CAUSAL INTERPRETATION OF NEURAL NETWORK COMPUTATIONS WITH CONTRIBUTION DECOMPOSITION (CODEC)

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding how neural networks transform inputs into outputs is crucial for interpreting and manipulating their behavior. Most existing approaches analyze internal representations by identifying hidden-layer activation patterns correlated with human-interpretable concepts. Here we take a direct approach to examine how hidden neurons act to drive network outputs. We introduce CODEC (**Co**ntribution **Dec**omposition), a method that uses sparse autoencoders to decompose network behavior into sparse motifs of hidden-neuron contributions, revealing causal processes that cannot be determined by analyzing activations alone. Applying CODEC to benchmark image-classification networks, we find that contributions grow in sparsity and dimensionality across layers and, unexpectedly, that they progressively decorrelate positive and negative effects on outputs. We further show that decomposing contributions into sparse modes enables greater control and interpretation of intermediate layers, supporting both causal manipulations of network output and human-interpretable visualizations of distinct image components that combine to drive that output. Finally, by analyzing state-of-the-art models of retinal activity, we demonstrate that CODEC uncovers combinatorial actions of model interneurons and identifies the sources of dynamic receptive fields. Overall, CODEC provides a rich and interpretable framework for understanding how nonlinear computations evolve across hierarchical layers, establishing contribution modes as an informative unit of analysis for mechanistic insights into artificial neural networks.

1 A FRAMEWORK FOR UNDERSTANDING BIOLOGICAL AND ARTIFICIAL NEURAL NETWORKS

Biological and artificial neural networks both produce computations using cascading nonlinear operations that do not lend themselves to simple interpretations. Despite the widespread study and use of neural networks, there is no standardized framework to understand how a given network output is generated from its input through its intermediate stages. Understanding the mechanisms by which networks behave promises to accelerate studies of the nervous system, lead to more effective design of efficient networks, reveal general principles of information processing in complex systems, and is also important for guiding the development of safe AI systems (Murdoch et al. (2019); Doshi-Velez & Kim (2017); Lipton (2017); Rudin (2019)).

An essential aspect of both artificial and biological neural networks is that their behavior is created by hierarchical sets of internal components. The question we approach here is: *How do the components of a network act to construct the output from the input?*

KEY PRINCIPLES FROM A CENTURY OF STUDIES OF THE NERVOUS SYSTEM

The historical roots of modern ANNs lie in the attempt to model the computational properties of biological neurons (McCulloch & Pitts (1990)). We highlight three key concepts that have been applied to the characterizations of biological neural circuits:

1. Neural coding: Considering a neuron as an intermediary between stimulus and behavior, the action of a neuron is a combination of two computational stages: the effect of the stimulus on the neuron, its *receptive field*, and the effect of the neuron on the downstream network or behavior, its *projection field* (Lehky & Sejnowski (1988)). **2. Cell-type specialization**: Neural function in the brain relies on over 5,000 diverse cell types with unique genetic, anatomical, and computational traits (Yao et al. (2023)). Whereas architectures like CNNs use implicitly contain cell-type specialization in the form of channels, there has been little attempt to analyze the computations therein using cell-type (channel) as a basis.

3. Population coding: Single-neuron analysis is limited due to redundancy, synergy, cancellation and interactions in neural circuits (Edelman & Gally (2001)). Understanding the synthesis of a computation requires a description of how all neural contributions are used to produce the given output (Olshausen & Field (2004)).

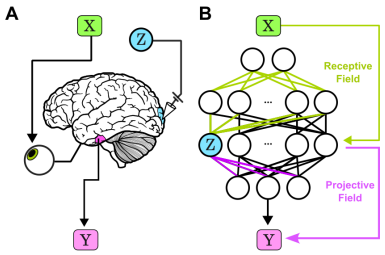

Figure 1: **Understanding the contribution of an intermediate component to downstream computation** (A) Biological and (B) artificial neural circuits construct computations by combining sets of upstream components in an input-dependent manner. The action of a network component Z is a composition of its receptive field, or sensitivity to input, X and its its projective field, or effect on output, Y . Measuring both is required to explain how the intermediate component contributes to the overall behavior of the system.

Like cell types in biology, hidden units in neural networks encode features whose meaning comes from how they shape downstream circuits. Understanding a network thus requires not just identifying features in the internal representation, but also explaining how combinations of those features are used to construct different outputs.

Existing tools for interpreting ANNs

For ANNs, a key challenge is identifying meaningful units within nonlinear systems of millions of parameters, as computational intermediates are poorly defined. In both biological and artificial networks, much attention has been devoted to studying the latent representations at intermediate stages in the network by analyzing the network activity. Although these methods have found patterns in activations via clustering or sparse autoencoders (Fel et al. (2023)), such approaches to analyzing representations fundamentally do not address the causal question of how internal elements act to influence the output.

More recently, mechanistic interpretability methods like Integrated Gradients, SmoothGrad, and Grad-CAM focused on input attribution through saliency maps, highlighting influential features (Selvaraju et al. (2020); Smilkov et al. (2017); Sundararajan et al. (2017)). However, they offer limited insight into how intermediate representations contribute to computations. Component visual features such as edges and textures may be needed to discriminate an object, but those same visual features in other areas of an image may not be related to the target object yet still represented in the intermediate activations. Thus, compared to the analysis of activations, analysis of contributions distinguishes between building blocks of computation that causally drive the output, and those that are irrelevant. Thus, a key gap is understanding how networks integrate distributed latent features across channels and neurons to generate outputs, similar to how biological cells derive functional effects from circuit interactions, rather than just representing the input.

A neuroscience-inspired ANN interpretability framework

We introduce a method for analyzing how intermediate neurons contribute to a network's output. First, using attribution techniques like Integrated Gradients on internal layers, we compute each hidden neuron's contributions across all stimuli, capturing the combined effects of their receptive and projective fields. These contributions represent the actions neurons take to construct the output.

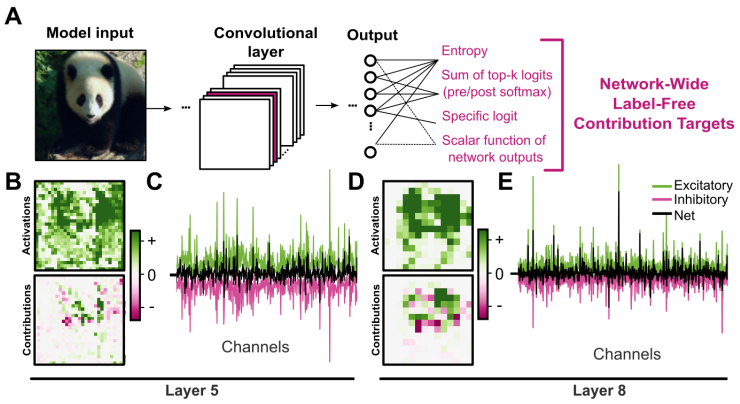

Figure 2: **Hidden-neuron contributions in a deep convolutional network.** (A) Pipeline of computing contributions for an image processed through ResNet-50. Single contribution targets including entropy, sum of top-k logits, and individual class logits are used as scalar objectives to compute gradient-based attribution. (B) Spatial map of activations and contributions of a single channel in Layer 5. (C) Mean positive, negative and net contribution for each channel. (D - E) Same as (B - C) for Layer 8.

