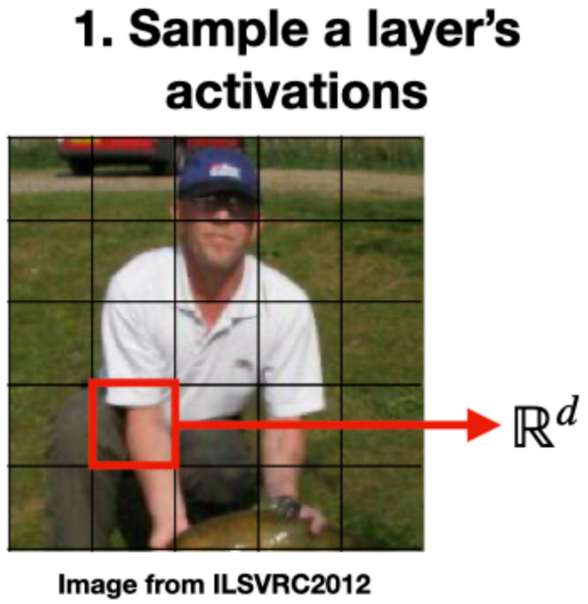


Figure 1: Overview of our methodology. We sample $\mathbb{R}^d$ activations from a layer (1), then identify basis vectors of the layer's activation space (e.g. neurons or PCs) (2). Finally we visualize points along each basis vector (3) and interpret the visualizations (4).

Sampling a layer's activations (1/4)

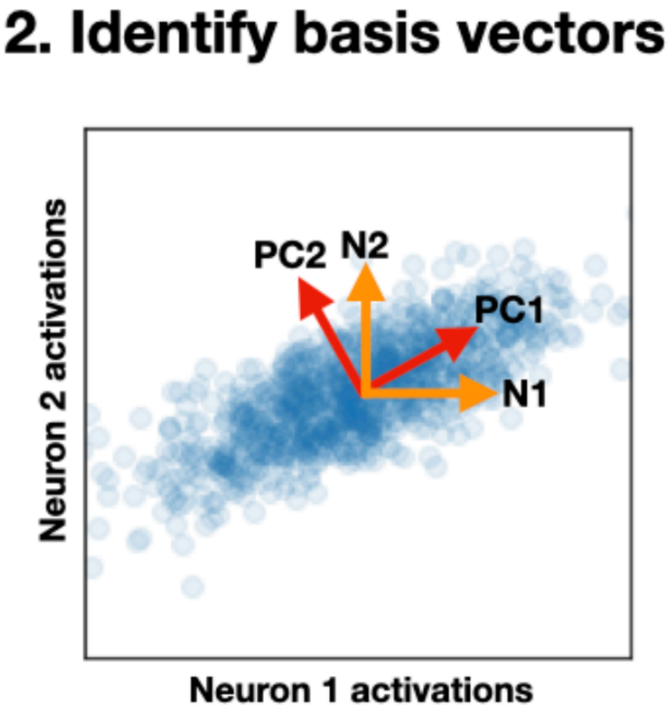
- Construct $A \in \mathbb{R}^{n \times d}$ by randomly sampling a single $\mathbb{R}^d$ vector from each of the n ImageNet examples
  - d is either n_channels or n_neurons
- Sample pre-nonlinearity



Network input $X \in \mathbb{R}^{l_X \times w_X \times d_X}$ → **NN up to layer l** → Layer output $A_{total} \in \mathbb{R}^{l \times w \times d}$

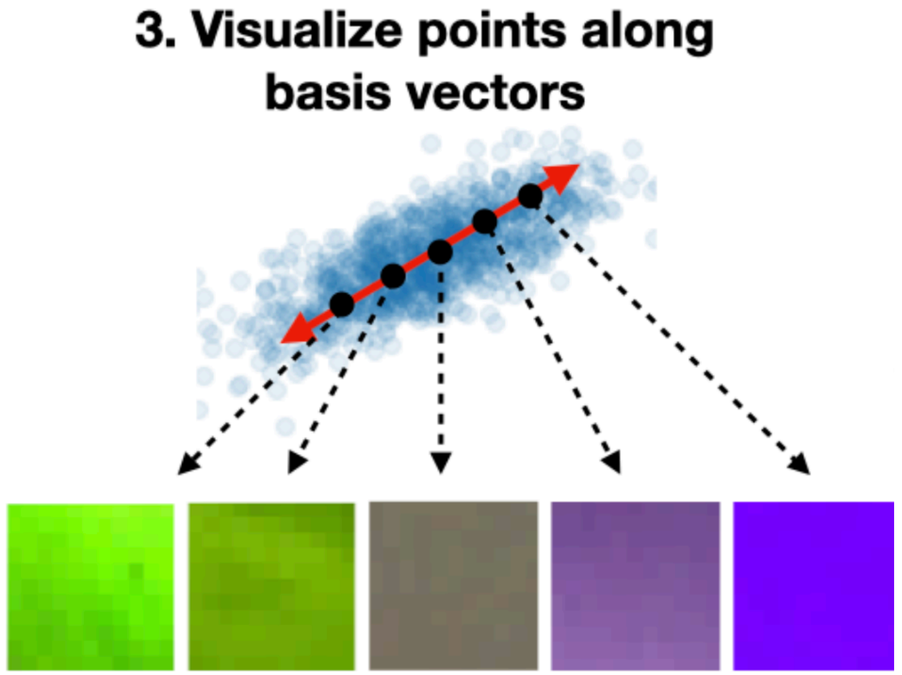
Identifying basis vectors (2/4)

- **Difficulties** with choosing a basis
  - High-dimensional activations
  - Distributed representations: Multiple neurons fire together to represent a concept
  - Human attention is limited; we can't look at every basis vector
- **Naive/common basis:** Neuron basis
  - Axis-aligned (e.g. orange arrows)
- **Alternative:** Principle component basis
  - Aligned to directions of largest variance (e.g. red arrows)



Visualizing points along basis vectors (3/4)

- Sample m points along each basis vector ($\mathcal{U}[\min, \max]$)
- For each point $a_{\text{target}} \in \mathbb{R}^d$, we find k receptive-field-sized image patch whose activations minimize the ℓ_2 distance to a_{target}



Distance minimizing vs activation maximization

- Activation maximization makes it more difficult to isolate the effect of a single basis vector

