# Understanding Inhibition Through Maximally Tense Images

**Christopher Hamblin** [1] **Srijani Saha** [1] **Talia Konkle** [1] **George Alvarez** [1]

[1]Department of Psychology, Harvard University

## Abstract

We address the functional role of *feature inhibition* in vision models; that is, what are the mechanisms by which a neural network ensures images do *not* express a given feature? We observe that standard interpretability tools in the literature are not immediately suited to the inhibitory case, given the asymmetry introduced by the ReLU activation function. Given this, we propose inhibition be understood through a study of *maximally tense images* (MTIs), i.e. those images that excite and inhibit a given feature simultaneously. We show how MTIs can be studied with two novel visualization techniques; +/- attribution inversions, which split single images into excitatory and inhibitory components, and the attribution atlas, which provides a global visualization of the various ways images can excite/inhibit a feature. Finally, we explore the difficulties introduced by superposition, as such interfering features induce the same attribution motif as MTIs.

## 1. Introduction

What makes an image *not* activate a given feature in a neural network? This is the opposite of the question one typically asks, but it is important nonetheless; features are only useful if they are *discriminative*, that is, if they activate in response to certain attributes of the input, but not others. A supposed 'banana' feature that activates for images of 'duck bills' isn't much of a 'banana' feature at all, and cannot be employed by the model as such. What, if any, are the mechanisms in a neural network that make features discriminative, that make duckbills *not* bananas? If such mechanisms exist, how do we identify them? Do we need new tools, or is the current interpretability toolbox up to the task?

**Maximally Exciting Images.** If we start by taking stock of this toolbox, we notice a common attribute of nearly every method is a reliance on *maximally exciting images*, or MEIs(Klindt et al., 2023). In the general case, a feature can be thought of as a scalar-valued function of images, and an MEI is any input for which this function returns a large value. The simplest form of MEIs are the top-$k$ activating images from a large dataset(Olah et al., 2017; Borowski et al., 2020), but more sophisticated interpretability techniques rely on them just them same. Feature visualization techniques synthesize MEIs with gradient ascent, in such a way that the optimized image expresses human-perceptible features, rather than adversarial ones(Olah et al., 2017; Mahendran & Vedaldi, 2015; Nguyen et al., 2015; Tyka; Tsipras et al., 2018; Santurkar et al., 2019; Engstrom et al., 2019; Nguyen et al., 2016b; Mordvintsev et al., 2015; Wei et al., 2015; Nguyen et al., 2016a; 2017). Saliency map techniques return a heatmap over an image that highlights the most important regions of a given image for the expression of a given feature(Simonyan et al., 2013; Bach et al., 2015; Baehrens et al., 2010; Smilkov et al., 2017; Sundararajan et al., 2017b; Fel et al., 2021; Novello et al., 2022; Fong & Vedaldi, 2017; Zintgraf et al., 2017; Petsiuk et al., 2018; Fel et al., 2023c; Ribeiro et al., 2016; Lundberg & Lee, 2017). In effect, saliency maps reveal smaller, spatially localized MEIs that the user should identify with the feature. 'Concept'-based techniques specify the features in a model we should be studying in the first place(Kim et al., 2018; Ghorbani et al., 2019; Fel et al., 2023d;b; Zhang et al., 2020), but these are typically paired with an assessment of MEIs, as features still need to be understood regardless of how they are identified in the model. Finally, where the above techniques characterize what features