Next, we decompose these contributions across inputs into a set of modes reflecting coordinated neuronal actions, an approach we call **contribution decomposition (CODEC)**. Unlike prior methods that analyze activations, CODEC directly captures causal effects on outputs and can be applied to any trained feedforward model without access to training data or labels, unlike methods such as the Average Gradient Outer Product (Radhakrishnan et al. (2024)). This general-purpose framework quantifies how groups of neurons drive behaviors. In the context of visual image recognition, the receptive fields of hidden neurons define visual features that are building blocks, and the contribution modes are the assembly instructions that show how those components are used to construct classes.

CODEC is a general framework composed of different stages that can be adapted for artificial and biological neural networks:

1. **Contribution target:** Identifying the specific output neuron or behavior (a scalar function of output neurons) whose computational basis we wish to understand.
2. **Contribution algorithm:** Quantifying how each hidden unit contributes to the target output for a given input.
3. **Decomposition of contributions:** Identifying the core computational modes and determining the patterns of how modes are combined across inputs and outputs.
4. **Visualization in input space**: Reveal how the key channels or neurons within each mode are being used for output identification by mapping contribution back to input space.

Using these methods, we introduce a new interpretability tool that examines the intermediate neurons in a network, identifies what input features they are sensitive to and their individual effects on network output, and reveals how their combined actions ultimately influence the network's behavior.

2 Measuring Contributions of Hidden-Layer Neurons

The contribution of a hidden neuron to network output is a composition of its overall input and its overall output (Fig 1) , and several methods have been used to calculate such effects. Integrated Gradients have most commonly been applied from network output to input Sundararajan et al. (2017), but have also been applied to analyze the effects of hidden neurons in models of biological networks (see Supp. material for derivation) (Tanaka et al. (2019); Maheswaranathan et al. (2023)). An alternative attribution method, ActGrad, is inspired by GradCAM (Selvaraju et al. (2020)) and naturally operates on hidden units. It is defined as the element-wise product of activations and gradients ($\text{ActGrad}_j = h_j \cdot \frac{\partial y}{\partial h_j}$) where h_j is the activation of hidden unit j (which could be the input). The

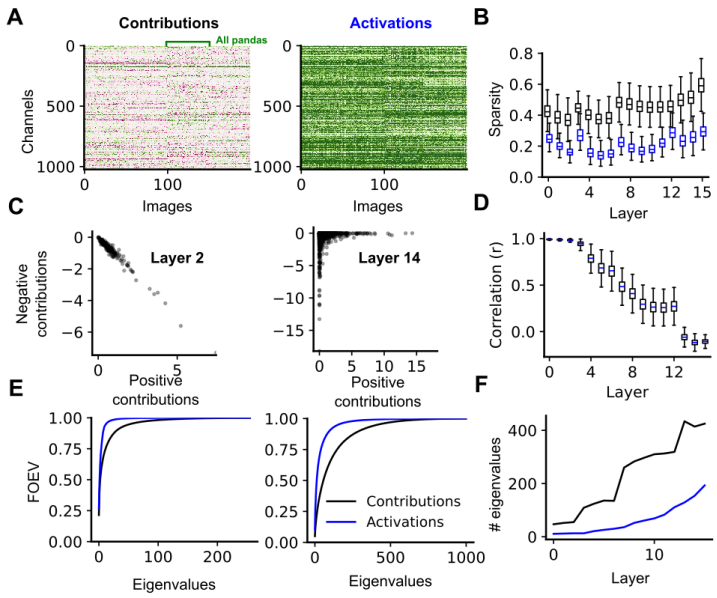

Figure 3: **Channel contributions through the network become more sparse, single-signed and high dimensional.** (A) Matrix of mean contributions and activations for all chanenls and images from four classes. (B) Hoyer sparsity index for contributions and activations across network layers. (B) Evolution of mean Hoyer sparsity across all network layers. (C) Scatter plots of positive and negative contribution magnitudes of each channel across the three layers, showing the relationship between the magnitude of positive and negative contributions. (D) Correlation coefficient between positive and negative contributions of individual channels across network depth. (E) Cumulative fraction of explained variance across all class-averaged channel weightings for layer 2 and layer 14 (F) Number of components required to reach 95 percent fraction of explained variance (FOEV).

key difference between ActGrad and GradCAM is that GradCAM performs a global average pool of the gradients over spatial dimensions before multiplying them with the activations, contracting the channel index and yielding a saliency map over space for an intermediate layer. ActGrad obtains the contribution for each hidden unit directly, preserving both the spatial and channel indices.

Formulating a single contribution target was crucial to avoid intractable 3D decompositions, which would occur if contributions were computed separately for each logit, producing an array of shape $n_{\text{inputs}} \times n_{\text{neurons}} \times n_{\text{logits}}$ for an intermediate layer. To efficiently capture the network's motifs, we use two network-wide objectives: (1) the sum of top-k output neurons, reflecting the model's confidence in its strongest predictions, and (2) the entropy of the output distribution, measuring prediction uncertainty.

Contributions of convolutional layers in image-classification networks

Biological visual systems, such as the retina, primary visual cortex, and inferotemporal cortex, share many architectural features with CNNs (Yamins et al. (2014)). Thus, we aimed to characterize the computational structure of each layer in benchmark CNNs such as ResNet-50 by analyzing neuronal contributions.

Starting with an input image, we compute the contributions of all hidden neurons to the sum of top-1 logits (Fig 2A), which indicates how hidden units drive the network's confidence in its strongest predictions. We applied different algorithms (ActGrad, Integrated Gradients, SmoothGrad) with target different outputs (e.g., class logits, top-k confidence, entropy), and found that contributions were consistently spatially sparse compared to activations (Fig 2B,D) (Supplementary Figure 1). A key property of Integrated Gradients is completeness: contributions sum to the scalar output target. Thus, spatially summing contributions within a channel gives its net effect on the prediction, enabling assignment of a single contribution value per channel, or cell type (Fig 2C). Applying

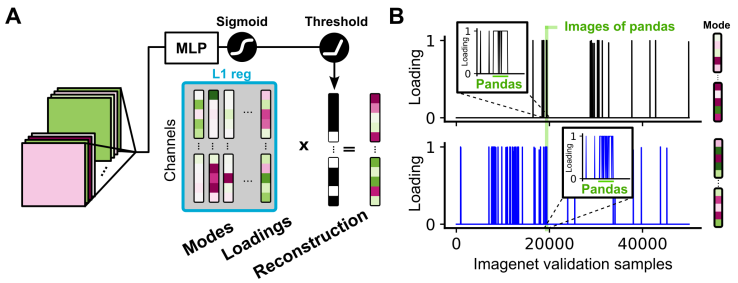

Figure 4: **Sparse autoencoder decomposition of network contributions.** (A) Schematic diagram of contribution decomposition. Channel contributions are spatially summed and an autoencoder is trained to reconstruct the matrix of channel contributions for each image. Loadings are passed through a sigmoid and thresholded and regularized to encourage sparsity. (B) Loadings from the mode that maximally correlated with the class "panda" for contributions (top, black) and activations (bottom, blue). Inset shows the loadings for 50 images of panda and 100 images of other classes.

this procedure across the entire 50,000 ImageNet validation dataset produced matrices of channel contributions for selected network blocks throughout ResNet-50, with each matrix having dimensionality d channels × 50,000 images, where d corresponds to the number of channels in each layer (Fig 1A). For the remainder of this paper, we use both ActGrad and Integrated Gradients with the sum of top-1 logits as our standard attribution method, as results are anecdotally similar between the two methods.

3 Layerwise Evolution of Neural Contributions in CNNs

To ask how the actions of the network evolve throughout its layers, we first examined the sparsity of channel contributions using the Hoyer sparsity index, a normalized measure that computes the ratio of L1 to L2 norms, and ranges from 0 (all channels equally active) to 1 (only one channel active). At all layers, contributions consistently showed high sparsity across channels than activations, indicating that only a small subset of channels are functionally relevant for each classification decision (Fig 1B). Furthermore, the contributions showed a sparsification of channel contributions throughout the network, aligning with the intuition that feature selectivity emerges with hierarchical depth.

