GRAPHICAL CLUSTERABILITY AND LOCAL SPECIALIZATION IN DEEP NEURAL NETWORKS

Stephen Casper,[*1,4] **Shlomi Hod,**[*1,3] **Daniel Filan,**[*1,2]
Cody Wild, [1] **Andrew Critch,** [1,2] **Stuart Russell** [1,2]
[1]Center for Human-Compatible AI (CHAI)
[2]University of California Berkeley
[3]Boston University
[4]MIT Computer Science and Artificial Intelligence Laboratory (CSAIL)
$*$ Equal contribution
scasper@csail.mit.edu shlomi@bu.edu daniel_filan@berkeley.edu

ABSTRACT

The learned weights of deep neural networks have often been considered devoid of scrutable internal structure, and tools for studying them have not traditionally relied on techniques from network science. In this paper, we present methods for studying structure among a network's neurons by clustering them and for quantifying how well this reveals both graphical clusterability and *local specialization* – the degree to which the network can be understood as having distinct, highly internally connected subsets of neurons that perform subtasks. We offer a pipeline for this analysis consisting of methods for (1) representing a network as a graph, (2) clustering that graph, and (3) performing statistical analysis to determine how graphically clusterable and (4) functionally specialized the clusters are. We demonstrate that image classification networks up to the ImageNet-scale are often highly clusterable and locally specialized.[1]

1 INTRODUCTION

In science, systems are frequently understood by taking them apart and analyzing their components individually. The ability for such a system to have distinct, abstractable components has benefits including intelligibility and adaptivity (Clune et al., 2013; Baldwin & Clark, 2000; Booch et al., 2007). Modern deep neural networks are composed of neurons arranged in layers. Both of these levels of abstraction have been useful for studying networks. For example, individual neurons in computer vision models are frequently understood as feature-detectors (Mu & Andreas, 2020) and the layers of a network form progressively higher-level representations (Olah et al., 2017). But aside from these natural building blocks, can networks be studied at a more flexible level of abstraction?

Here, we aim to quantify how well deep networks can be divided into distinct subsets of neurons. This paper serves as a capstone for two of our preprints on graphical clusterability and local specialization in deep networks: Filan et al. (2021) and Hod et al. (2021). We present a set of tools for investigating how structurally decomposable deep networks are and how this translates to functional abstractability. These form a framework consisting of four steps: (1) *Graphification*: representing a network as a graph with nodes corresponding to neurons; (2) *Clustering*: dividing the nodes into subsets (clusters); (3) *Analysis of graphical clusterability*: quantifying how strong between- versus within-cluster weights are; and (4) *Analysis of local specialization*: using interpretability tools on the subsets to quantify how specialized they are.

We present a set of experiments which show that clustering in these networks often reveals a surprising amount of graphical structure and clusters that are likely to be abstractly characterizable. These tools require no human in the loop yet allow us to quantify *local specialization*, that is, how much a network's functionality can be abstracted into comprehensible sub-tasks localized to different groups of neurons. Ultimately, these methods can be used to automatedly screen for interesting sets of

[1]Code available at https://github.com/thestephencasper/local_specialization

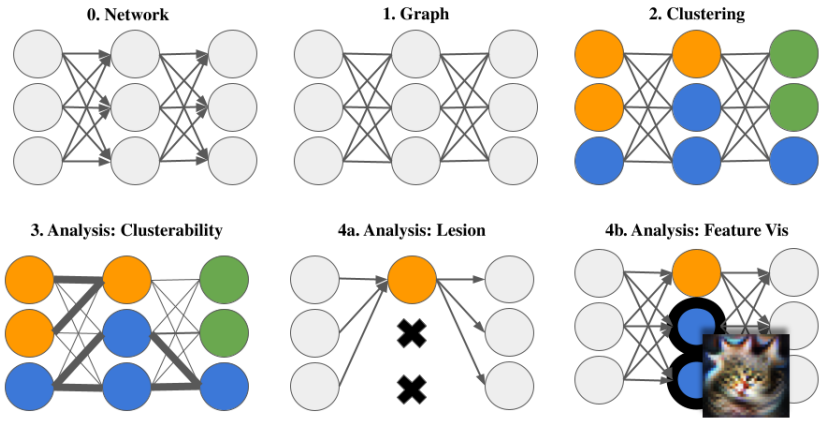


Figure 1: **Our procedural pipeline.** The first three steps generate a partitioning of the network into "clusters" which we analyze for (3) graphical clusterability and (4) local specialization using (4a) lesion and (4b) feature visualization methods. See Figure 4 for further detail.