**Figure 1:** How might a network construct an accurate 'banana' feature, that *doesn't* activate for 'duckbills'?

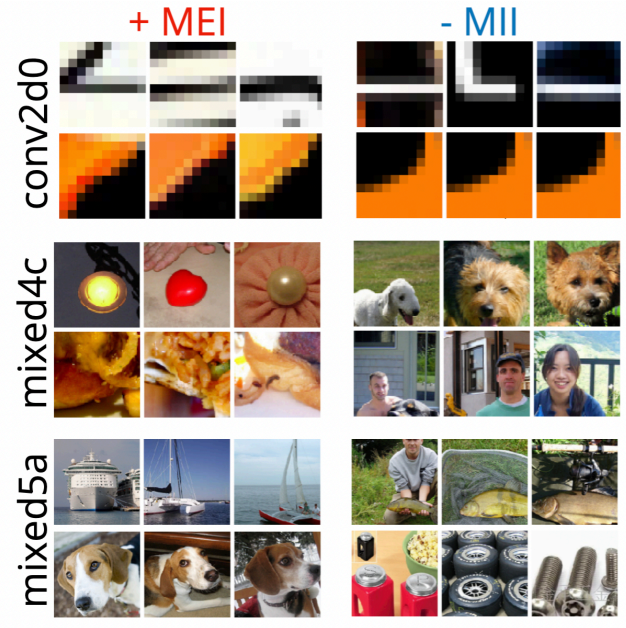


**Figure 2:** Imagenet validation dataset example MEI and MII images for random features across several layers of InceptionV1. For each layer, the top row images correspond to MEIs/MIIs for a unit in that layer. The bottom row images correspond to a feature direction identified with k-means clustering. For both unit and k-means features, MEIs and their respective MIIs seem relatable in early layers, but arbitrarily paired in later layers.

*represent*, mechanistic approaches seek to explain how features are *computed*(Olah et al., 2020a; Elhage et al., 2021). Mechanistic accounts of a model typically describe functions that operate on simpler/earlier features to compute complex ones. Even here, it is common practice to visualize the component features that comprise such functions with MEIs(Fel et al., 2023d; Cammarata et al., 2021; Olah et al., 2020b; Carter et al., 2019).

**Maximally Inhibiting Images.** Given MEIs are ubiquitous in our understanding of what features *are*, a natural starting point for understanding what features *aren't* is MIIs, or "maximally inhibiting images". Such images are sometimes considered in the literature; for example, when characterizing many units in a single model, each can be quickly conveyed through its set of MEIs and MIIs(Olah et al., 2017). Additionally, human experiments on the interpretability of features often invoke MIIs in their design; a feature is considered 'interpretable' if humans can extrapolate from its set of MEIs and MIIs, correctly predicting whether new images belong to the MEI or MII set (Borowski et al., 2020; Zimmermann et al., 2021; Klindt et al., 2023). Mechanistic interpretability methods that consider inhibition also invoke MII; for example, by looking at the *MEI*s for units connected by large negative weights(Olah et al., 2020a; Cammarata et al., 2021; Olah et al., 2020b) or large negative attributions(Carter et al., 2019).

MIIs may be a natural starting point for understanding what inhibits features, but they raise immediate concerns. First, consider the model architecture, specifically the ReLU activation function, $\text{ReLU}(\boldsymbol{x}) = \max(\boldsymbol{x}, 0)$, which introduces an asymmetry between positive and negative activations. What is the use in knowing an image induces a large negative activation if this is precisely the information the ReLU function *throws out*? Negative weights are not learned by the model *so that* features return large negative values for certain inputs. It's not clear what MIIs mean to the network, but there's also a second problem; in many cases MIIs are not meaningful to humans. This is an empirical point, by which we mean a feature's MEIs and MIIs often bear no visual relationship to each other, particularly for the high-level features represented in the later layers of the network. One might hope that MIIs have some property we can intuit as the 'opposite' of their respective MEIs, in which case the two image sets would constitute the poles of a meaningful axis. For example, consider the MEIs and MIIs for features in layer 'Conv2d0' of InceptionV1 (Szegedy et al., 2015), shown in Figure 2. In this first convolution layer the MIIs are approximately the MEIs but with their colors reversed; for example, orange above black becomes black above orange. However in late layers like 'mixed5a', where features have large receptive fields and rich semantics, the relationship between MEIs and MIIs seems arbitrary. We validate these intuitions with a human experiment, showing people can learn to predict MIIs from MEIs in the first layer, but not in later ones (appendix B).

To address these issues with MIIs, we present the following contributions:

- We introduce analyses of *maximally tense images* (MTIs), which excite and inhibit a target feature simultaneously. With regards to MTIs, negative weights play a meaningful functional role, as the feature would erroneously activate in response to its MTIs were it not for these weights.
- We present 2 novel feature visualization techniques that explain feature inhibition both locally and globally.
- We explore how inhibitory weights facilitate superposition, and the difficulty this introduces for a mechanistic understanding of inhibition.