Hidden unit activations are constrained by ReLU nonlinearities to be positive. However, contributions can be positive or negative, indicating whether a spatial position increases or decreases the likelihood of the target output. Thus, a highly active unit in the activation map can be used to inhibit the network's output, as revealed by its contribution. These opposing influences, similar to excitatory and inhibitory interactions in biological vision (e.g., on-off receptive fields), are essential to neural computation but hidden in activation patterns. We therefore examined the relationship between positive and negative contributions across layers in ResNet50. Importantly, contribution sign reflects the net impact on network output, not the polarity of synaptic weights.

To compare positive and negative contributions, we computed the contribution at each spatial location (Fig 2B-E), and then separated the contribution into positive and negative components prior to spatial summation. We found that in channels of earlier layers, the magnitude of positive and negative contributions were highly correlated. However, through the network, positive and negative contributions became progressively decorrelated (Fig 1C-D). One possible explanation for this shift is that lower-level features such as edges and textures encoded in earlier layers exhibit strong spatial correlations, necessitating that individual channels contribute both positively and negatively to remove these correlations, as do center-surround receptive fields (Pitkow & Meister (2012)) and object motion sensitivity (Ölveczky et al. (2003)).

We examined the dimensionality of activations and contributions using principal components analysis on spatially summed values. Both contributions and activations increased in dimensionality through the network, but contributions showed considerably higher dimensionality than activations as measured by eigenvalues needed to reach 95 % variance (Fig 1E-F). Throughout the network,

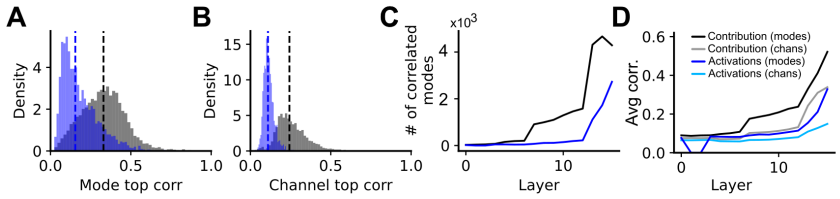

Figure 5: **Emergence of meaningful contribution-modes in intermediate layers**. (A) Histograms of each mode's maximum correlation with binary class indicators for contributions (grey) and activations (blue). (B) Same as (A) for the correlation of individual channel contributions or activations with class indicators. (C) Number of modes and channels with a correlation to a class of greater than 0.2. (D) Median of the maximal class-correlation as a function of layer.

contributions increased in sparsity, decorrelated their positive and negative effects, and increased in dimensionality.

4 Decomposing Contributions into Computational Modes

To uncover structure within these contributions, we decomposed them into a set of modes using an autoencoder consisting of an encoder network $f_{\text{enc}} : \mathbb{R}^d \to \mathbb{R}^k$, and a non-negative dictionary $\mathbf{D} \in \mathbb{R}_+^{d \times k}$, where k is the number of modes (also called atoms) typically $k = N \cdot d$ for an overcomplete representation with expansion factor N. Each column $\mathbf{m}_i \in \mathbb{R}^d$ of $\mathbf{D}$ defines one mode.

Given contributions $\mathbf{c} \in \mathbb{R}^d$, the encoder computes pre-activation loadings (also called codes) $\mathbf{z}_{\text{pre}} = f_{\text{enc}}(|\mathbf{c}|)$, where $|\cdot|$ ensures non-negative inputs. To enforce sparsity, we apply hard thresholding: $z_i = z_{\text{pre},i}$ if $z_{\text{pre},i} \geq \tau$, and $z_i = 0$ otherwise. The reconstruction is $\hat{\mathbf{c}} = \mathbf{D}\mathbf{z}$, and the autoencoder is trained to minimize the loss $\mathcal{L} = \|\mathbf{c} - \hat{\mathbf{c}}\|_2^2 = \|\mathbf{c} - \mathbf{D}\mathbf{z}\|_2^2$, with optional L1 regularization applied to the loadings and modes. Non-negativity constraints are imposed on the dictionary $\mathbf{D}$. Decomposition of contributions resulted in a set of k modes of dimension d, and a set of n loadings for each mode reflecting the weighting of those modes for each image that reconstructed the matrix of contributions with high accuracy, (average $R^2 = 0.85$ and 0.84 for contributions and activations, respectively) across all layers (Supplementary Figure 2).

To measure how closely related these modes of were related to specific network outputs, we correlated the loadings over the entire dataset (50,000 validation images) with a binary vector indicating whether a given image belonged to a given class (Fig). This resulted in a k by 1000 correlation matrix, representing the correlation of each mode with each ImageNet category. We found that contributions were more correlated with network output than activations, in particular at intermediate layers (Fig 5A-C). In addition, we found that contribution modes, despite not having access to class label during optimization were more correlated with classes than individual channels. This indicates the success of CODEC at revealing patterns of contributions that had relevance to specific network outputs (Fig 5D).

5 Controlling Network Behavior Using Contribution Modes

To test the causal link between contribution modes and ImageNet classification, we perturbed ResNet-50 by targeting channels identified through CODEC analysis. For each class, we identified the mode most correlated with that class, then measured classification accuracy under two conditions: **ablation** (removing the top-weighted channels) and **preservation** (retaining only those channels). We quantified these effects across all 1000 ImageNet classes by calculating the difference in accuracy between unperturbed and ablated networks for each class. For the "black widow" class, ablating 2% of salient channels identified from the top 2 most correlated modes greatly reduced target-class accuracy while leaving off-target classification performance largely unaffected.(Fig 6C(bottom)). Additionally, preservation analysis yielded networks that could accurately classify only the targeted class (Fig 6C(top)). We randomly sampled the Imagenet validation dataset and compared perturbation performance for a given mode and target class, and a random

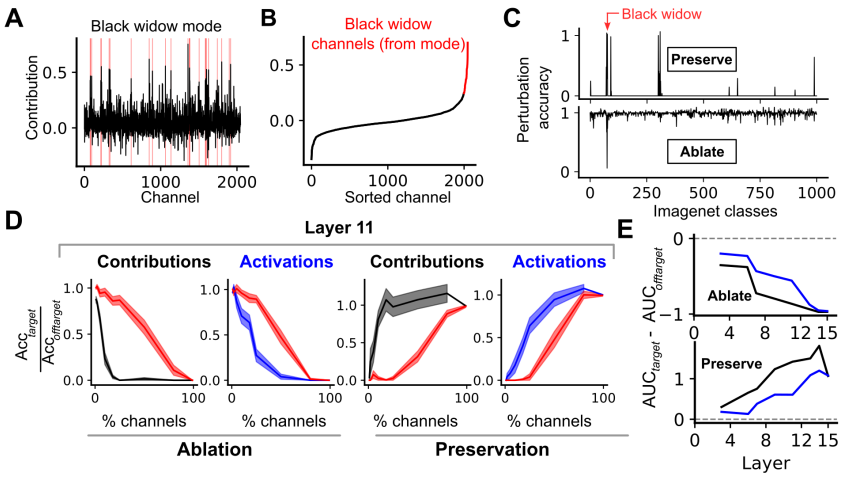

Figure 6: **Contribution-targeted network control** (A) Single mode most correlated with the "black widow" class, showing channel weightings. (B) Sorted channel weightings from (A) with few top channels highlighted for selection. (C) Results for "black widow" classification: Top shows preservation analysis (accuracy when keeping only selected channels), bottom shows ablation analysis (accuracy when removing selected channels). (D) Normalized accuracy change for target class when ablating (left) or preserving (right) channels from the most correlated contribution mode (black), activation mode (blue), or random class (red, representing non-target/off-target performance). (E) Performance score quantifying perturbation effectiveness as the area between target and off-target curves from (D) across all blocks, normalized to the off-target area.

off-target class, while varying the percentage of channels perturbed. Targeting channels by contribution modes more reliably identified necessary and sufficient channels for classification compared to activation-based analyses, requiring fewer channels to completely ablate target-class computation. (Fig 6D). A sharp increase in ablation efficacy was observed between blocks 6 and 7, potentially suggesting a shift in how semantic information is represented at this depth (Fig 6E). Additionally, we demonstrated that our approach generalizes beyond the 1000 ImageNet classes, successfully ablating / preserving specific taxonomic categories, indicating that CODEC can identify non-labled computational pathways even within broader semantic groupings (Supplementary Figure 3).