neurons, and our results suggest advantages to understanding neural networks in terms of distinct neural subsets performing different functions

2 METHODS

Figure 1 outlines our pipeline. To evaluate graphical clusterability and local specialization of (0) a trained network, our procedure is to (1) "graphify" the network treating each neuron as a node; (2) perform spectral clustering on the graph to obtain a partitioning or "clustering" of neurons which we further divide by layer to obtain a "subclustering"; (3) calculate a measure of how clusterable the network is, and compare it to that of versions of the network with shuffled weights; and (4) use proxies for local specialization to analyze the subclusters and perform statistical analysis on the results, comparing true subclusters to random subsets of neurons.

2.1 GRAPHIFICATION AND CLUSTERING

Graphification ({weights, activations} $\times$ {layer-wise, network-wide}): Our goal is to represent a network as a graph with weights reflecting association between nodes. In our approach, units in MLPs (multilayer perceptrons) and channels in CNNs (convolutional neural networks) are nodes. We experiment with two ways of assigning edges: with weights and with correlations between activations. Both result in nonnegative weights which is required by the spectral clustering method we apply next. For weight-based graphification, if two nodes have a weight (in MLPs) or weights (in CNNs) connecting them in the network, their nodes are connected by an edge weighted with the L_1 norm of the weights. If layers are connected with a batch normalization layer in between, we mimic the scaling, multiplying weights by $\gamma/\sqrt{\sigma^2+\varepsilon}$ where γ is the scaling factor, σ^2 is the moving variance, and ε is a small constant. Importantly, this method requires no runtime analysis of the network. For correlation-based graphification, we connect the nodes for two neurons with the squared Spearman correlation between their pre-ReLU activations across a validation set (we take the L_1 norm of the activation for channels in CNNs).

We also test two scopes with which to construct graphs: network-wide and layer-wise. For network-wide graphification, we create one graph for the network as a whole. For layer-wise graphification, we produce a graph for each layer individually by only considering its connections to adjacent layers.

Spectral Clustering: We perform normalized spectral clustering (Shi & Malik, 2000) on the resulting graphs to partition the neurons into clusters. The algorithm is given in Appendix A.1. Layers at different depths of a network develop different representations, so for experiments where we measure

local specialization, we look at one layer at a time, even for network-wide clusterings in which clusters span more than one layer. We call these sets of neurons in the same cluster and layer *subclusters*. In local specialization experiments, we compare these subclusters to other random sets of neurons of the same size and layer. We refer to subclusters identified by the clustering algorithm as "true subclusters" and sets of random neurons as "random subclusters."

2.2 Measuring Graphical Clusterability

We measure the *absolute clusterability* of a graph by calculating its normalized cut or "n-cut" which is defined for a set of clusters $X_1, \ldots, X_k$ as $\text{n-cut}(X_1, \ldots, X_k) := \sum_{i=1}^{k} W(X_i, \bar{X}_i)/V(X_i)$ where W gives the sum of weight magnitudes between two clusters and V gives the sum of all weights incident to neurons in the cluster. We measure the *relative clusterability* of a network by comparing its n-cut to networks with the same set of weights in each layer but shuffled randomly, in order to determine whether any absolute clusterability is simply due to each layer's distribution of weights. The relative clusterability is quantified by the z-score of the neural network's n-cut when compared to the n-cuts of weight-shuffled versions of the network.

2.3 Measuring Local Specialization

Given a measure of how networks can be graphically clusterable, we next connect the graphical partitioning to localized functional specialization.

Importance and coherence: Our definition of local specialization requires comprehensible sub-tasks to be localized to subsets of neurons. To measure how well subsets can be abstractly characterized, we introduce two proxies: *importance* and *coherence*. Importance refers to how crucial a subcluster is to the network's performance overall. Coherence refers to how consistently the neurons in a subcluster activate in response to particular features in data. We assess importance and coherence using two methods: *lesions* and *feature visualization*.