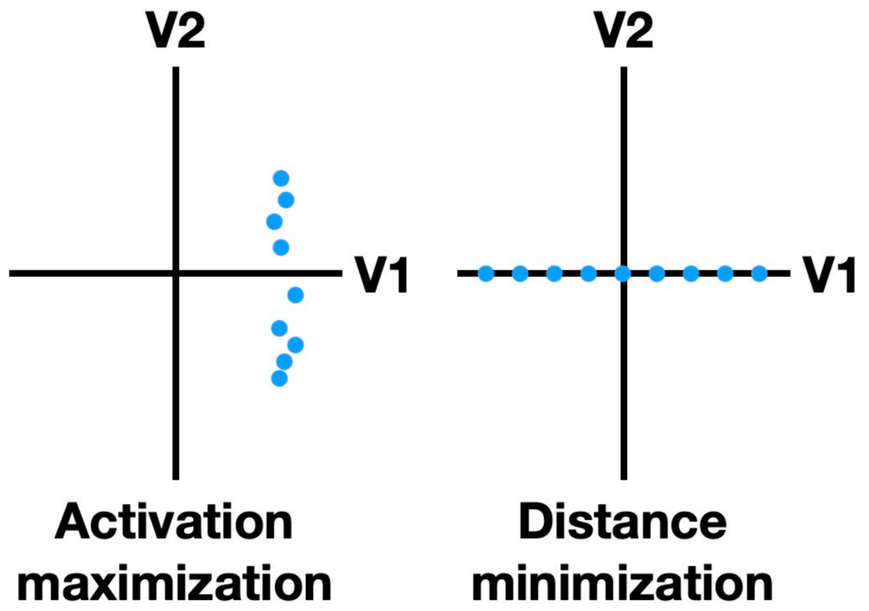
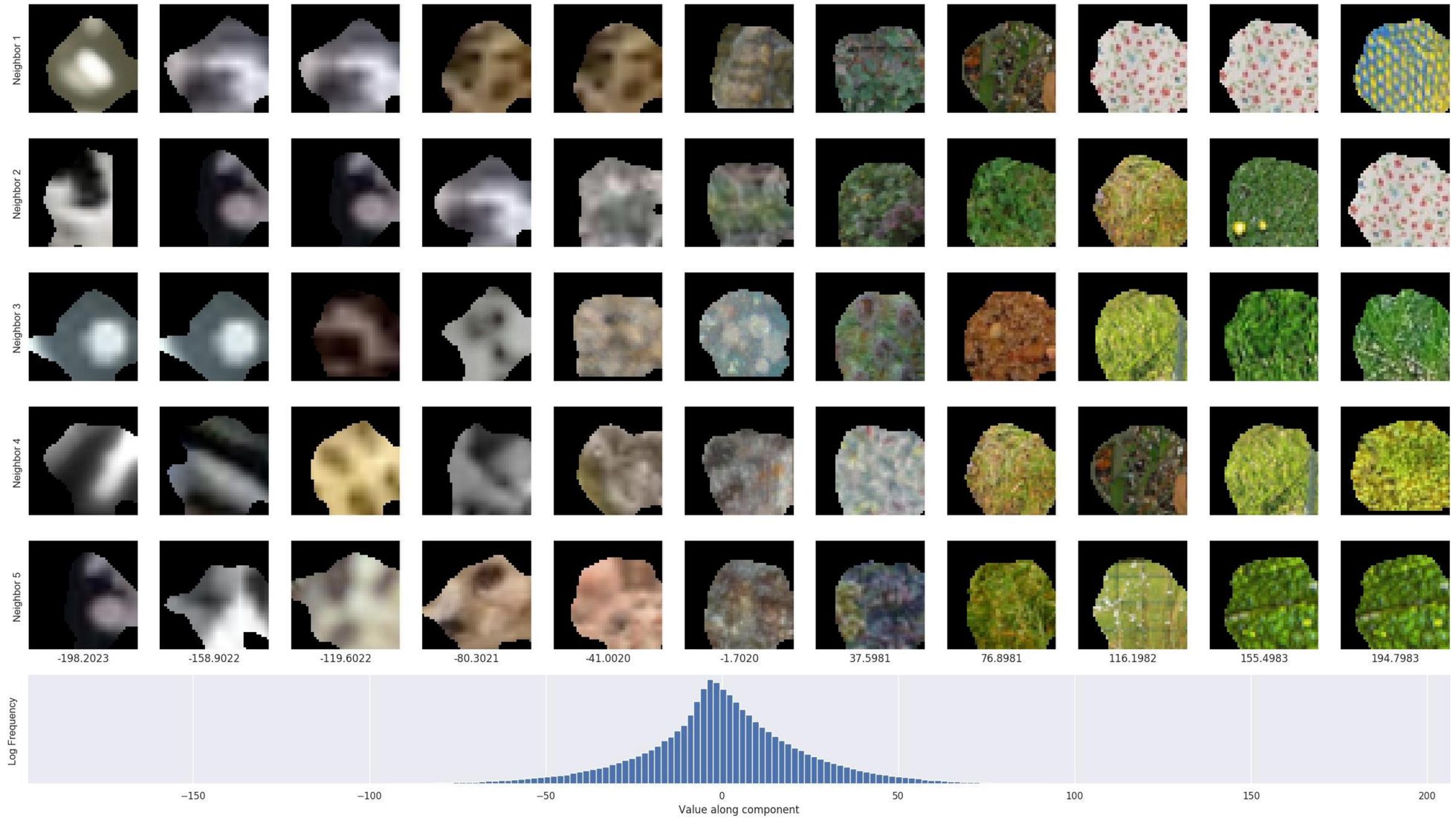


Figure 12: Toy example illustrating the difference between activation maximization and distance minimization when attempting to visualize the basis vector V1. Blue dots represent the resulting optimized visualizations.

Demo

- Interactive demo: https://ndey96.github.io/neuron-explanations-sacrifice
- Includes AlexNet, ResNet-18, ResNet-50, ViT-B/16



Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

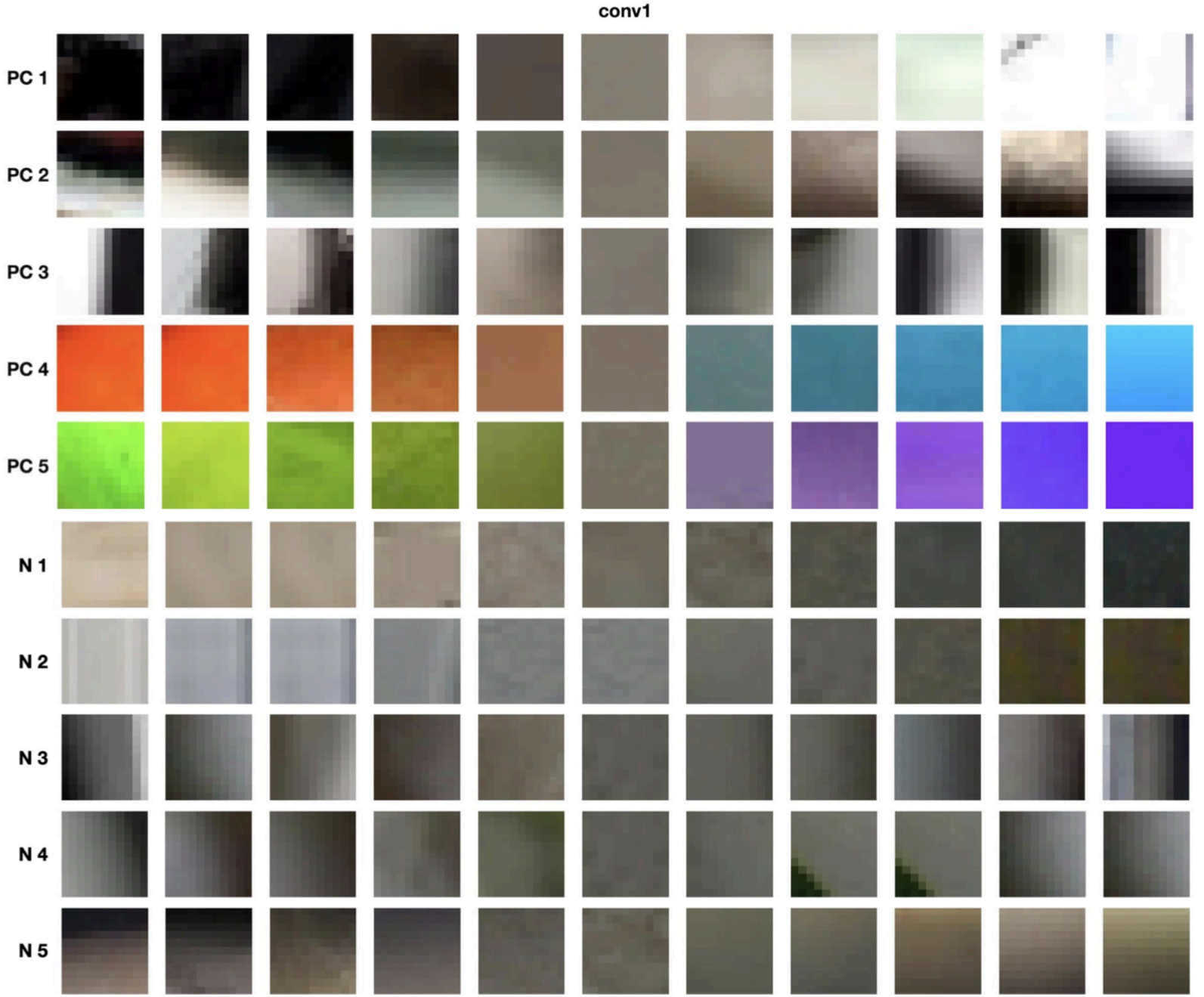


Figure 2: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `conv1` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