6 Visualizing inputs that cause hidden contributions

Attribution methods such as Integrated Gradients compute gradients from outputs to inputs to identify regions most influential to a decision. We extend this idea to hidden channels: what features of the input does a channel use to drive the output for a given image? Using CODEC-selected salient channels, we isolated the gradient pathway from outputs to inputs that passes only through channels of interest, decomposing traditional input-output saliency into interpretable, channel-specific contributions. For each c of mode $\mathbf{m}$ and each p, the input sensitivity is $A_i^{(c,p)} = J_{y,h_{c,p}} J_{h_{c,p},x_i}$, where $h_{c,p}$ is the activation at position p in channel c $J_{y,h_{c,p}} = \frac{\partial y}{\partial h_{c,p}}$ and $J_{h_{c,p},x_i} = \frac{\partial h_{c,p}}{\partial x_i}$. Aggregating over all selected channels and positions yields $A_i^{(m)} = \sum_{c \in \mathbf{m}_m} \sum_p A_i^{(c,p)}$. Whereas $A_i^{(m)}$ captures output sensitivity to pixel i, the contribution map viewed in input space reflects sensitivity weighted by the input: $C_i^{(m)} = A_i^{(m)} \odot x_i$. This highlights regions that most strongly drive contributions within the mode's most relevant channels. Figure 7 shows example visualizations of pixels used by hidden layers to drive network output for a specific class. We expect that further analysis of the structure of these modes in input space will reveal meaningful visual components of objects used to construct visual classes.

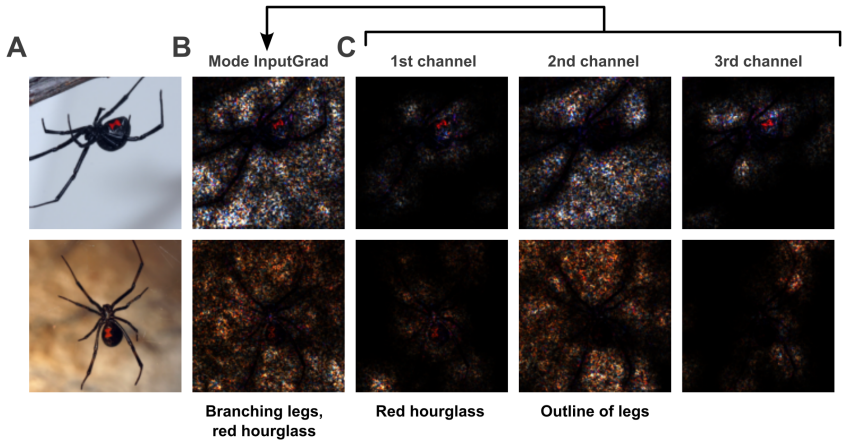

Figure 7: **Visualizing stimuli acting *through* contribution modes** By summing the product of receptive and projective fields over channels, we can visualize the saliency map of stimuli passing through individual mode with respect to a given output logit. Here, we show the saliency map of several images of black-widows (A) to the black-widow logit, through one of several highly-correlated black-widow modes (B). Individual channels convey unique features (C) from the stimulus, and together, convey an interpretable visualization of a black widow.

7 Interpreting biological neural network models with CODEC

We then used contribution decomposition to interpret the structure of computation in the biological neural circuits of the early visual system. Previous results Maheswaranathan et al. (2023) have shown that three layer convolutional neural network models capture the responses of retinal ganglion cells to natural scenes, and are interpretable in that hidden units are highly correlated with recordings from retinal interneurons not used to fit the model. We therefore applied CODEC to these CNN models to interpret how groups of model interneurons contributed to retinal output.

The CNN model consisted of 3 layers, with 8 channels (cell types) in the first two layers, yielding a response of 4–17 recorded ganglion cells (Fig. 8A). In order to choose a single contribution target, rather than choosing the entropy as we did in the case of the image classification network, we chose the surprisal, or self-information, $I(x) = -\log_2 P(x)$, an information theory measure that reflects how unexpected a response is. Estimated from the covariance of the output, $I(x)$ varies with the Mahalanobis distance of the population response from the mean, $(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$, plus a constant term, where $\boldsymbol{\Sigma}$ is the covariance matrix and $\boldsymbol{\mu}$ is the mean response. With this target, positive contributions are those that create a more unexpected response.

CODEC identified sets of modes in the first two layers of the model that combined to drive cells at different times, as measured by computing the correlation between cell firing rates and mode loadings(Fig 8B-C). This allowed a clustering of cell types by the average pattern of active modes that drove the cell. Clustering of the active mode pattern in the first and second layer yielded very similar results, indicating the robustness of identifying pathways in the model that drove different cell types (Fig 8C). We further analyzed at different times the instantaneous receptive field (IRF) of ganglion cells, which is the gradient of a cell's response with respect to the stimulus (Fig 8B,D)., revealing the visual feature driving the cell for a particular stimulus. We found that individual modes could contribute similar IRF patterns across different ganglion cells. Interestingly, when multiple modes simultaneously drove a given ganglion cell, the resulting IRF dynamically varied according to the specific combination of active modes, with patterns ranging from familiar center-surround structures to oriented or textured responses (Fig 8D). As the units of this model are highly correlated with actual interneuron recordings (Maheswaranathan et al. (2023)), these results serve as an automatically generated hypothesis for the combined activity of neural pathways that can be tested by neural recordings and perturbation.

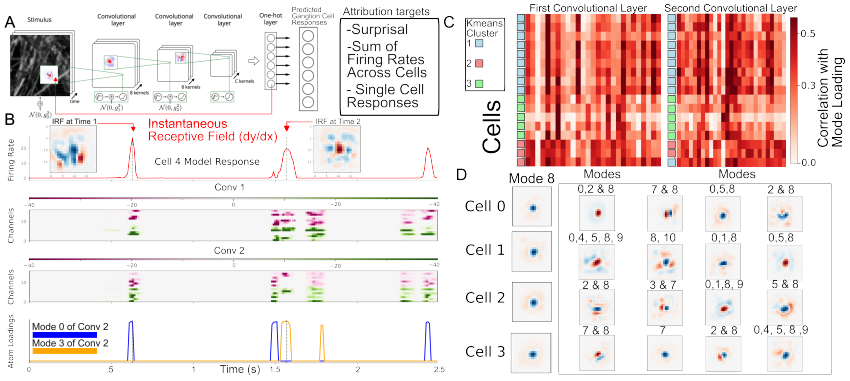

Figure 8: **Contribution modes generate dynamic receptive fields in the retina.** (A) Convolutional neural network trained to predict retinal ganglion cell responses to natural scene stimuli, from (Ding et al. (2023)). CODEC is performed for hidden layer units to single target outputs including surprisal (self-information) of the population or from single cell responses. (B) Model firing rate predictions for an example cell (top) aligned with the matrix of contributions, and with loadings for two example modes after SAE decomposition (bottom). Also shown are instantaneous receptive fields (IRFS) of the cell computed at two time points. (C) Clustering of cells using their contribution modes. Matrix shows the correlation of firing of each cell with the loading of each mode. Three clusters computed using k-means remain consistent across layers. The number of clusters was selected using the silhouette value. (D) Instantaneous receptive fields during sparse mode combinations. Left column: IRFs when one specific mode was active alone for four different cells at different times. Right: IRFs for example cells at times when 5 or fewer modes were active and cells were firing more than 1 Hz. Each row is from a different cell.