Lesions: One approach (e.g., Gazzaniga & Ivry (2013); Casper et al. (2020); Ghorbani & Zou (2020)) for studying biological and artificial neural systems involves disrupting neurons during inference. We experiment with "lesion" tests in which we analyze network performance on the test set when a subcluster is dropped out. First, we measure importance by the drop in overall accuracy. Second, we measure coherence with respect to class by the range of class-specific accuracy drops: the larger the range, the more the lesioned cluster is crucial for some classes over others. See Appendix A.2.1 for additional details.

Feature visualization: To further analyze coherence, we use feature visualization (Olah et al., 2017). We optimize an input image to maximize the L_1 norm of the pre-ReLU activations of a subcluster. Properties of these visualizations can suggest what role a subcluster plays, and we use two techniques to analyze coherence using them. First, we analyze the value of the maximization objective for each visualization which we call the "score." This gives one measure of how coherent a subcluster may be with respect to input features. Another measure of coherence is the entropy of the softmax outputs of the network when these visualizations are passed through. If the entropy is low, this suggests that a cluster is coherent with respect to class labels. See Appendix A.2.2 for additional details.

Statistical analysis: We measure how important and coherent subclusters are compared to random ones of the same size and layer. For each subcluster, we calculate a measure of importance or coherence and compare it to those of random subclusters to obtain a percentile. We then aggregate these percentiles across a network into a *Fisher statistic*. A higher Fisher statistic indicates a more significant result. Details on how this and an associated p value are calculated are in Appendix A.2.3. We also perform a multiple correction using the Benjamini Hochberg method.

3 Results

3.1 Graphical Clusterability

ImageNet networks are clusterable. To measure how well our clusterings capture graphical structure in the network, here we focus on weight-based, network-wide clusterings. We find that networks at

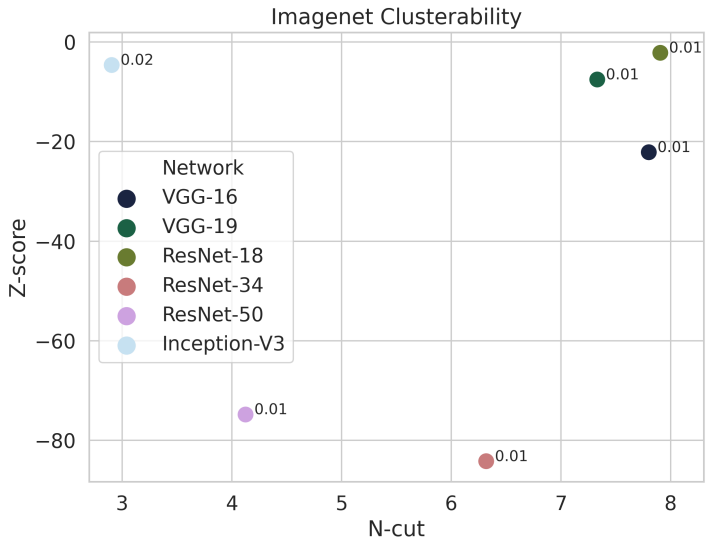


Figure 2: ImageNet classifiers are highly clusterable compared to versions of themselves with shuffled weights. N-cuts measure absolute clusterability and z-scores measure it relative to 100 weight-shuffled networks. Lower values on both axes means more clusterability. Points are labeled with their one-sided p-value based on the comparisons to shuffled networks.

the ImageNet scale are consistently more clusterable than weight-shuffled versions of themselves. Figure 2 shows the n-cuts and z-scores of 6 different ImageNet classifiers.

3.2 Local Specialization

Table 1 gives Fisher statistics for all four types of graphification methods and all four measures of local specialization across diversity of of networks. Values that are statistically significantly large are bolded. We highlight three key findings.

Our clusterings identify important subclusters. Fisher statistics for lesion accuracy drops are high and significant, indicating that sub-clusters are more likely to be highly important relative to random groups of neurons.

Our clusterings identify subclusters that are coherent w.r.t. input features but not class label. Class-specific measures of coherence, (class-wise lesion accuracy drop range and output entropy) showed significant coherence in almost no conditions. However, subclusters were reliably coherent as measured by *visualization score*. This suggests that subclusters tended to perform coherent sub-tasks, but not in a class-specific way.