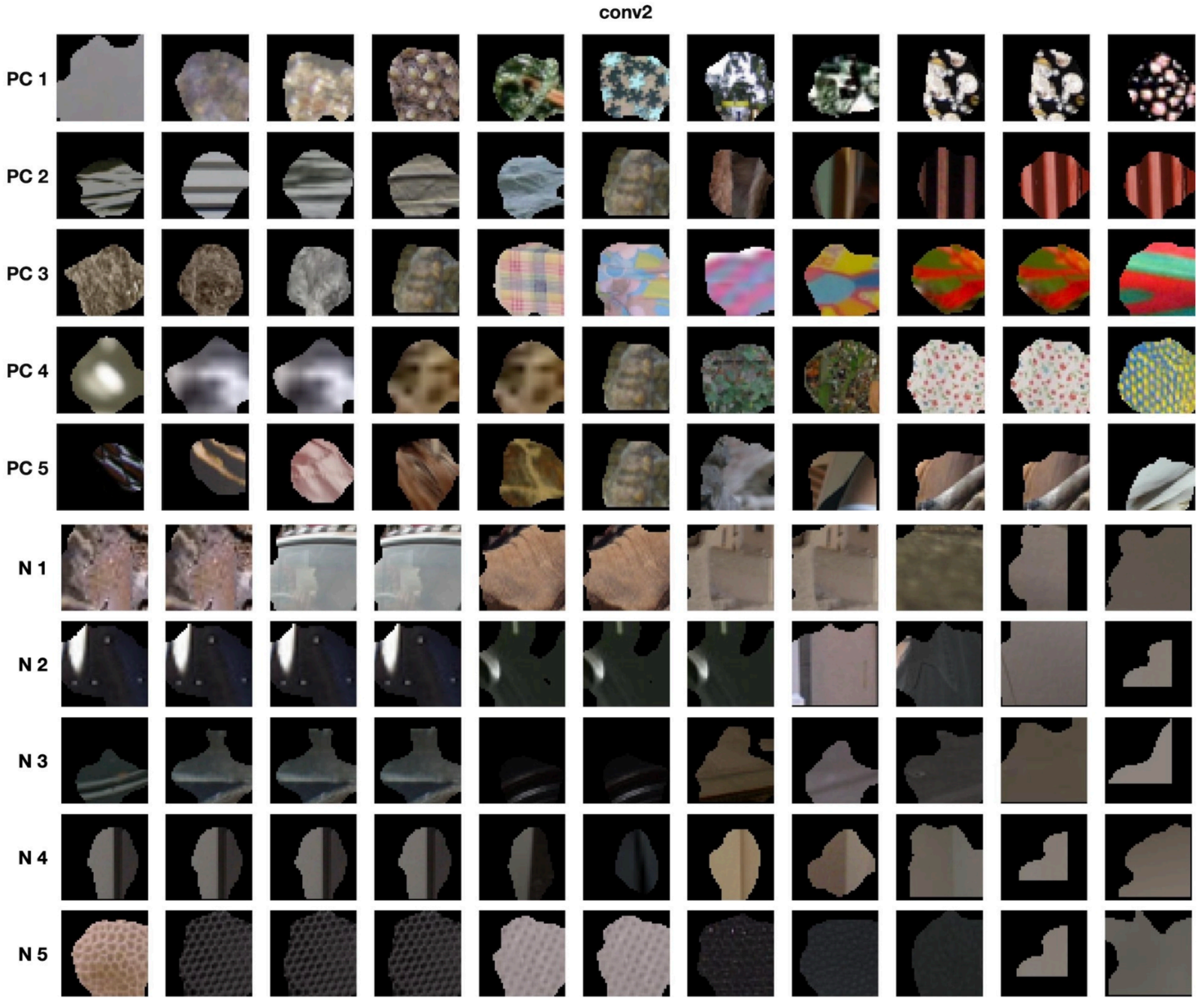


Figure 3: Visualizations of points along the 6 highest variance PCs and 5 highest variance neurons for AlexNet's `conv2` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

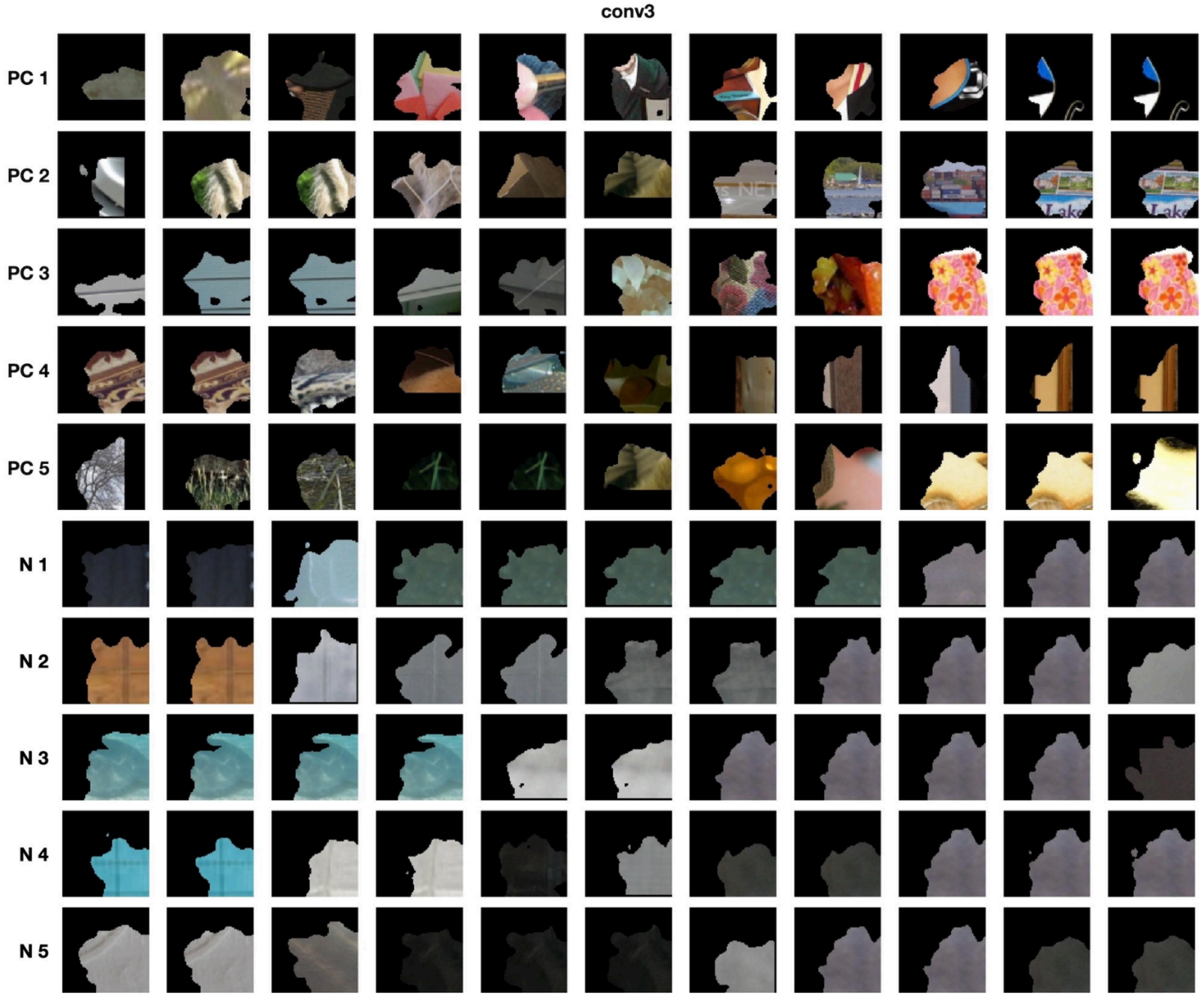


Figure 16: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `conv3` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