**Notation.** In what follows, we consider a neural network $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$, which transforms an input image, $\boldsymbol{x} \in \mathcal{X}$, through a sequence of $L$ hidden representations. Let $\boldsymbol{f}_\ell : \mathcal{X} \to \mathcal{H}_\ell$ denote the function mapping the image to the $\ell^{\text{th}}$ such hidden representation, $\boldsymbol{h}_\ell = (h_1, ..., h_{n_\ell})^\mathsf{T} \in \mathcal{H}_\ell \subset \mathbb{R}^{n_\ell}$. In this work, a feature corresponds to a vector $\boldsymbol{v} \in \mathcal{H}_\ell$, and the function that computes the feature's 'activation' as $\boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{x}) = \boldsymbol{f}_\ell(\boldsymbol{x}) \cdot \boldsymbol{v}$.

## 2. Attribution Completeness

Specifying MTIs will rely on computing 'attributions' for feature activations in an earlier layer of the model. In particular, we will leverage an empirical property of these attributions, that they behave *additively*. Consider a fully linear model, $y = \mathbf{w}\boldsymbol{x} = w_1 x_1 + ... + w_n x_n$. Observe that in this linear case $w_i = \frac{\partial y}{\partial x_i}$, thus $y = \nabla_{\boldsymbol{x}} y \cdot \boldsymbol{x}$. When $y$ is a nonlinear function of $\boldsymbol{x}$, $y = \nabla_{\boldsymbol{x}} y \cdot \boldsymbol{x}$ is a linear approximation, useful for many different applications, such as pruning (Cun et al., 1990; Molchanov et al., 2019; Lee et al., 2018) and saliency maps (Simonyan et al., 2014; Smilkov et al., 2017). In our case, we want to understand the computation of feature $\boldsymbol{f}_{\boldsymbol{v}}$ as some function of features computed in an earlier layer, so let's define a layer-to-feature attribution vector, $\boldsymbol{S}_l$:

$$\boldsymbol{S}_l(\boldsymbol{x}, \boldsymbol{f}_{\boldsymbol{v}}) := \nabla_{\boldsymbol{h}_l} \boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{x}) \odot \boldsymbol{h}_l \tag{1}$$

Let $E_l$ denote the sum of all elements in $\boldsymbol{S}_l$; if $\boldsymbol{f}_{\boldsymbol{v}} \approx E_l$, then $\boldsymbol{S}_l$ can be used to select *maximally tense images* (MTIs). MTIs are those images for which the attribution vector $\boldsymbol{S}_l$ has both large positive and negative terms, indicating instances in which inhibition plays an important functional role, additively negating excitation. Previous work has called the property in which an attribution vector sums to $\boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{x})$ 'completeness', and unfortunately has observed that $E_l$ as defined does not satisfy completeness when attributing across entire image classification models, from a class probability to pixels(Sundararajan et al., 2017a; Shrikumar et al., 2017; Bach et al., 2015). However, $E_l$ may still be complete when attributing between latent layers of the model, avoiding the gradient-flattening effects of the final softmax, as well as the non-linear relationship between pixel intensities and representations in later layers.

To test this, we computed $E_l(\boldsymbol{x}, \boldsymbol{f}_{\boldsymbol{v}})$ for 20 random logits and all Imagenet (Deng et al., 2009) validation set images across several models. Figure 3 shows the average Pearson's correlation of $\boldsymbol{f}_{\boldsymbol{v}}$ and $E_l$ across logits. $E_l$ is computed for all ReLU layers in the model and the pixel space, and Figure 3 orders these measurements by layer depth. We find that $E_l$ are close to ceiling in their correlation with logits when computed through most layers of the model. When measured through pixels however, the correlation deteriorates drastically. We find batch normalization layers also corrupt this relationship (appendix).