8 Conclusion

Contribution decomposition identifies how hidden units construct specific outputs, revealing both the input components that causally drive model behavior and how the effects of those features are summed across outputs. Our approach thus achieves both a deeper understanding of the computation than results from examining representations alone, along with more effective manipulation of those networks. Our analysis reveals insights into the structure of neural computation, in particular at intermediate layers of networks that have been difficult to analyze and interpret. In biological networks such as the retina, CODEC allows an analysis of how dynamic sensitivity to visual input arises from the coordinated actions of model interneurons. In artificial networks, the emergence of sparse, interpretable motifs suggests that network output can be understood in terms of a relatively small set of input-specific computations. Future work might leverage these computations as building blocks for more efficient architectures or transfer learning approaches. In the nervous system, the ability to relate computational building blocks to a downstream computation, combined with experimental measurements and manipulations at different levels could reveal how information is recombined across diverging and converging neural pathways. Such a unified approach to neural computation could bridge the gap between computers and biology, potentially enabling more powerful AI systems and deeper insights into biological intelligence.

8.1 Limitations and Ethical Concerns

Whereas we focused on image classification, our approach lays the groundwork for extensions to other architectures and applications. One significant limitation is that our decomposition technique is sensitive to hyperparameters, including hidden layer size and regularization, which may require tuning for different architectures. Further, our experiments on ResNet-50 were limited to specific blocks rather than analyzing the entire network, and although CODEC is architecturally agnostic and could be applied to more complex models such as LLMs, we have not yet empirically validated this extension.

8.2 Reproducibility statement

Our CODEC framework implementation, including code for computing hidden-layer contributions using ActGrad, Integrated Gradients, and InputGrad methods, will be made available as supplementary materials. The sparse autoencoder decomposition algorithms and visualization tools for mapping contributions back to input space are documented in the supplement, with mathematical derivations provided. The retinal neural network models analyzed are publicly available on GitHub as referenced in the citations. All experimental configurations, including autoencoder architectures, sparsity constraints, and statistical analysis procedures, are documented to enable direct to encourage replication of our findings across both artificial and biological network interpretations.

References

Xuehao Ding, Dongsoo Lee, Joshua B. Melander, George Sivulka, Surya Ganguli, and Stephen A. Baccus. Information geometry of the retinal representation manifold. In *Advances in Neural Information Processing Systems (NeurIPS) 36*, pp. 44310–44322, 2023. doi: 10.5555/3666122.3668039. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8a267516a7a697965c6ae4f48b908605-Abstract-Conference.html.

Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning, March 2017. URL http://arxiv.org/abs/1702.08608. arXiv:1702.08608 [stat].

Gerald M. Edelman and Joseph A. Gally. Degeneracy and complexity in biological systems. *Proceedings of the National Academy of Sciences*, 98(24):13763–13768, November 2001. doi: 10.1073/pnas.231499798. URL https://www.pnas.org/doi/10.1073/pnas.231499798. Publisher: Proceedings of the National Academy of Sciences.

Thomas Fel, Victor Boutin, Mazda Moayeri, Rémi Cadène, Louis Bethune, Léo andéol, Mathieu Chalvidal, and Thomas Serre. A Holistic Approach to Unifying Automatic Concept Extraction and Concept Importance Estimation, October 2023. URL http://arxiv.org/abs/2306.07304. arXiv:2306.07304 [cs].

S. R. Lehky and T. J. Sejnowski. Network model of shape-from-shading: neural function arises from both receptive and projective fields. *Nature*, 333(6172):452–454, June 1988. ISSN 0028-0836. doi: 10.1038/333452a0.

Zachary C. Lipton. The Mythos of Model Interpretability, March 2017. URL http://arxiv.org/abs/1606.03490. arXiv:1606.03490 [cs].

Mirtha Lucas, Miguel Lerma, Jacob Furst, and Daniela Raicu. RSI-Grad-CAM: Visual Explanations from Deep Networks via Riemann-Stieltjes Integrated Gradient-Based Localization. In George Bebis, Bo Li, Angela Yao, Yang Liu, Ye Duan, Manfred Lau, Rajiv Khadka, Ana Crisan, and Remco Chang (eds.), *Advances in Visual Computing*, pp. 262–274, Cham, 2022. Springer International Publishing. ISBN 978-3-031-20713-6. doi: 10.1007/978-3-031-20713-6_20.

Niru Maheswaranathan, Lane T. McIntosh, Hidenori Tanaka, Satchel Grant, David B. Kastner, Joshua B. Melander, Aran Nayebi, Luke E. Brezovec, Julia H. Wang, Surya Ganguli, and Stephen A. Baccus. Interpreting the retinal neural code for natural scenes: From computations to neurons. *Neuron*, 111(17):2742–2755.e4, September 2023. ISSN 1097-4199. doi: 10.1016/j.neuron.2023.06.007.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. 1943. *Bulletin of Mathematical Biology*, 52(1-2):99–115; discussion 73–97, 1990. ISSN 0092-8240.

W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, October 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1900654116. URL http://arxiv.org/abs/1901.04592. arXiv:1901.04592 [stat].

Bruno A. Olshausen and David J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, August 2004. ISSN 0959-4388. doi: 10.1016/j.conb.2004.07.007.

Xaq Pitkow and Markus Meister. Decorrelation and efficient coding by retinal ganglion cells. *Nature Neuroscience*, 15(4):628–635, 2012. doi: 10.1038/nn.3064.

Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690):1461–1467, March 2024. doi: 10.1126/science.adi5639. URL https://www.science.org/doi/10.1126/science.adi5639. Publisher: American Association for the Advancement of Science.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL https://www.nature.com/articles/s42256-019-0048-x. Publisher: Nature Publishing Group.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2):336–359, February 2020. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://arxiv.org/abs/1610.02391. arXiv:1610.02391 [cs].

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3145–3153, Sydney, NSW, Australia, August 2017. JMLR.org.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise, June 2017. URL http://arxiv.org/abs/1706.03825. arXiv:1706.03825 [cs].

Suraj Srinivas and Francois Fleuret. Full-Gradient Representation for Neural Network Visualization, December 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks, June 2017. URL http://arxiv.org/abs/1703.01365. arXiv:1703.01365 [cs].

Hidenori Tanaka, Aran Nayebi, Niru Maheswaranathan, Lane McIntosh, Stephen Baccus, and Surya Ganguli. From deep learning to mechanistic understanding in neuroscience: The structure of retinal prediction. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, June 2014. doi: 10.1073/pnas.1403112111. URL https://www.pnas.org/doi/10.1073/pnas.1403112111. Publisher: Proceedings of the National Academy of Sciences.

Zizhen Yao, Cornelius T. J. van Velthoven, Menno Kunst, et al. A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature*, 624:317–332, 2023. doi: 10.1038/s41586-023-06812-z. URL https://doi.org/10.1038/s41586-023-06812-z.