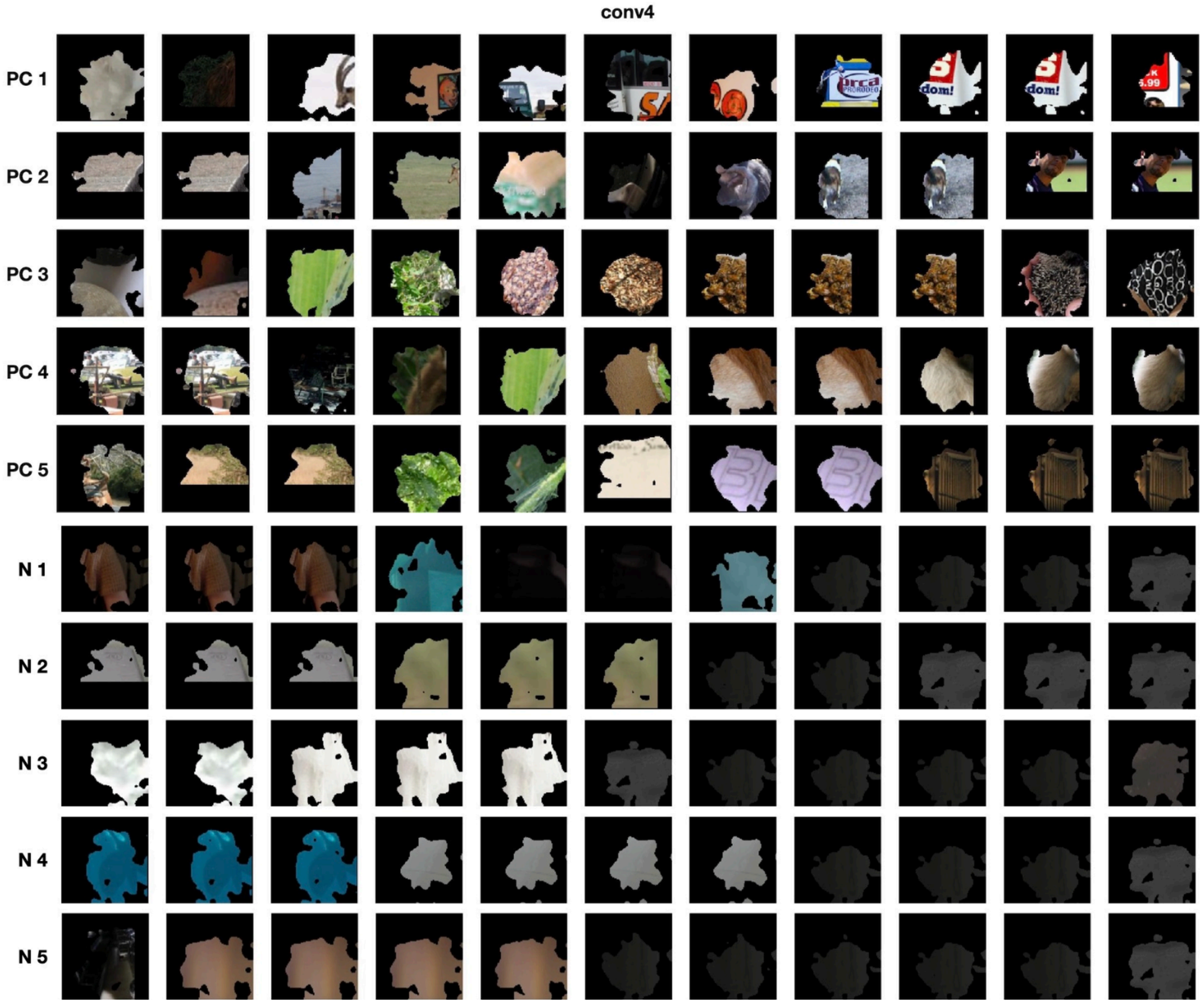


Figure 17: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `conv4` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

Figure 18: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `conv5` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

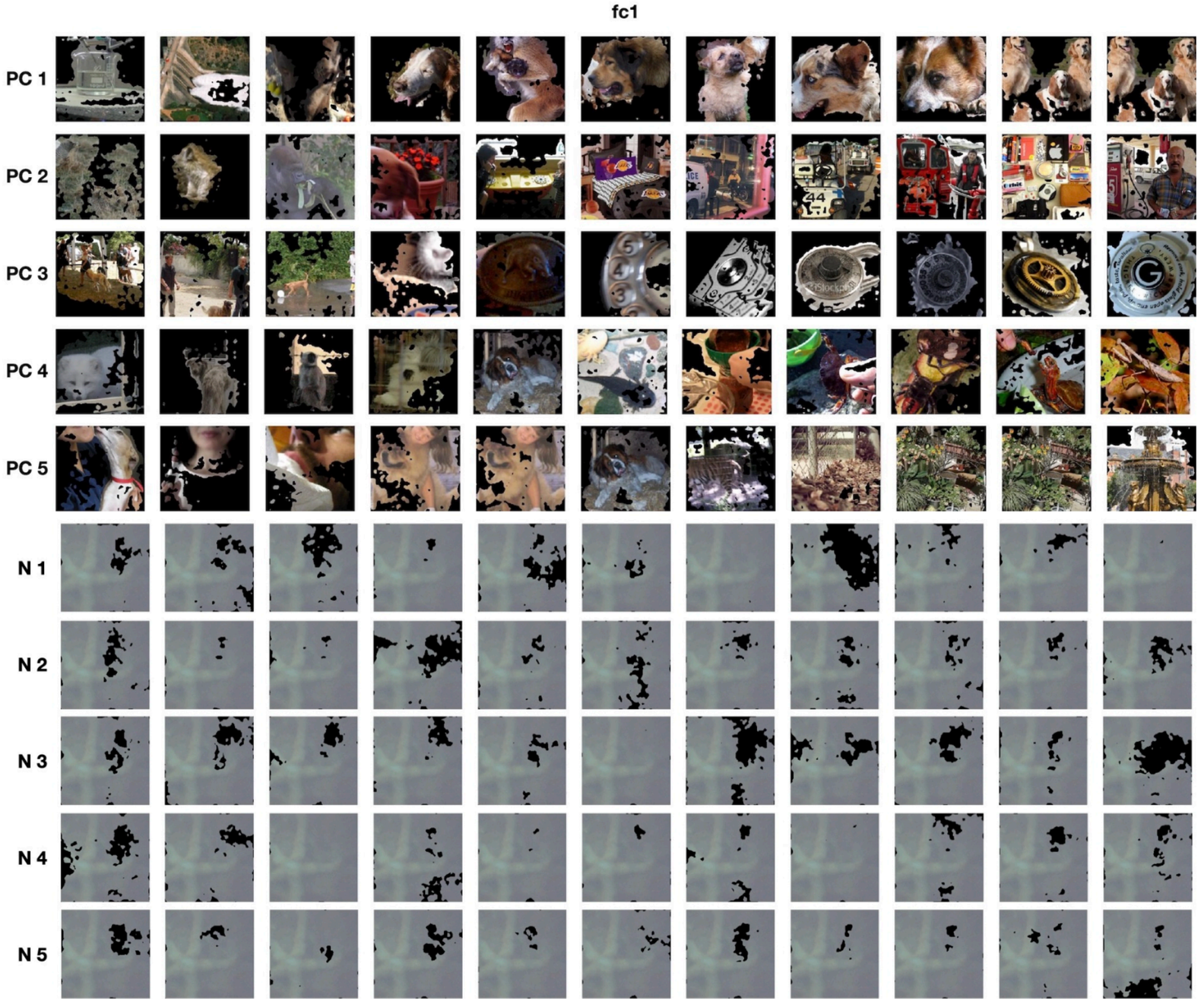


Figure 4: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `fc1` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

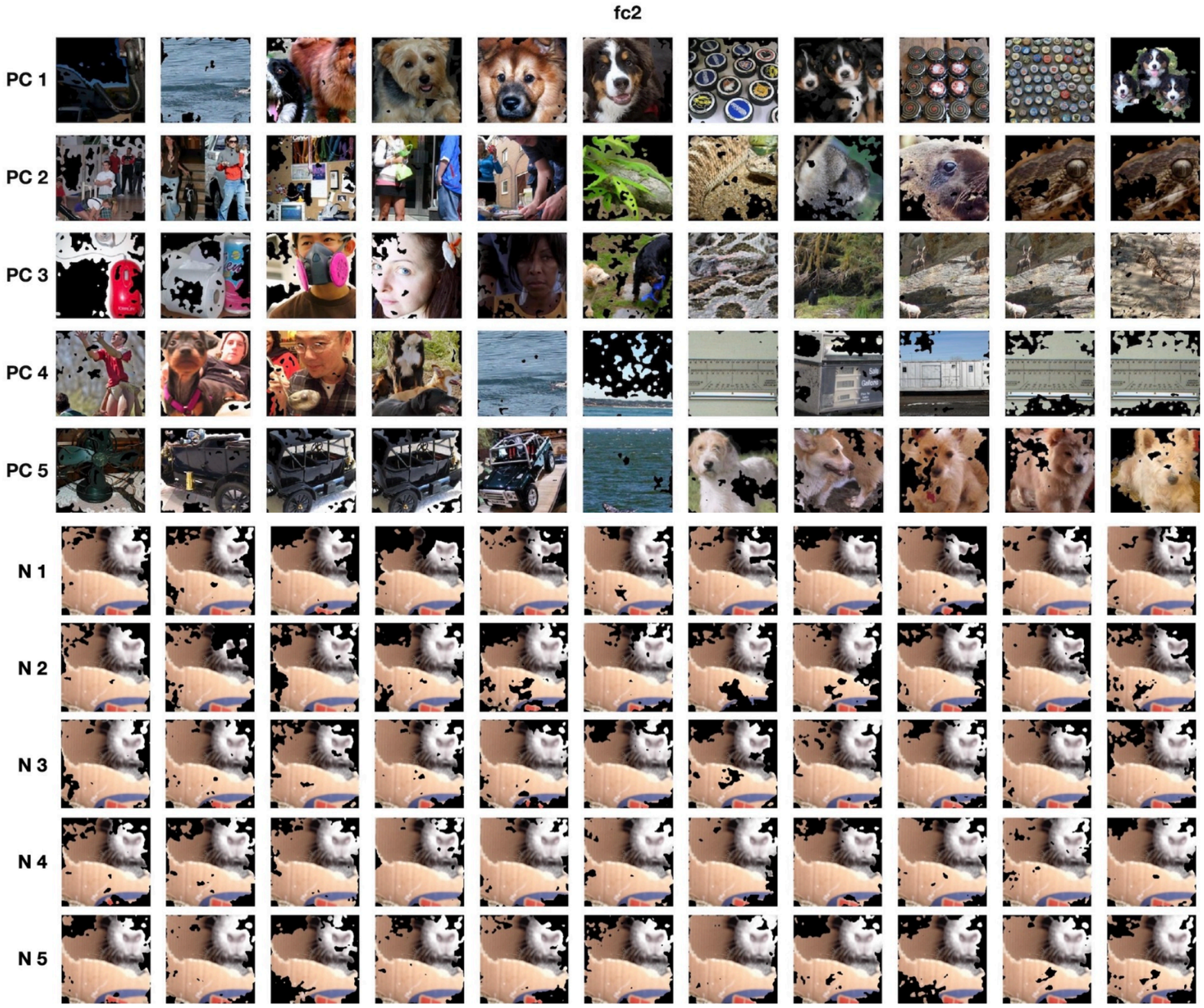


Figure 19: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `fc2` layer.