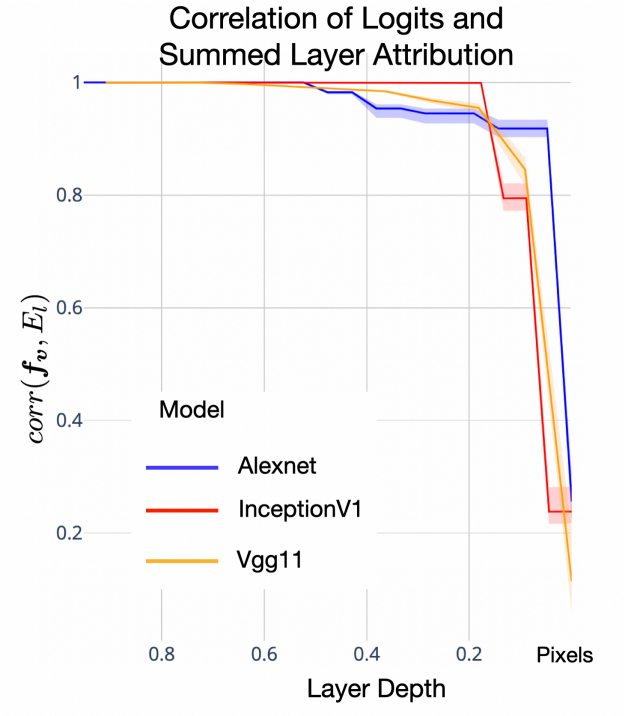


**Figure 3:** The correlation between logits, $\boldsymbol{f}_{\boldsymbol{v}}$, and total attribution $E_l$ measured across layers. $\boldsymbol{f}_{\boldsymbol{v}} \approx E_l$ across all layers, except when measured through very early layers and pixels.

## 3. Curve Feature Case Study

We'll now explore several visualization techniques that leverage the completeness of the layer-to-feature attribution vector. Throughout, we'll apply these techniques to feature "mixed3b:379" in Inceptionv1 (Szegedy et al., 2015) – a purported curve detector studied extensively in Cammarata et al. (2020)(Cammarata et al., 2020). We do this so the information provided by different techniques can be easily compared, but the methods can be applied to other features as well (appendix). To begin, let's group $\boldsymbol{S}_l$ into the sum of its positive and negative terms;

$$E_l^+ := \sum_{i=1}^{n_\ell} \text{ReLU}(\boldsymbol{S}_l)_i, \quad E_l^- := \sum_{i=1}^{n_\ell} \text{ReLU}(-\boldsymbol{S}_l)_i. \tag{2}$$

Figure 4.a shows the distribution of attributions $(E_l^+, E_l^-)$ for feature "mixed3b:379" in the preceding layer. Each point represents the attributions for the central activation in the feature's activation map in

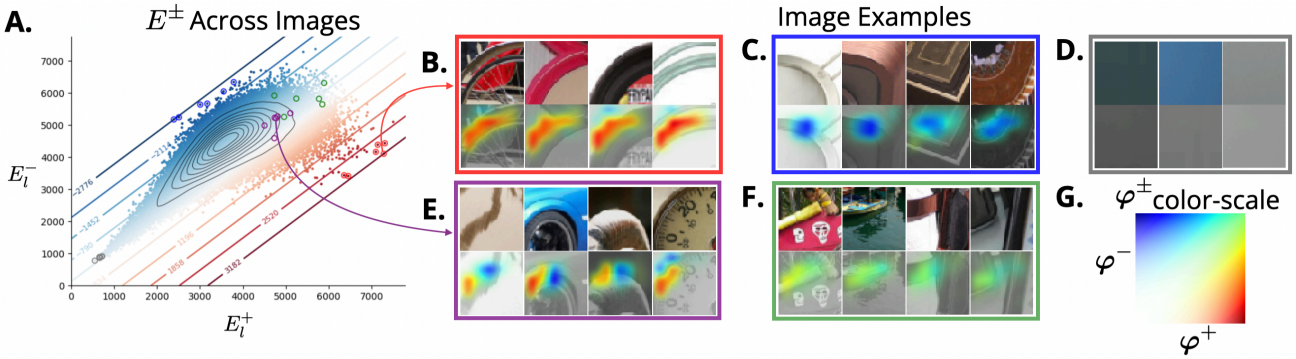


**Figure 4:** **A.** A scatterplot of $E_l^+$ and $E_l^-$ for a proposed 'curve detector' unit, across validation set images. Selected images visualized in **B.-F.** are circled in with the corresponding color. **B.** shows MEI examples, **C.** MIIs, and **D.** images with no attribution. **E.** shows images with positive and negative attributions in different spatial location, while **F.** shows images with positive and negative attribution in the channel dimension, at the same spatial location. **G.** The colorscale used for the $(\boldsymbol{\varphi}_l^+, \boldsymbol{\varphi}_l^-)$ cam maps, which spatialize the positive and negative attribution in a given image.

response to the Imagenet validation set. The color of each point corresponds to the activation value (pre-ReLU), and the diagonal lines are contours lines for $E_l$ using the same color scale. Because $\boldsymbol{f_v} \approx E_l$, the coloring of the data points and the contour lines are well-aligned.