Bence P. Ölveczky, Stephen A. Baccus, and Markus Meister. Segregation of object and background motion in the retina. *Nature*, 423(6938):401–408, 2003. doi: 10.1038/nature01652.

9 Appendix

Supplemental Figures

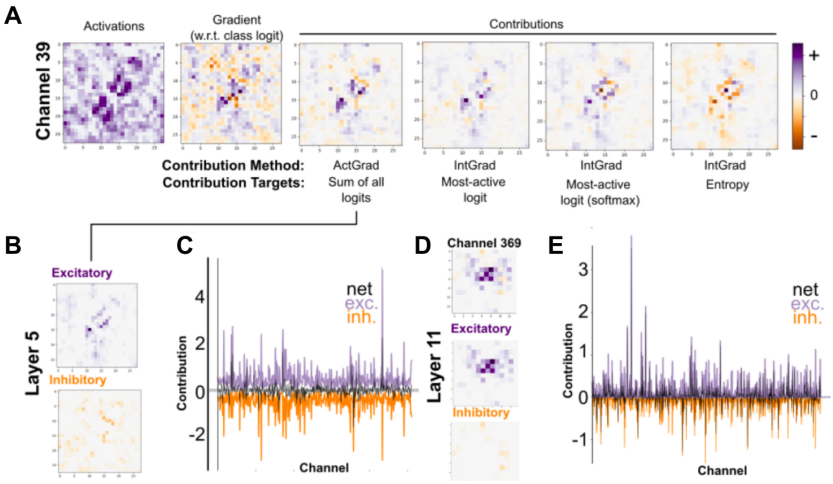

Figure 9: **Comparison of attribution methods and targets reveals consistent channel contribution patterns**. (A) Spatial maps of activations, gradients, and contributions computed using different attribution methods (ActGrad, IntGrad) and targets (sum of all logits, most-active logit, most-active logit with softmax, entropy) for Channel 39. All methods show similar spatial sparsity. (B) Excitatory and inhibitory contribution maps for Channel 39 in Layer 5 for a single image demonstrating the separation of exication and inhibition in an early layer. (C) Contribution values across channels for one image showing the distribution of net excitatory and inhibitory effects for Layer 5. (D) Same as (B) but for Channel 369 in Layer 11, illustrating consistent excitatory/inhibitory patterns across different channels. (E) Same as (C) but for Layer 11, showing how the excitatory/inhibitory decorrelation evolves across network depth.

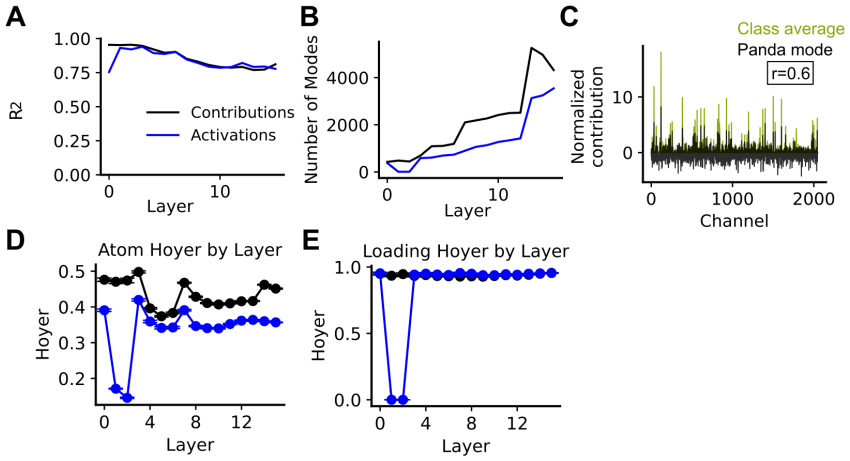

Figure 10: **Sparse autoencoder decomposition performance and sparsity analysis across network layers**. (A) Reconstruction accuracy (R^2) of sparse autoencoder decompositions for contributions (black) and activations (blue) across network layers. Both show high reconstruction fidelity, with contributions maintaining slightly higher R^2 values in intermediate layers. (B) Number of modes discovered by the sparse autoencoder decomposition as a function of network depth for contributions (black) and activations (blue). The number of modes increases through the network, with contributions yielding more modes than activations in deeper layers. (C) Comparison of the most correlated panda mode with the average contribution pattern during presentation of panda images. The decomposition identified a mode (black) that shows similar contribution patterns to the class-average contributions when panda images are presented (r=0.6) (D) Median Hoyer sparsity index of learned modes (channel atoms) across network layers for contributions (black) and activations (blue). Contribution modes maintain higher sparsity than activation modes throughout the network. (E) Median Hoyer sparsity index of mode loadings across network layers. Loading sparsity drops sharply in early layers then stabilizes

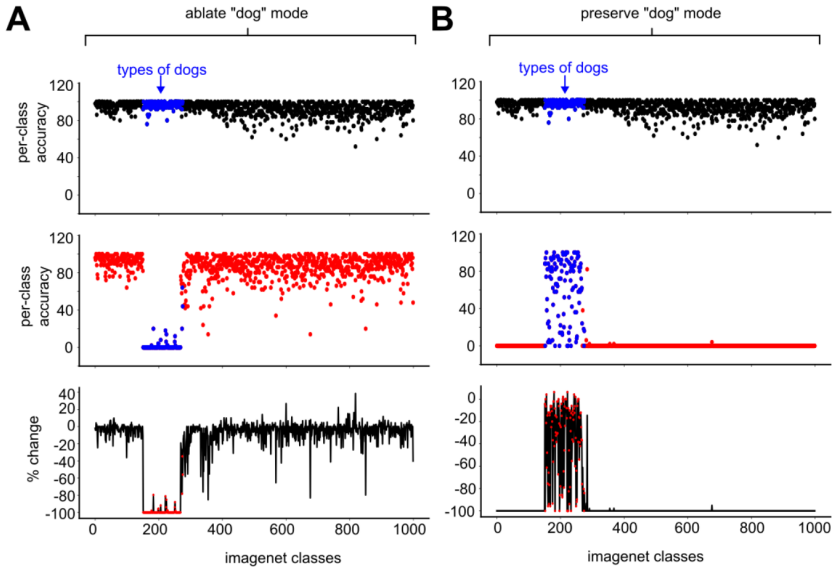

Figure 11: **Validation of contribution modes through targeted channel manipulation**. (A) Ablation experiment: A binary mask was created for all presented images containing dogs, then correlated with mode loadings to identify the top dog-correlated mode. Removing channels from this dog mode in an intermediate layer eliminates the network's ability to classify dog images while leaving other classes largely unaffected. (B) Preservation experiment: Using the same dog mode identified through correlation analysis, retaining only channels from this mode allows the network to correctly identify only dog images, with all other classes showing near-zero accuracy.

CODEC IN DEPTH

The fundamental algorithmic innovation of our work is that we design an approach to address the question of how combinations of hidden neurons drive network output. This contrasts with an analysis of activations, which at any level and for any input are not guaranteed to drive network output. Here, we provide additional details of our methods.