Interpreting visualizations (4/4)

Layer name	Layer type	Activation	Parameters	Output shape
	Input			3 × 224 × 224
conv1	Conv2d	ReLU	k=11,s=4,p=2	64 × 55 × 55
	MaxPool2d		k=3,s=2,p=0	64 × 27 × 27
conv2	Conv2d	ReLU	k=5,s=1,p=2	192 × 27 × 27
	MaxPool2d		k=3,s=2,p=0	192 × 13 × 13
conv3	Conv2d	ReLU	k=3,s=1,p=1	384 × 13 × 13
conv4	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
conv5	Conv2d	ReLU	k=3,s=1,p=1	256 × 13 × 13
	MaxPool2d		k=3,s=2,p=0	256 × 6 × 6
	AdaptiveAvgPool2d			256 × 6 × 6
	Flatten			9216
	Dropout		d=0.5	9216
fc1	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc2	Linear	ReLU		4096
	Dropout		d=0.5	4096
fc3	Linear	ReLU		1000

Figure 20: Visualizations of points along the 5 highest variance PCs and 5 highest variance neurons for AlexNet's `fc3` layer.

Quantifying completeness: Explained variance

- Cumulative sum of explained variance ratio of top-k basis vectors $= \frac{\sum_{i=1}^{k} \text{Var}(A_i')}{\sum_{j=1}^{n} \text{Var}(A_j')}$

- Much of the activation variance is concentrated in the most important PCs (blue line) whereas explained variance is far less concentrated in the neuron basis (orange line).

- For example, to explain 80% of the activation variance for fc1, one could either study the first 42 PCs, or the 2782 highest variance neurons.

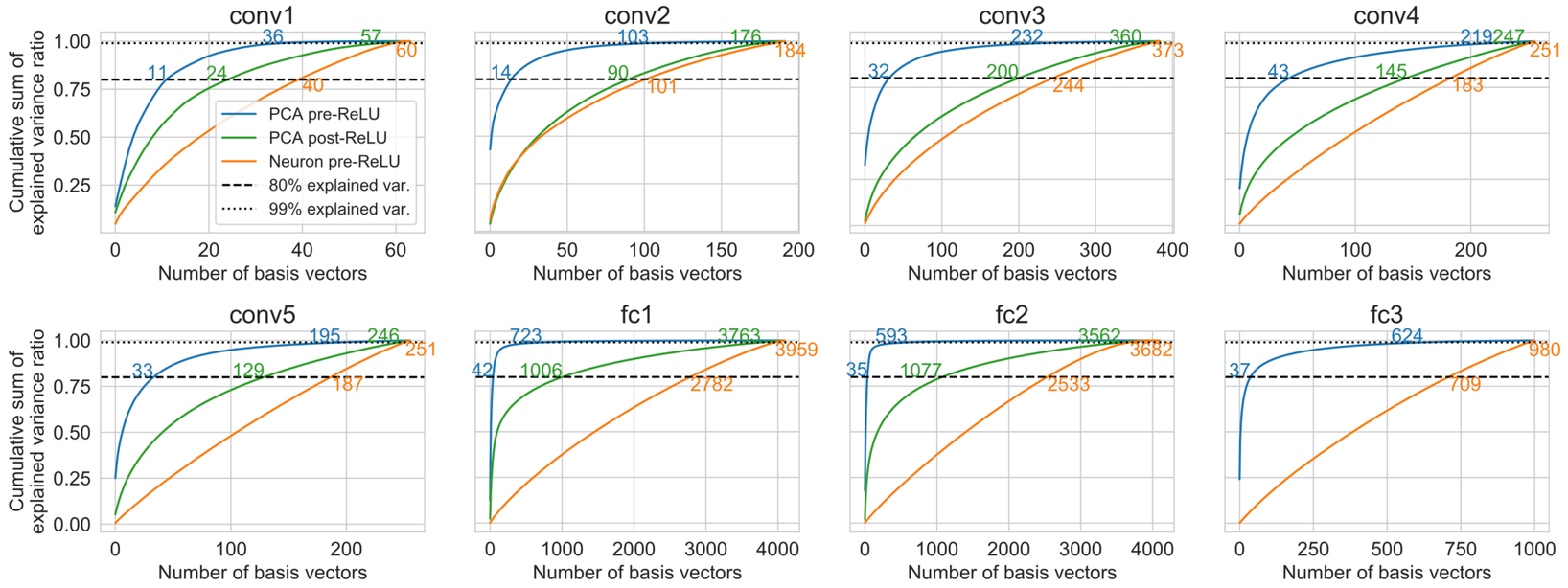


Figure 5: Cumulative sum of explained variance ratio for each AlexNet layer plotted against the number of basis vectors being used. Both PCs and neurons are ordered by descending variance. The number of basis vectors required to explain 80% and 99% variance is annotated.

Quantifying completeness: Activation ablation

- Cumulatively ablate basis vectors and observe how much accuracy degrades. Basis vectors more important for a network's function should degrade accuracy rapidly compared to less important basis vectors.

- For most layers in AlexNet, ablating the highest variance PCs (solid blue line) damages accuracy more than ablating the highest variance neurons (solid orange line).