Some data points have been selected from Figure 4.a, with the corresponding images shown in Figures 4.b-f (cropped at the receptive field). The first and second of these selections correspond to the MEIs (b.) and MIIs (c.), which show excitatory and inhibitory curves at opposing orientations, as noted in Cammarata et. al. (2020)(Cammarata et al., 2020). A saliency map, similar to GradCAM (Selvaraju et al., 2019), accompanies each image, which serves to spatially localize our attributions. Observe that when layer $l$ is convolutional, attribution vector $\boldsymbol{S}_l(\boldsymbol{x}, \boldsymbol{f_v})$ is actually a tensor in $\mathbb{R}^{C\times H\times W}$. Spatial maps analogous to $E_l^{\pm}$ can be computed by summing over only the channel dimension,[1]

$$\boldsymbol{\varphi}^{\pm}(\boldsymbol{f_v}, \boldsymbol{x}) := \sum_{c=1}^{C} \text{ReLU}(\pm \boldsymbol{S}_{l,c}(\boldsymbol{f_v}, \boldsymbol{x})). \tag{3}$$

Such maps can be viewed as a heatmap over the image by upsampling them to the input image dimensions, as with GradCAM. We can visualize the negative and positive maps simultaneously by using a special color scale, which we show in Figure 4.g. With this coloring excitatory image regions appear red, inhibitory regions appear blue, and regions that have positive and negative terms in the channel dimension appear green.

[1] $E^{\pm}$ refers to the tuple $(E^+, E^-)$. Well use the superscript $\pm$ analogously for related variables.

These maps are normalized *across* images, making it possible these attribution maps to highlight no regions of the image. For example, Figure 4.d shows those inputs that have near zero $E_l^+$ and $E_l^-$. The cropped images show homogeneous color patches that neither excite nor inhibit the curve detector, and the attribution maps highlight nothing in these crops.

The images of most interest for our purposes are those which yield large values for both $E_l^+$ and $E_l^-$. Such images are depicted in Figure 4.e and .f, which each illustrate distinct cases. In the first case, inhibition and excitation to the curve detector come from distinct regions of the image. These images were selected from the set for which $\boldsymbol{\varphi}^+(\boldsymbol{x}) - \boldsymbol{\varphi}^-(\boldsymbol{x})$ contains both values $< P_1$ and $> P_{99}$. These images are easy to understand; they contain an excitatory curve at a consistent position/orientation, but also an inhibitory curve in a different location. The images in Figure 4.f depict a different case, in which simultaneous excitation/inhibition happens in the channel dimension at a single spatial position. These images were selected from the set for which both $\boldsymbol{\varphi}^+(\boldsymbol{x})$ and $\boldsymbol{\varphi}^-(\boldsymbol{x})$ contain a value $> P_{99}$ at the same spatial position. These images are harder to interpret, as the heatmap directs our attention to the same region to explain what excites and inhibits the curve detector. How should we understanding this channel-wise inhibition and excitation of a feature in the general case?

## 4. Accentuating and Inverting MTIs

For a given image, we want to exaggerate $\boldsymbol{S}_l^{\pm}$, which we'll do by treating these attributions themselves as the feature activations to be maximized. This is similar to a technique utilized in Cammarata et al. (2020), but isolating positive and negative attributions sep-