9.1 ADDITIONAL DETAILS ON CONTRIBUTION ALGORITHMS AND THEIR COMPLETE DECOMPOSITIONS IN INPUT SPACE

We revisit and describe the motivation behind our several contribution algorithms, which are inspired by gradient-based attribution methods, and we derive their complete decompositions in input space. For notation, let h_i be the hidden units in a specific layer L. All attribution methods need a scalar output target, typically a select output neuron or a scalar function of the output neurons. Let T denote the target (for example the output logit of a target class or some general scalar function of the output neurons) and $y \coloneqq f_T$ be the neural network with a scalar output y corresponding to T. For clarity, we define the complete input space decomposition of hidden units:

Definition 1 (Complete input space decomposition of hidden units)**.** *Let C_j be the contribution of hidden unit j and $\widetilde{C}_{ij}$ be an input space decomposition of C_j, i.e., $\widetilde{C}_{ij}$ is the part of C_j assigned to input pixel i. Then, we call this input space decomposition complete if*

$$\sum_i \widetilde{C}_{ij} = C_j. \tag{1}$$

We derive the complete input space decomposition of ActGrad, InputGrad, and Integrated Gradients under this definition, which will naturally generalize to mode contributions by linearity as a simple sum over the all the hidden units in that mode.

A note about ActGrad To be consistent with the rest of the derivation, we slightly generalize ActGrad to account for possible nonzero baselines:

$$\text{ActGrad}_j = (h_j - h'_j) \cdot \frac{\partial y}{\partial h_j}, \tag{2}$$

where h_j is the activation of hidden unit j (which could be the input) at input $\mathbf{x}$ and h'_j is the hidden activation at baseline input $\mathbf{x}'$. For image classification, we usually take a baseline hidden activation of zero, which is the definition in the main text and what we used in experiments.

Input $\times$ Gradient (InputGrad) Used in Ref. Shrikumar et al. (2017) as a baseline, this is a special case of ActGrad where Act is the input values. However, seeing InputGrad this way means it can only attribute to the input layer. We naturally extend InputGrad to hidden layers to obtain contributions by defining the decomposition:

$$\widetilde{\text{InputGrad}}_{ij} \coloneqq \frac{\partial y}{\partial h_j}\frac{\partial h_j}{\partial x_i}(x_i - x'_i) \tag{3}$$

$$= ((J_y)_j (J_z)_{ji}) \cdot (x_i - x'_i), \tag{4}$$

where $J_y \coloneqq \frac{\partial y}{\partial h_j}$ is the Jacobian from the output to the hidden layer and $J_z \coloneqq \frac{\partial h_j}{\partial x_i}$ is the Jacobian from the hidden layer to the input. If we sum over i, we obtain a contribution algorithm to hidden neurons:

$$\text{HInputGrad}_j \coloneqq \sum_i \widetilde{\text{InputGrad}}_{\text{ij}}. \tag{5}$$

Under this definition of InputGrad contribution, the decomposition is trivially complete. This is a well-motivated extension of InputGrad because we recover input space InputGrad by summing over j instead:

$$\text{InputGrad}_i = \sum_j \widetilde{\text{InputGrad}}_{\text{ij}}. \tag{6}$$

This natural extension rests on the chain rule of partial derivatives, where we essentially postpone summing gradients over the hidden layer until after multiplying by the input element-wise. InputGrad as a contribution algorithm may seem a bit ad hoc, but it will turn out to have the most computationally efficient input space decomposition that we consider and we theoretically motivate it later by showing a connection to Integrated Gradients.

Integrated Gradients (IG) Sundararajan et al. (2017) IG assigns attributions to input pixels by calculating the integral of gradients along a straight-line path from a baseline input $\mathbf{x}'$ to the actual input $\mathbf{x}$:

$$\text{IG}_i = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial y(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial x_i} d\alpha. \tag{7}$$

IG satisfies the (output) *completeness* property, often desired in attribution methods:

$$\sum_i \text{IG}_i = y(x) - y(x'). \tag{8}$$

In words, completeness means that the attributions to each input pixel sum up to the (change in) output. In this original form, IG only works for the input layer. We present two ways to extend IG to hidden layers:

- Treat the hidden layer as the input and follow through. This is the extension by Lucas et al. (2022), where they combine this extension of IG with GradCAM. Mathematically, the "hidden integrated gradients" (HIG) is defined as

$$\text{HIG}_j = (h_j - h_j') \times \int_{\alpha=0}^{1} \frac{\partial y(\mathbf{h}' + \alpha(\mathbf{h} - \mathbf{h}'))}{\partial h_j} d\alpha, \tag{9}$$

  where $\mathbf{h}$ is the hidden layer activation vector at input $\mathbf{x}$, and $\mathbf{h}'$ is the hidden layer activation vector at input $\mathbf{x}'$.

- Take the same path in input space but decompose the gradients at the hidden layer and sum over the input layer. Tanaka et al. (2019) first introduced this method and applied it to the first hidden layer of a model of the retina. Here, we use it on any hidden layer. Mathematically, first, we decompose the gradients over the hidden layer:

$$\widetilde{\text{HIG}}_{ij} \coloneqq (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial y(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial h_j} \frac{\partial h_j}{\partial x_i} d\alpha, \tag{10}$$

  where i indexes the input pixels and j indexes the hidden neurons. Note that we can recover standard IG by summing over j due to the linearity of the integral and chain rule of partial derivatives:

$$\text{IG}_i = \sum_j \widetilde{\text{HIG}}_{ij}, \tag{11}$$

  which motivates this extension. Now, to obtain attributions over the hidden layer, we simply sum over i instead of j:

$$\text{HIG}_j \coloneqq \sum_i \widetilde{\text{HIG}}_{ij}. \tag{12}$$

In practice, we approximate the integral using a Riemann sum with m steps. For standard IG, we have

$$\text{IG}_i \approx (x_i - x_i') \times \frac{1}{m} \sum_{k=1}^{m} \frac{\partial y(\mathbf{x}' + \frac{k}{m}(\mathbf{x} - \mathbf{x}'))}{\partial x_i}. \tag{13}$$

This approximation extends straightforwardly to the first method of HIG, simply by replacing $\mathbf{x}$ with $\mathbf{h}$. For the second method, we could naively discretize in input space to get $\widetilde{\text{HIG}}_{ij}$ first, but that would involve a backward pass for each hidden neuron at each step. Instead, we analytically derive

an approximation that still only involves one backward pass at each step. We have

$$\text{HIG}_j = \sum_i (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial y(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial h_j} \frac{\partial h_j}{\partial x_i} d\alpha \tag{14}$$

$$= \int_{\alpha=0}^{1} \sum_i \left[\frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial x_i} \frac{\partial x_i}{\partial \alpha} \right]_{\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}')} d\alpha \tag{15}$$

$$= \int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j} \sum_i \frac{\partial h_j}{\partial x_i} \frac{\partial x_i}{\partial \alpha} \right]_{\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}')} d\alpha \tag{16}$$

$$= \int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial \alpha} \right]_{h(\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}'))} d\alpha. \tag{17}$$

Note that this is precisely IG treating $\mathbf{h}$ as the input layer with the important distinction that the path is not a straight line (it is warped by the nonlinear functional relationship between $\mathbf{x}$ and $\mathbf{h}$). The crucial advantage of computing HIG this way over the first method is the complete input space decomposition (Equation 10) we get for free. To approximate this line integral with a Riemann sum, we have

$$\text{HIG}_j \approx \sum_{k=1}^{m} \left. \frac{\partial y}{\partial h_j} \right|_{\mathbf{h}(\alpha=\frac{k}{m})} dh_j, \tag{18}$$

where we can compute $\mathbf{h}(\alpha = \frac{k}{m})$ at each k in a forward pass and $dh_j = h_j(\alpha = \frac{k}{m}) - h_j(\alpha = \frac{k-1}{m})$ in a single backward pass.