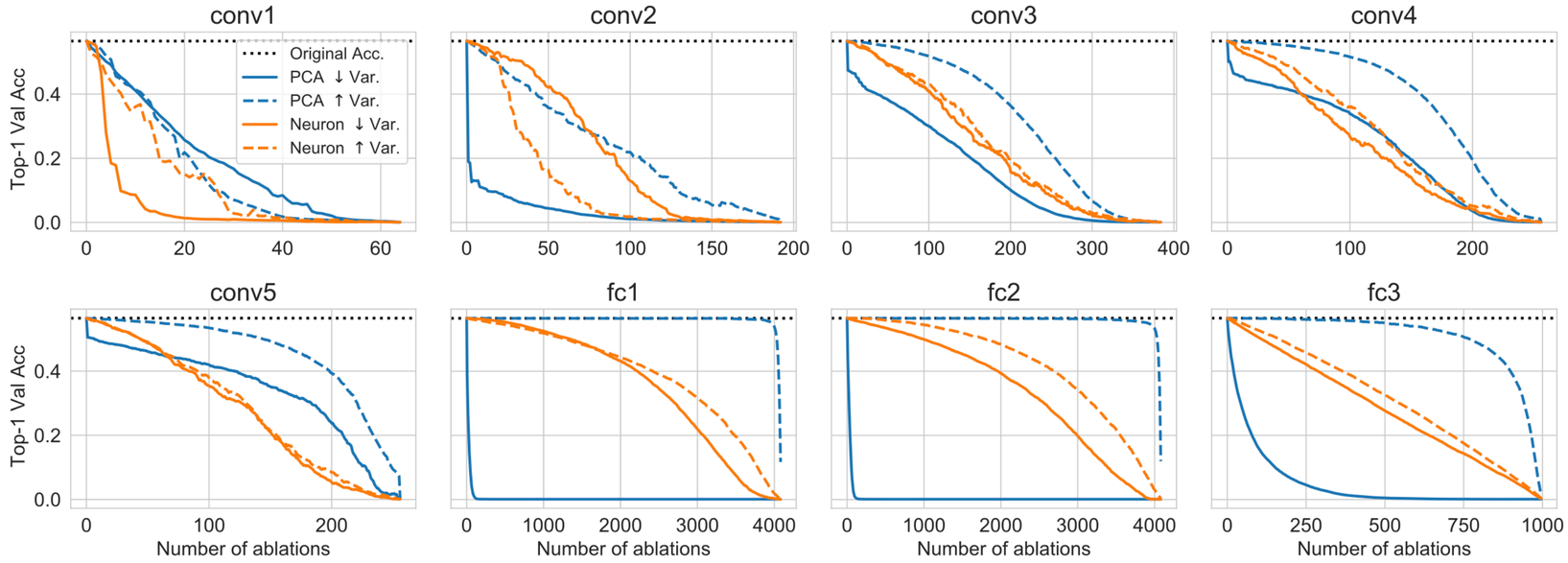
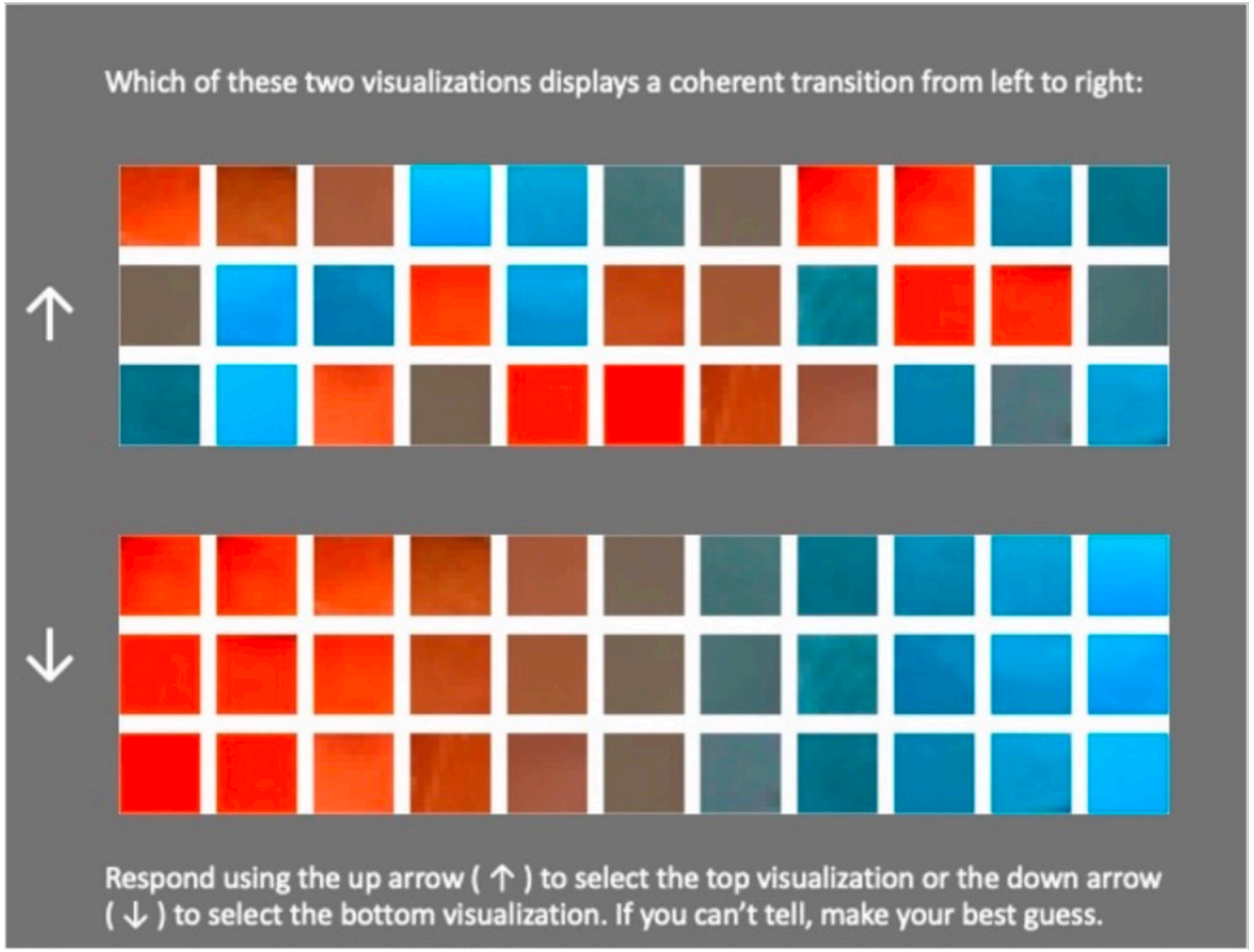


Figure 6: For each AlexNet layer, we ablate basis vectors in activation space and measure the effect on ImageNet top-1 validation accuracy. Both PCs and neurons are ordered by their explained variance. Reverse order corresponds to ascending explained variance.

Quantifying interpretability: User study (1/3)



Quantifying interpretability: User study (2/3)

Figure 14: Screenshot of the user study task.

Quantifying interpretability: User study (3/3)

- PC visualizations were, on average, more interpretable than Neuron visualizations for each layer in AlexNet with the most pronounced differences seen in layers conv2, fc1, and fc2.

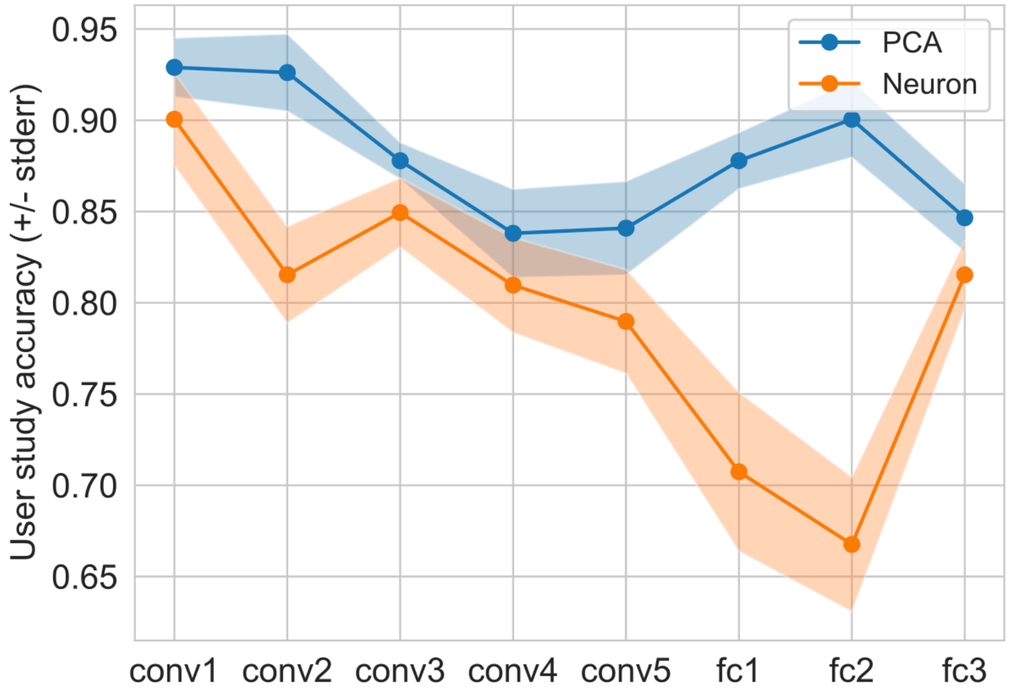


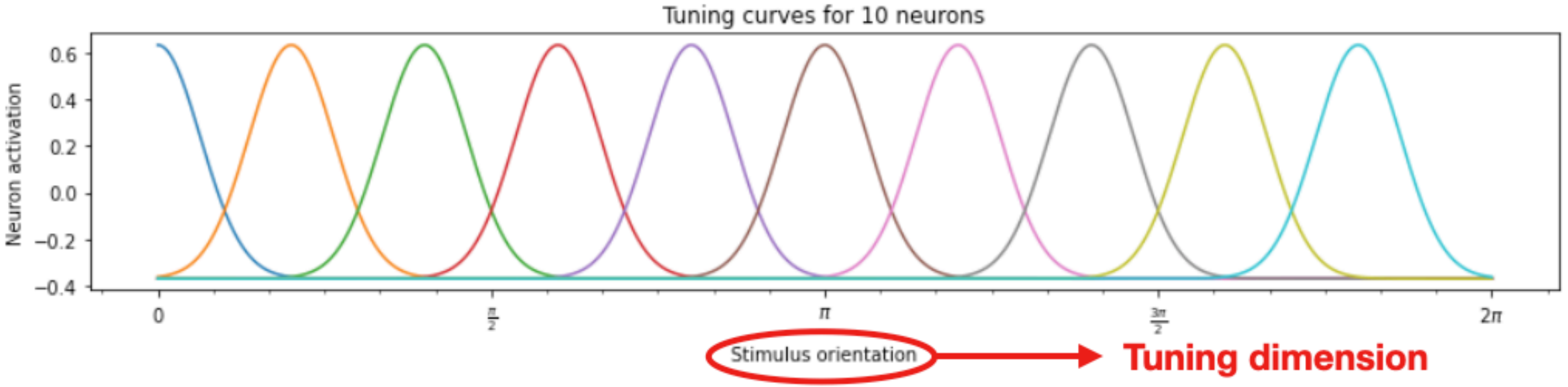
Figure 7: User study accuracy for each AlexNet layer. Shaded regions indicate the standard error across 22 study participants.