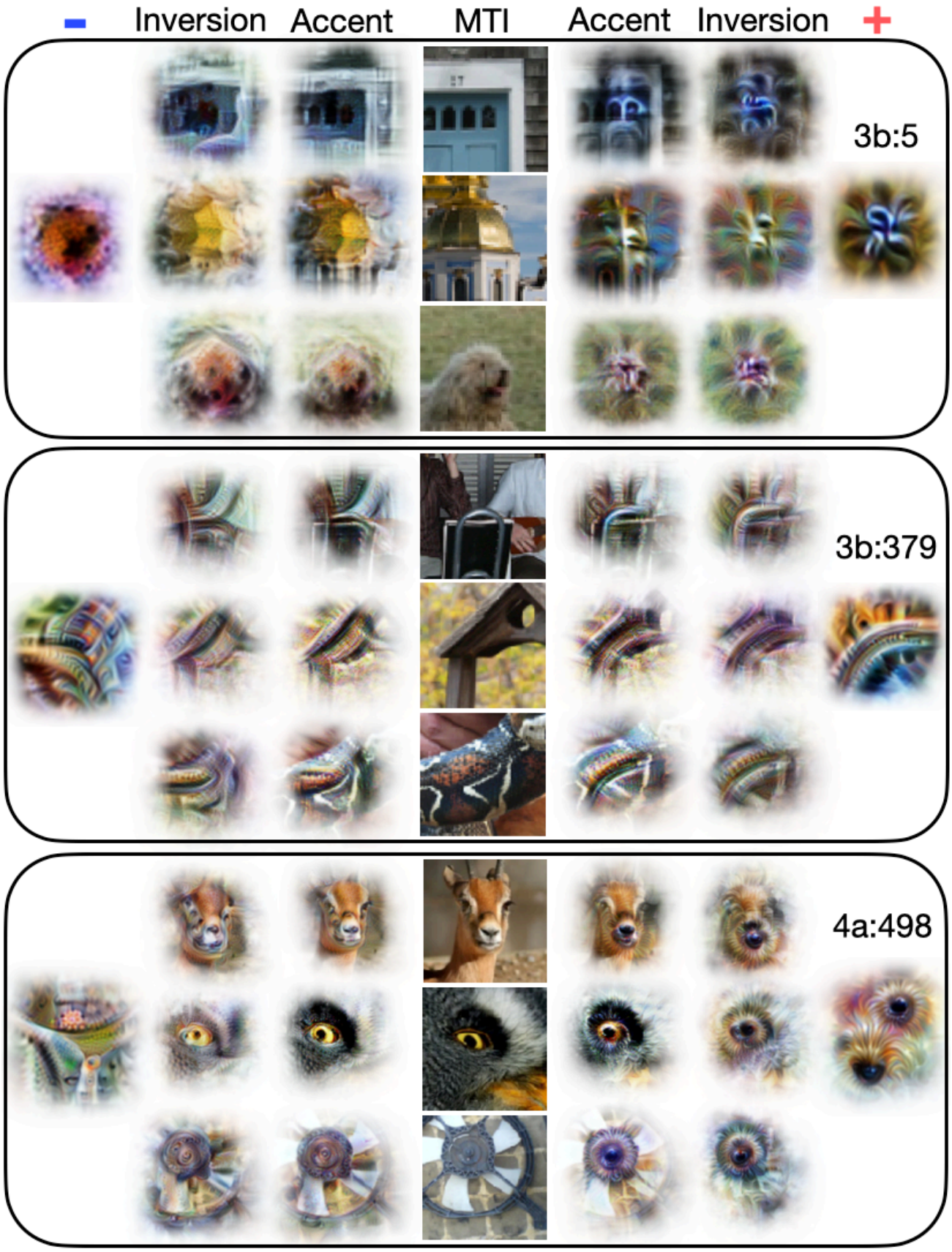


**Figure 5:** 3 MTIs for 3 units, with their ± attribution accentuations, inversions, and standard feature visualizations.

arately. Additionally, rather than maximize the dot product with the attribution vector, we'll scale the dot product by the cosine similarity as in Carter et al. (2019), which encourages the optimized image to point in the same direction as $\boldsymbol{S}_l^{\pm}$ (appendix C). As is typical with feature visualization, we can optimize the image with a parameterization $\mathcal{P}(\boldsymbol{x})$ (Mahendran & Vedaldi, 2015; Olah et al., 2017; Mordvintsev et al., 2018) and under a set of transformations (Mordvintsev et al., 2015) $\boldsymbol{\tau} \sim \mathcal{T}$, giving us the optimization;

$$\boldsymbol{z}^* = \arg\max_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{\tau} \circ \mathcal{P}^{-1}(\boldsymbol{z}); \boldsymbol{S}_l) \tag{4}$$

$$with \quad \mathcal{L}(\boldsymbol{x}; \boldsymbol{S}_l) := \frac{(\boldsymbol{f}_l(\boldsymbol{x}) \cdot \boldsymbol{S}_l)^{p+1}}{(||\boldsymbol{f}_l(\boldsymbol{x})|| \cdot ||\boldsymbol{S}_l||)^p} \tag{5}$$