All clustering methods yield similar results. In Table 1, we find no clear difference between the Fisher statistics of activation-based and weight-based clusterings, or between layer-wise and network-wide clusterings. This is somewhat unexpected: one might have predicted that weight-based methods' lack of runtime information or layer-wise methods' lack of global information would lead to lower quality clusterings, but this was not the case.

4 Related Work

The line of work uses methods for clustering from network science (Girvan & Newman, 2002; Newman & Girvan, 2004; Shi & Malik, 2000; von Luxburg, 2007). Importantly, any method of representing and partitioning neurons could be used for graphification and clustering. For our analysis, we use lesions (Zhou et al., 2018) and feature visualization (Olah et al., 2017), but other interpretability techniques (e.g., (Morcos et al., 2018; Madan et al., 2020; Bau et al., 2017; Testolin

Table 1: Fisher statistics for (1) lesion-based experiments measuring importance via overall accuracy drops (Acc. Drop) and coherence via the class-wise range of accuracy drops (Class Range); and (2) Feature visualization-based experiments in networks measuring coherence via the optimization score (Vis Score) and the entropy of network outputs (Softmax H). Each row corresponds to a network paired with a partitioning method. Values that are statistically significantly high are **bold**.

		Lesion		Feature Visualization	
Network	**Partitioning**	**Acc. Drop**	**Class Range**	**Vis Score**	**Softmax H**
MLP, MNIST	Weight/Network	**2.13**	0.91	**1.32**	0.92
	Weight/Layer	**2.05**	0.91	**1.21**	**1.23**
	Act./Network	**1.46**	1.00	**1.34**	**1.15**
	Act./Layer	**1.69**	1.03	**1.36**	**1.12**
CNN, MNIST	Weight/Network	**1.29**	0.84	1.10	0.93
	Weight/Layer	1.10	0.94	1.02	0.99
	Act./Network	**1.73**	0.70	1.09	0.90
	Act./Layer	**1.46**	0.92	1.05	0.98
VGG, CIFAR-10	Weight/Network	**1.50**	**2.12**	**1.46**	0.99
	Weight/Layer	**1.15**	**1.27**	1.00	0.97
	Act./Network	**1.40**	0.97	**2.34**	1.08
	Act./Layer	**1.56**	1.03	**2.67**	**1.12**
VGG-16, ImageNet	Weight/Network	**2.54**	0.49	**1.72**	1.19
	Weight/Layer	**2.15**	0.56	**1.90**	1.06
	Act./Network	**1.89**	0.63	**1.82**	1.07
	Act./Layer	**1.66**	0.70	**1.85**	0.98
VGG-19, ImageNet	Weight/Network			**1.91**	1.03
	Weight/Layer			**2.23**	1.00
	Act./Network			**1.87**	1.10
	Act./Layer			**2.01**	0.98
ResNet18, ImageNet	Weight/Network	**1.42**	1.13		
	Weight/Layer	**1.29**	0.99		
	Act./Network	**1.30**	0.92		
	Act./Layer	**1.31**	0.96		

et al., 2020; Panzeri et al., 2017)) could also be used in a similar way. We add to a body of research focused on modularity and compositionality in neural systems (You et al., 2020; Mu & Andreas, 2020; Voss et al., 2021; Lake et al., 2015; Csordás et al., 2021; Udrescu et al., 2020).

5 Discussion

We introduce several methods for clustering the neurons of a network and rigorously analyzing these clusters for graphical clusterability and local specialization. To the best of our knowledge, we provide the first methods to quantitatively assess local specialization without a human in the loop. Having effective tools for interpreting networks is important for understanding AI systems, in particular by helping to diagnose failure modes (e.g., Carter et al. (2019); Mu & Andreas (2020); Casper et al. (2021)). Our work relates to this goal, though indirectly. Rather than directly using interpretability tools to explain clusters, our focus is one step back: on automatedly testing whether they are worth analyzing at all. These results show that clustering can reveal structurally and functionally distinct sets of neurons which suggests a useful level of abstraction with which to study networks.