Discussion

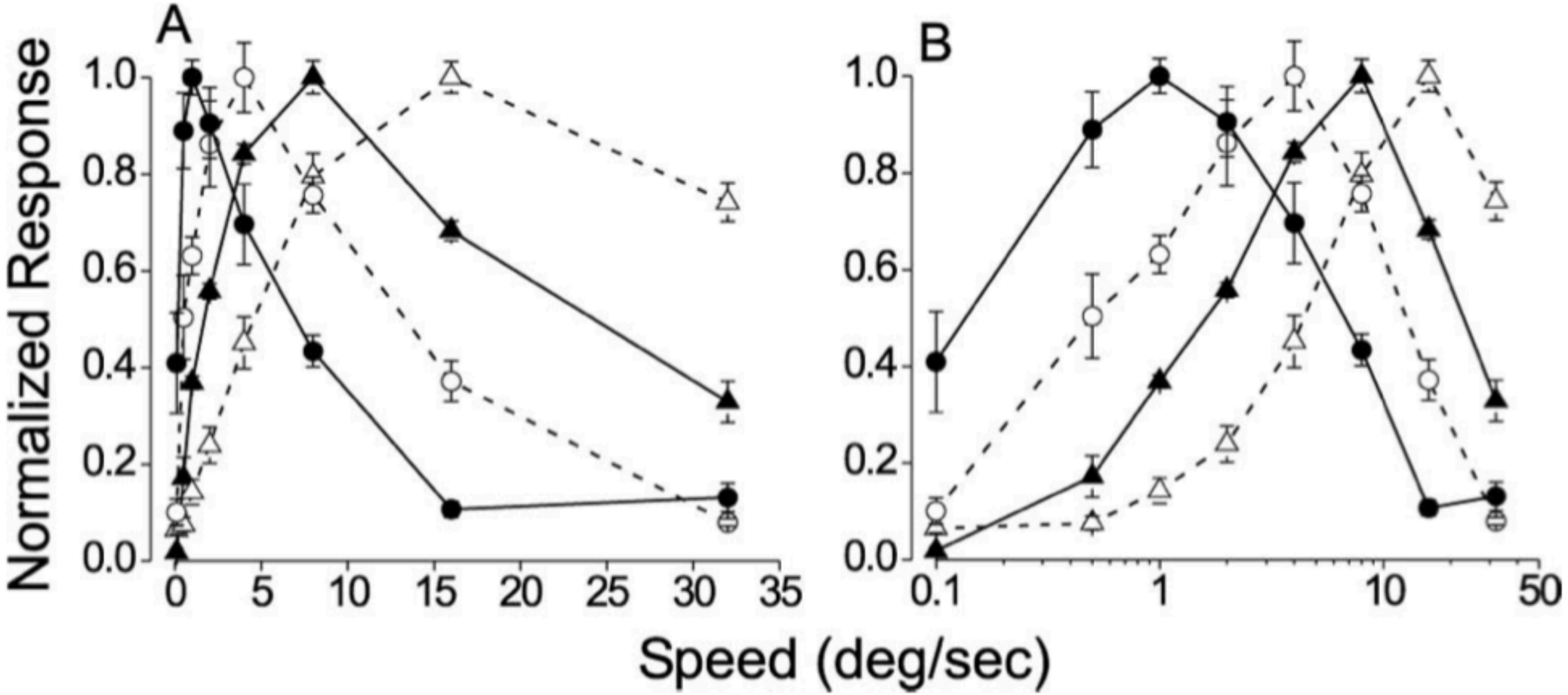
- **TL;DR:** The most important principal components provide more complete and interpretable explanations than the most important neurons.
- **Intended impact:** Motivate the community to think more carefully about the basis try to explain.
- **Limitations:** PCA offers a linear view of the nonlinear activation manifolds in NNs. PCA is far from the ideal decomposition.
- **Future work:** This motivates nonlinear dimensionality reduction such as SAEs which offer a scalable approach!

Extra slides

What is a tuning dimension?

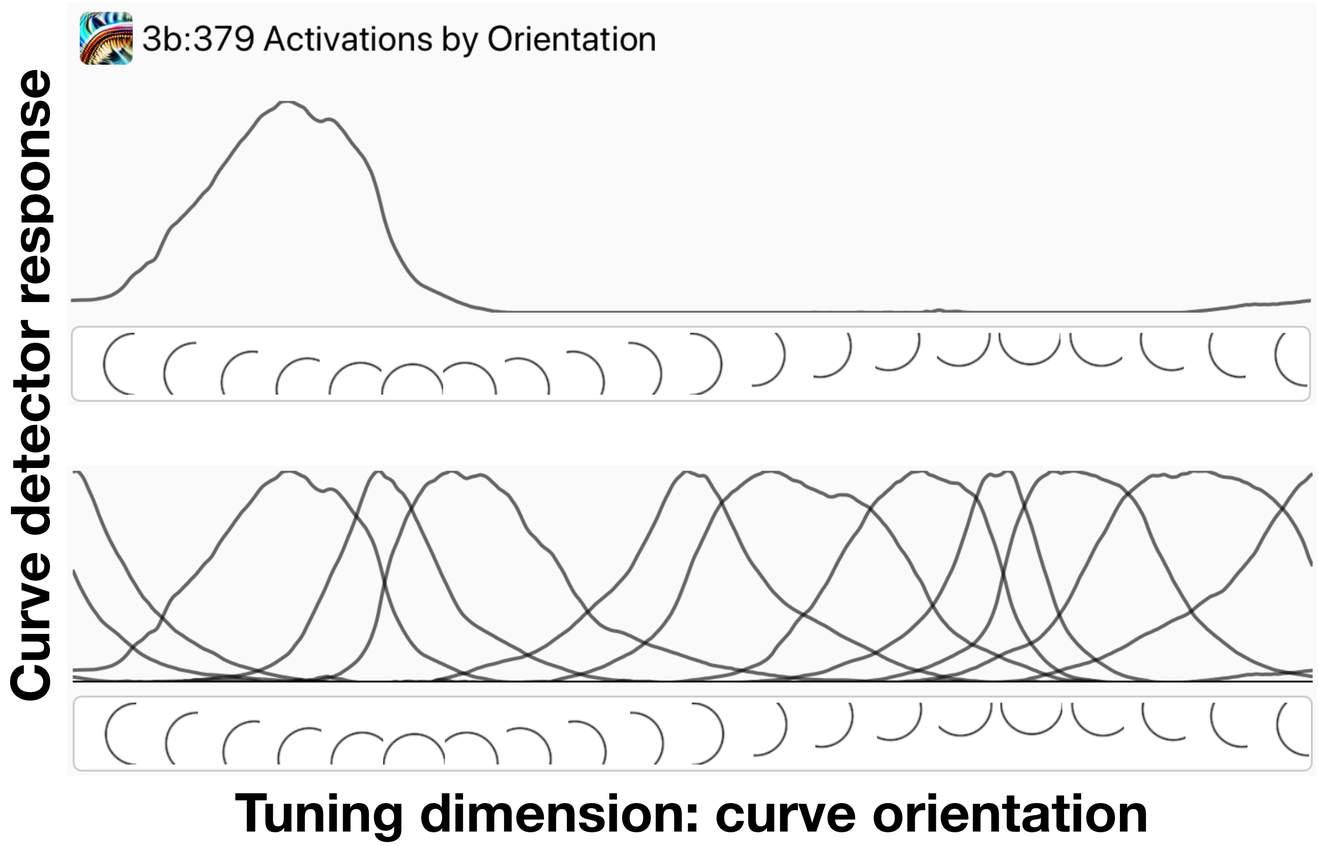


Why are tuning dimensions useful?



Harris Nover, Charles H. Anderson, and Gregory C. DeAngelis. A logarithmic, scale-invariant representation of speed in macaque middle temporal area accounts for speed discrimination performance. Journal of Neuroscience, 25(43):10049–10060, 2005. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1661-05.2005. URL https://www.jneurosci.org/content/25/ 43/10049.

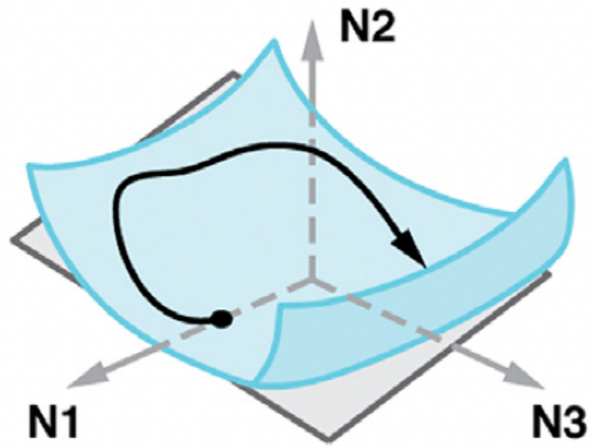
Manually identifying tuning dimensions in deep networks

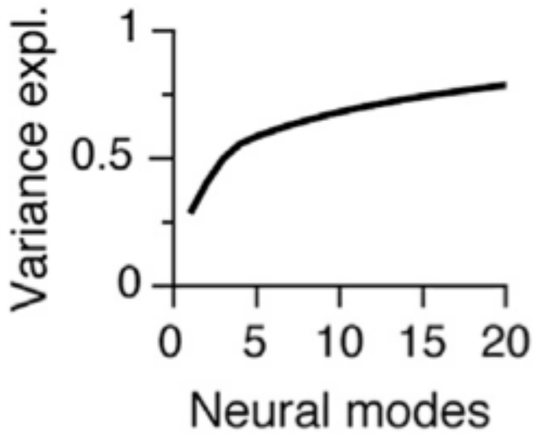


Nick Cammarata, Gabriel Goh, Shan Carter, Ludwig Schubert, Michael Petrov, and Chris Olah. Curve detectors. Distill, 2020. doi: 10.23915/distill.00024.003. https://distill.pub/2020/circuits/curve-detectors.