Complete input space decomposition of ActGrad It is a bit more nuanced to define a complete input space decomposition for ActGrad in hidden layers. The idea is to use IG to attribute Act to inputs, which we then multiply by Grad to get the decomposition. Mathematically, we exploit the output completeness of IG to express Act as an integral of gradients, which we can then naturally decompose linearly in input space. We define the input space decomposition as

$$\widetilde{\text{ActGrad}}_{ij} \coloneqq (x_i - x_i') \left. \frac{\partial y}{\partial h_j} \right|_{\mathbf{x}} \int_{\alpha=0}^{1} \frac{\partial h_j}{\partial x_i} d\alpha, \tag{19}$$

where we take a straight-line path in input space parameterized by α. We can prove that this decomposition is complete by recovering the standard ActGrad for hidden layers if we sum over i:

$$\sum_i \widetilde{\text{ActGrad}}_{ij} = \sum_i (x_i - x_i') \left. \frac{\partial y}{\partial h_j} \right|_{\mathbf{x}} \int_{\alpha=0}^{1} \frac{\partial h_j}{\partial x_i} d\alpha \tag{20}$$

$$= \left. \frac{\partial y}{\partial h_j} \right|_{\mathbf{x}} \sum_i (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial h_j}{\partial x_i} d\alpha \tag{21}$$

$$= \left. \frac{\partial y}{\partial h_j} \right|_{\mathbf{x}} (h_j - h_j') \text{ (due to the output completeness of IG)} \tag{22}$$

$$= \text{ActGrad}_j, \tag{23}$$

where the second-to-last step is true even though the path in h space is not necessarily straight by the fundamental theorem of calculus for line integrals. Note that if we summed over j instead, we would obtain a new attribution method for the input layer that is like a hybrid between IG and InputGrad. We do not explore that method any further as we are primarily interested in contributions of hidden neurons.

InputGrad and ActGrad as approximations to IG The unifying connection behind all three contribution algorithms is that they are the same for a linear network. The three algorithms represent successively more comprehensive ways to capture the nonlinearity of the network in their contribution assignments. Recall the definition of HIG (Equation 17):

$$\text{HIG}_j = \int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j} \frac{\partial h_j}{\partial \alpha} \right]_{h(\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}'))} d\alpha. \tag{24}$$

If we assume that the downstream network is linear along the integration path, i.e., the downstream gradients $\frac{\partial y}{\partial h_j}$ are constant, we recover ActGrad:

$$\int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j}\frac{\partial h_j}{\partial \alpha}\right]_{h(\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}'))} d\alpha = \left.\frac{\partial y}{\partial h_j}\right|_{\mathbf{x}} \int_{\alpha=0}^{1} \left[\frac{\partial h_j}{\partial \alpha}\right]_{h(\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}'))} d\alpha \tag{25}$$

$$= \left.\frac{\partial y}{\partial h_j}\right|_{\mathbf{x}} (h_j - h'_j) \tag{26}$$

$$= \mathrm{ActGrad}_j. \tag{27}$$

If we further assume the whole network is linear along the integration path, i.e., all gradients are constant, we recover InputGrad:

$$\int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j}\frac{\partial h_j}{\partial \alpha}\right]_{h(\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}'))} d\alpha = \int_{\alpha=0}^{1} \left[\frac{\partial y}{\partial h_j}\sum_i \frac{\partial h_j}{\partial x_i}\frac{\partial x_i}{\partial \alpha}\right]_{\mathbf{x}'+\alpha(\mathbf{x}-\mathbf{x}')} d\alpha \tag{28}$$

$$= \left[\frac{\partial y}{\partial h_j}\sum_i \frac{\partial h_j}{\partial x_i}\frac{\partial x_i}{\partial \alpha}\right]_{\mathbf{x}} \int_{\alpha=0}^{1} d\alpha \tag{29}$$

$$= \sum_i \frac{\partial y}{\partial h_j}\frac{\partial h_j}{\partial x_i}(x_i - x'_i) \tag{30}$$

$$= \sum_i \widetilde{\mathrm{InputGrad}}_{ij} \tag{31}$$

$$= \mathrm{InputGrad}_j. \tag{32}$$

By taking the full integral, IG has the output completeness property $\sum_j \mathrm{HIG}_j = y - y'$ that ActGrad and InputGrad don't have. Furthermore, IG alleviates the saturation problem where a zero gradient counterintuitively leads to a zero attribution (and therefore contribution) Shrikumar et al. (2017); Srinivas & Fleuret (2019).

Under our definition, a mode's contribution is the sum of the contributions of its top-k channels. Due to the linearity of sums and integrals, the complete input space decompositions derived above for individual hidden units naturally generalize to those of modes.

The key insight that underlies completeness is the chain rule of partial derivatives and linearity of integrals and sums (and additionally the output completeness of IG for ActGrad decomposition). The high-level procedure is:

1. When computing the contribution of each hidden unit, do not sum the gradients over the input space just yet; save the full contribution tensor with the input index.
2. Sum over the input space and cluster to find modes.
3. Go back to the unsummed contributions and sum over each *mode* to get a complete decomposition of mode contributions in input space.

Practically, IG is more computationally expensive than ActGrad due to the integration, and we find ActGrad to perform well enough, so we use ActGrad to discover features for all our experiments. Furthermore, we use the input space decomposition of InputGrad as an approximation to that of ActGrad, again due to the computational cost of integration. Ref. Selvaraju et al. (2020) notes that ActGrad at earlier layers produce less clear heatmaps of attributions, but since we use ActGrad at the hidden layer first to identify modes, we do not expect the loss in quality to be as significant by switching to InputGrad just for visualization. Empirically, our experimental visualizations show sufficient quality for our purposes. However, our theoretical framework allows for a principled, complete decomposition if necessary for the application at hand.

9.2 Spatial aggregation and E/I separation

Once the contributions have been computed using the chosen contribution method and target, we obtain a map of the intermediate layer of the same size as the layer's activations. For convolutional layers, we perform spatial summation over the height and width dimensions to derive a single

contribution value per channel:

$$\text{ChanC} = \sum_{h=1}^{H} \sum_{w=1}^{W} \text{IG}_{:,h,w}, \tag{33}$$

where "ChanC" means "channel contribution". We further reduce these contributions into excitatory (positive) and inhibitory (negative) components:

$$\text{ChanC}^{+} = \max(\text{ChanC}, 0); \quad \text{ChanC}^{-} = \min(\text{ChanC}, 0). \tag{34}$$

This E/I decomposition enhances interpretability by separating units that promote versus suppress specific outputs, revealing antagonistic computational mechanisms within the network. For the majority of analyses presented here, we focus exclusively on the positive contributions, which can be interpreted as the hidden unit contributions that positively drive the attribution target. However, future work could extend this approach to also incorporate negative contributions, depending on the nature of the output target.

9.3 Correlation analysis with semantic concepts

To interpret the discovered modes, we analyzed their correlation with known semantic concepts from the ImageNet hierarchy. We constructed binary masks $\mathbf{M} \in \{0, 1\}^{n \times c}$ representing c different semantic concepts , where $M_{i,j} = 1$ if sample i belongs to a higher order concept/ ImageNet Class j and 0 otherwise.

We then computed the Pearson correlation coefficient between each mode loading and each concept mask:

$$\text{Corr}(L_i, M_j) = \frac{\text{Cov}(L_i, M_j)}{\sigma_{L_i} \sigma_{M_j}} \tag{35}$$

where L_i is the i-th column of the loading matrix $\mathbf{L}$ (corresponding to mode i), M_j is the j-th column of the mask matrix $\mathbf{M}$ (corresponding to concept j), Cov is the covariance, and σ is the standard deviation.

This analysis enabled us to assign interpretable meanings to the learned modes and understand how semantic information is distributed across the network.