This approach has limitations including a lack of assurance that importance and coherence are reliably strong proxies for human-comprehensible descriptions and that we do not identify subtasks performed by clusters of neurons. Accordingly, our tools should be seen as methods for screening a network for sets of associated neurons where sub-task functionality is localized. Despite ours and related work, neural systems are still complex, and more insights are needed to develop useful understandings of them. The ultimate goal should be to develop reliable methods for building effective models which lend themselves to faithful interpretations. We believe this approach should involve building and analyzing models under the principles of graphical clusterability and local specialization.

REFERENCES

Carliss Young Baldwin and Kim B Clark. *Design rules: The power of modularity*, volume 1. MIT press, 2000.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

Grady Booch, Robert A Maksimchuk, Michael W Engle, Bobbi Young, Jim Conallen, and Kelli A Houston. *Object-Oriented Analysis and Design with Applications*. Addison-Wesley Professional, third edition, 2007.

Alfio Borzì and Giuseppe Borzì. Algebraic multigrid methods for solving generalized eigenvalue problems. *International journal for numerical methods in engineering*, 65(8):1186–1196, 2006.

Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 4(3):e15, 2019.

Stephen Casper, Xavier Boix, Vanessa D'Amario, Ling Guo, Kasper Vinken, and Gabriel Kreiman. Frivolous units: Wider networks are not really that wide. *arXiv preprint arXiv:1912.04783*, 2020.

Stephen Casper, Max Nadeau, and Gabriel Kreiman. One thing to fool them all: Generating interpretable, universal, and physically-realizable adversarial features. *arXiv preprint arXiv:2110.03605*, 2021.

Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society B: Biological sciences*, 280(1755), 2013.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Are neural nets modular? inspecting their functionality through differentiable weight masks. In *International Conference on Learning Representations*, 2021.

Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. *arXiv preprint arXiv:2103.03386*, 2021.

Michael Gazzaniga and Richard B Ivry. *Cognitive Neuroscience: The Biology of the Mind: Fourth International Student Edition*. WW Norton, 2013.

Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in Neural Information Processing Systems*, 33:5922–5932, 2020.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Shlomi Hod, Stephen Casper, Daniel Filan, Cody Wild, Andrew Critch, and Stuart Russell. Quantifying local specialization in deep neural networks. *CoRR*, abs/2110.08058, 2021. URL `https://arxiv.org/abs/2110.08058`.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Spandan Madan, Timothy Henry, Jamell Dozier, Helen Ho, Nishchal Bhandari, Tomotake Sasaki, Frédo Durand, Hanspeter Pfister, and Xavier Boix. On the capability of neural networks to generalize to unseen category-pose combinations. *arXiv preprint arXiv:2007.08032*, 2020.

Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *arXiv preprint arXiv:2006.14032*, 2020.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

Stefano Panzeri, Christopher D Harvey, Eugenio Piasini, Peter E Latham, and Tommaso Fellin. Cracking the neural code for sensory perception by combining statistics, intervention, and behavior. *Neuron*, 93(3): 491–507, 2017.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Alberto Testolin, Michele Piccolini, and Samir Suweis. Deep learning systems as complex networks. *Journal of Complex Networks*, 8(1):cnz018, 2020.

Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *arXiv preprint arXiv:2006.10782*, 2020.

Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Chelsea Voss, Gabriel Goh, Nick Cammarata, Michael Petrov, Ludwig Schubert, and Chris Olah. Branch specialization. *Distill*, 6(4):e00024–008, 2021.

Jiaxuan You, Jure Leskovec, Kaiming He, and Saining Xie. Graph structure of neural networks. *arXiv preprint arXiv:2007.06559*, 2020.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in CNNs via ablation. *arXiv preprint arXiv:1806.02891*, 2018.

A APPENDIX

A.1 SPECTRAL CLUSTERING ALGORITHM

The spectral clustering algorithm on the graph $G = (V, E)$ produces a partition of its vertices in which there are stronger connections within sets of vertices than between them (Shi & Malik, 2000). It does so by solving a relaxation of the NP-Hard problem of minimizing the ***n-cut*** (normalized cut) for a partition. For disjoint, non-empty sets $X_1, ...X_k$ where $\cup_{i=1}^{k} X_i = V$, this is defined by von Luxburg (2007) as:

$$\text{n-cut}(X_1, ..., X_k) \coloneqq \frac{1}{2} \sum_{i=1}^{k} \frac{W(X_i, \overline{X_i})}{\text{vol}(X_i)}$$

for two sets of vertices $X, Y \subseteq V$, we define $W(X, Y) \coloneqq \sum_{v_i \in X, v_j \in Y} w_{ij}$; the degree of a vertex $v_i \in V$ is $d_i = \sum_{j=1}^{n} w_{ij}$; and the volume of a subset $X \subseteq V$ is $\text{vol}(X) \coloneqq \sum_{i \in X} d_i$.