Neural Manifolds for the Control of Movement (Neuron 2017)

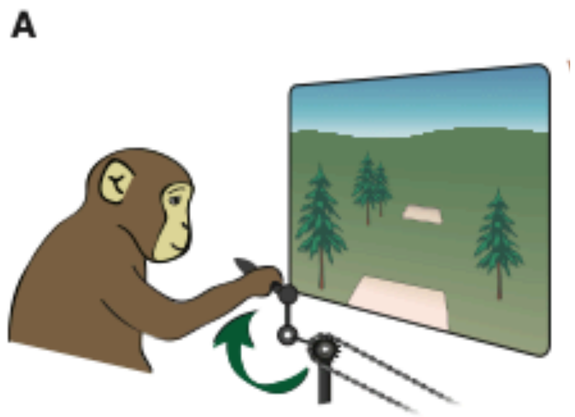
- using dimensionality reduction like PCA or factor analysis to find low-dimensional latent space for a neuron population
- interpreting each PC as a “neural mode” rather than a tuning dimension
- neural manifolds embody patterns of correlated activity
- evidence that network connectivity underlies the interactions among neurons captured by dimensionality reduction methods and the resulting neural modes.





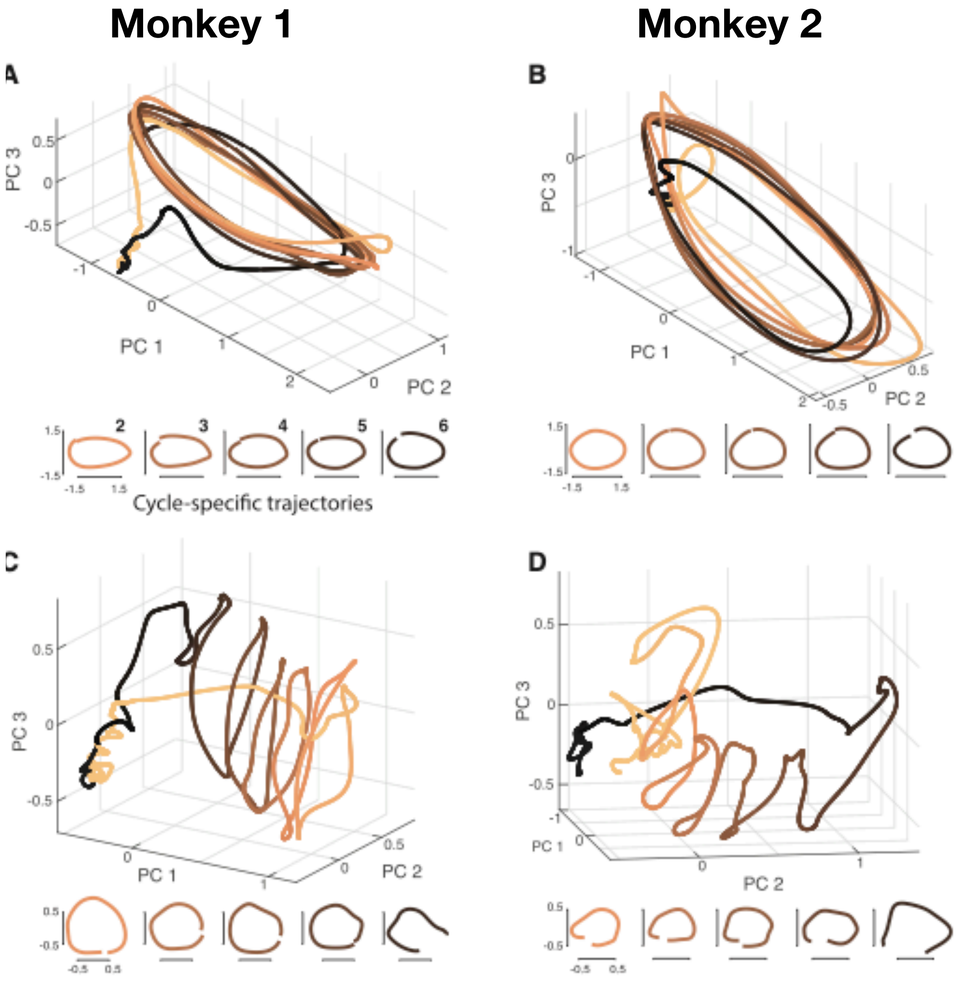
https://www.cell.com/neuron/pdf/S0896-6273(17)30463-4.pdf

Neural Trajectories in the Supplementary Motor Area and Motor Cortex Exhibit Distinct Geometries, Compatible with Different Classes of Computation - Neuron



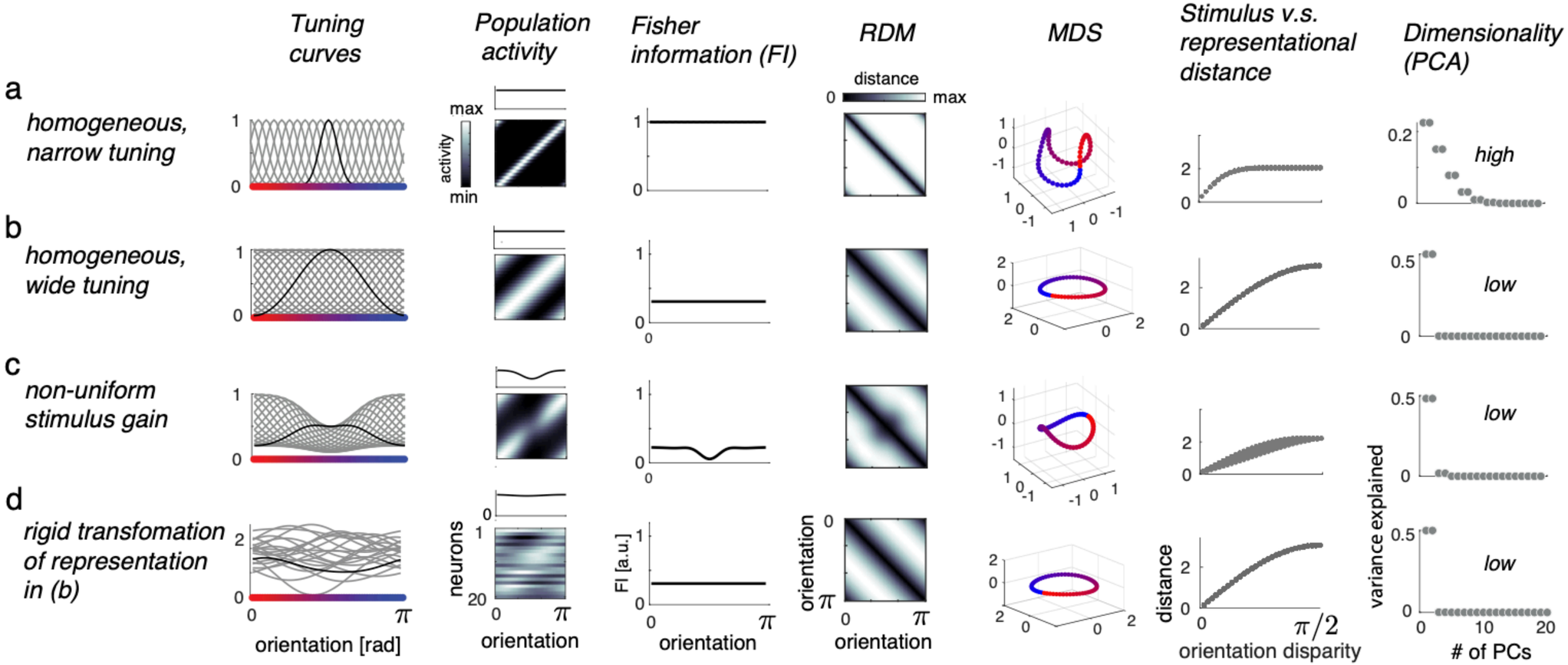
primary motor cortex (M1)

supplementary motor area (SMA)



http://www.columbia.edu/cu/neurotheory/Larry/RussoNeuron20.pdf

Neural tuning and representational geometry (Apr 20, 2021)



https://arxiv.org/pdf/2104.09743.pdf