We use the *scikit-learn* implementation (Pedregosa et al., 2011) with the ARPACK eigenvalue solver (Borzì & Borzì, 2006).

Algorithm 1: Normalized spectral clustering according to Shi & Malik (2000), implemented in *scikit-learn* (Pedregosa et al., 2011), description taken from von Luxburg (2007).

Input : Weighted adjacency matrix $W \in \mathbb{R}^{n \times n}$, number k of clusters to construct

1 Compute the unnormalized Laplacian L.

2 Compute the first k generalized eigenvectors $u_1, ..., u_k$ of the generalized eigenproblem $Lu = \lambda Du$.

3 Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, ..., u_k$ as columns.

4 For $i = 1, .., n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i^{th} row of U.

5 Cluster the points $(y_i)_{i=1,...,n}$ in $\mathbb{R}^k$ with the ***k-means*** algorithm into clusters $C_1, ..., C_k$,

Output : Clusters $A_1, ..., A_k$ with $A_i = \{j | y_j \in C_i\}$.

A.2 ADDITIONAL DETAILS FOR LOCAL SPECIALIZATION

A.2.1 LESION TEST

Let θ be the parameter vector of the neural network, c be a set of neurons, $\mathcal{M}(\theta, c)$ be a masked version of θ where weights into or out of nodes in c have been set to 0, and $\text{Acc}(\vartheta, \mathcal{D})$ be the accuracy of the network parameterized by ϑ on dataset $\mathcal{D}$.

Importance: accuracy drop

Then, our measure for importance is $\text{Acc}(\theta, \texttt{test}) - \text{Acc}(\mathcal{M}(\theta, c), \texttt{test})$, where `test` is a test dataset that was not used to construct the activation-based partitionings.

Coherence: class range

Let $\texttt{test}_i$ be the subset of the test set with label i, and let $\Delta(\theta, c, i) := \text{Acc}(\theta, \texttt{test}_i) - \text{Acc}(\mathcal{M}(\theta, c), \texttt{test}_i)$ be the drop in accuracy for examples with label i from lesioning c. Then, this measure of coherence is the range $(\max_i \Delta(\theta, c, i) - \min_i \Delta(\theta, c, i))$, of accuracy drops over classes. This detects whether clusters are more crucial for some classes over others, which would suggest that they coherently act to correctly label those classes.

A.2.2 FEATURE VISUALIZATION

Letting the parameter vector be θ and the subcluster be c, we write $\text{Act}(x, \theta, c)$ for the vector of pre-ReLU activations of neurons in c in network θ on input x, and denote this optimized input image as $x(\theta, c)$, which approximately maximizes $\|\text{Act}(x, \theta, c)\|_1$.

Coherence: vis score The score $\|\text{Act}(x(\theta, c), \theta, c)\|_1$ gives one measure of how coherent a subcluster may be with respect to input features.

Coherence: entropy Another measure of coherence is the entropy $H(\text{label} \mid x(\theta, c); \theta)$ of the softmax outputs of the network when these visualizations are passed through. If the entropy is low, this suggests that a cluster is coherent with respect to class labels.

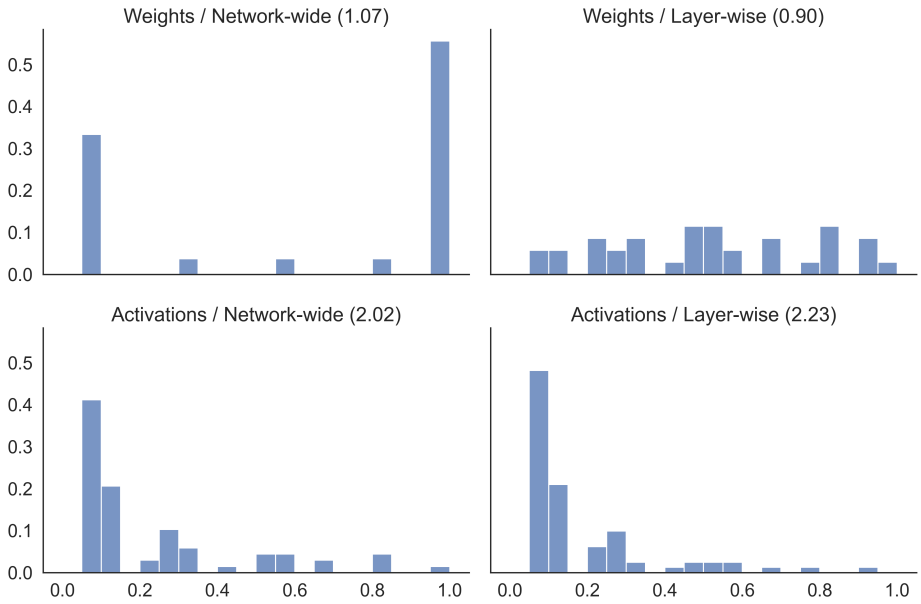
A.2.3 Statistical Pipeline

Figure 4 outlines this overall approach.

Fisher statistics: We wish to test whether spectral clustering methods find subsets of neurons which satisfy our proxies for local specialization more than if we had simply chosen random subsets. For each subcluster measurement we take the percentile of each true subcluster relative to the distribution of measurements of 19 random subclusters. Next, we use the Fisher method to test whether the subclusters in a network satisfy our proxies more than random subsets of neurons. To do so, we first center the subcluster percentiles around 0.5, which under the null hypothesis would give a granular, unbiased approximation of the uniform distribution. We then combine the centered percentiles $\{p_1, \ldots, p_n\}$ into the Fisher statistic $(-1/n) \sum_{i=1}^{n} \log p_i$. For reference, the Fisher statistic of a uniform distribution of percentiles in our setting is 0.98. A higher value gives evidence of subclusters that exhibit local specialization. Figure 3 shows example distributions of percentiles and their Fisher statistics. Note also that since the $\log$ function has a larger derivative near 0 than near 1, low percentiles have greater influence on the Fisher statistic, so J-shaped distributions can also have Fisher statistics greater than 1. For all non-ImageNet architectures, we train and analyze 5 networks per condition and report the mean Fisher statistic.

The Fisher statistic on n uniformly-distributed random percentiles multiplied by $2n$ takes a chi-squared distribution with $2n$ degrees of freedom. This lets us produce a p value for each network, testing whether this statistic is higher than the null hypothesis would produce. The fact that we coarsely measure percentiles and then center them makes this test conservative because our statistic

Figure 3: **Illustration of Fisher statistics of various percentile distributions.** A VGG network trained on CIFAR-10 is partitioned using four methods ($\{\text{weights, activations}\} \times \{\text{network-wide, layer-wise}\}$) and analyzed for coherence to produce the collection of percentiles for each subcluster. This shows histograms of the percentile distribution for each clustering, and their associated Fisher statistics. Recall that a lower percentile means that a true subcluster is more coherent than random subclusters while controlling for layer and size. The activation-based clusterings have disproportionately many low percentiles, and Fisher statistics greater than 2. Table 1 shows that these trends are statistically significant when aggregated over five models.



is more sensitive to low percentiles than high percentiles. This procedure is illustrated in Appendix Figure 4. For all non-ImageNet networks, we train and analyze 5 different versions of each, take the mean p value, and correct it by taking the corresponding quantile of a Bates$(n = 5)$ distribution which gives the distribution of the mean of 5 independent uniformly-distributed random variables. This produces a p value for every network architecture, partitioning method, and local specialization proxy. We then correct for multiple testing using the Benjamini Hochberg method (Benjamini & Hochberg, 1995), controlling the false discovery rate at 0.05.

Figure 4: **Our extended procedural pipeline.** This figure expands Figure 1 and shows the successive steps after generating a partitioning of clusters (step 2 in Figure 1). After performing either lesion or feature visualization analysis, the results from each true subcluster and its random subclusters are aggregated to produce p values. For simplicity, only the analysis for the lesion experiment is presented, but the same pipeline is used for the feature visualization experiments.