We can view the image that results from this as $\boldsymbol{x}^* = \mathcal{P}^{-1}(\boldsymbol{z}^*)$, and optionally seed from noise (*inversion*) or the natural image (*accentuation*) we used to calculate $\boldsymbol{S}_l$. In Figure 5 we show accentuations and inversions for MTIs across 3 InceptionV1 units, including our curve detector from earlier. Each MTI is among the top 10 images with the largest norm, $||\boldsymbol{S}_l(\boldsymbol{x})||_1$, under the constraint that $.5\sigma(\boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{X})) < \boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{x}) < 0$, which ensures excitation and inhibition are balanced. We use the fourier phase space image parameterization from MACO(Fel et al., 2023a), as well as their opacity masking technique, which integrates the pixel gradients over the optimization steps and sets this to the alpha channel of the image. We use random crops that cover .9-.99 of the entire image, as well as uniform and gaussian noise, as our set of transformations.

## 5. Feature Attribution Atlas

Feature inversion and accentuation can help us understand how an individual image can excite and inhibit a feature, but what if we want a global view? That is, can we generate a visualization that depicts the relationships between all the various ways different images excite/inhibit a feature? Here we will adopt techniques from the *activation atlas*(Carter et al., 2019), which combines *UMAP*(McInnes et al., 2018) and feature visualization to map the space of activations of whole neural network layers. Here we propose the *Feature Attribution Atlas*, which conditions this technique on single features, mapping the space of feature attributions in a layer. Generating the atlas is a three step process.

**Image Selection:** The atlas should be a function of those images relevant to the target feature, which we can be determined by the its attributions. For a large set of images,$\mathcal{D} = \boldsymbol{x}_i^n$, we compute $\boldsymbol{S}_l(\mathcal{D}$, then select a subset of images $\boldsymbol{X}$ to construct our atlas. In this demonstration we use 100,000 ImageNet training images as $\mathcal{D}$ and select $\boldsymbol{X}$ as the top 10,000 with the largest $L^2$ attribution norm, $||S_l(\boldsymbol{x})||_2$. This selection criteria allows our atlas to convey those images which excite and/or inhibit $\boldsymbol{f}_{\boldsymbol{v}}$, but not those images which are strictly orthogonal. We find using the $L^2$ norm yields more diverse atlases than using $L^1$, see section 6 for details on why this might be the case.

**Attribution UMAP:** Next we organize images by their attribution vector using UMAP; $map_{\boldsymbol{S}} = umap(\boldsymbol{S}_l(\boldsymbol{X}))$. This map conveys the various 'reasons why' images excite/inhibit $\boldsymbol{f}_{\boldsymbol{v}}$, as points close in $map_{\boldsymbol{S}}$ will correspond to images with a similar attribution vector. We color points in the map by the correspond activation of $\boldsymbol{f}_{\boldsymbol{v}}$, to get a sense for where inhibitory and excitatory images land in the map (Figure 6.b).

**Feature Inversion:** It's difficult to visualize all $\tilde{1}$0,000 images represented in $map_{\boldsymbol{S}}$ directly, so we perform a coarse-graining operation. We overlay $map_{\boldsymbol{S}}$ with an $n \times n$ grid, then average all the attribution vectors within each grid cell, associating a new average attribution $S_{l,j,i}$ with each position $(i, j)$ in the grid. We can then generate a visualization for each position in the map using our attribution inversion technique (equation 4). Where in section 4 we visualized $\boldsymbol{S}_l^+$ and $\boldsymbol{S}_l^-$ independently, in this application we will optimize towards $|\boldsymbol{S}_{l,i,j}|$, which will allow positive and negative influences to be expressed in a single icon when in a 'tense' region of the map. Finally, we pass these icon images back through the network and compute $\boldsymbol{f}_{\boldsymbol{v}}$, then color the icon border with its activation value, and position the icons in the corresponding grid location, yielding the attribution atlas (Figure 6.c).

Viewing a feature's attribution atlas can provide much insight over dataset examples and feature visualizations alone. We find it particularly useful for understanding the role of negative weights into a feature. For example, mixed3b:9, appears to be inhibited by low frequencies, given its negative feature visualization, but as discussed earlier, its not sensible to conclude these weights are present *so that* the feature returns large negative activations to low frequency inputs. Rather we can understand the functional role of inhibition through the 'tense' regions of the attribution atlas (Figure 6.c), where inhibition and excitation meet. Inhibition ensure the feature returns only moderate activations in response to such inputs.

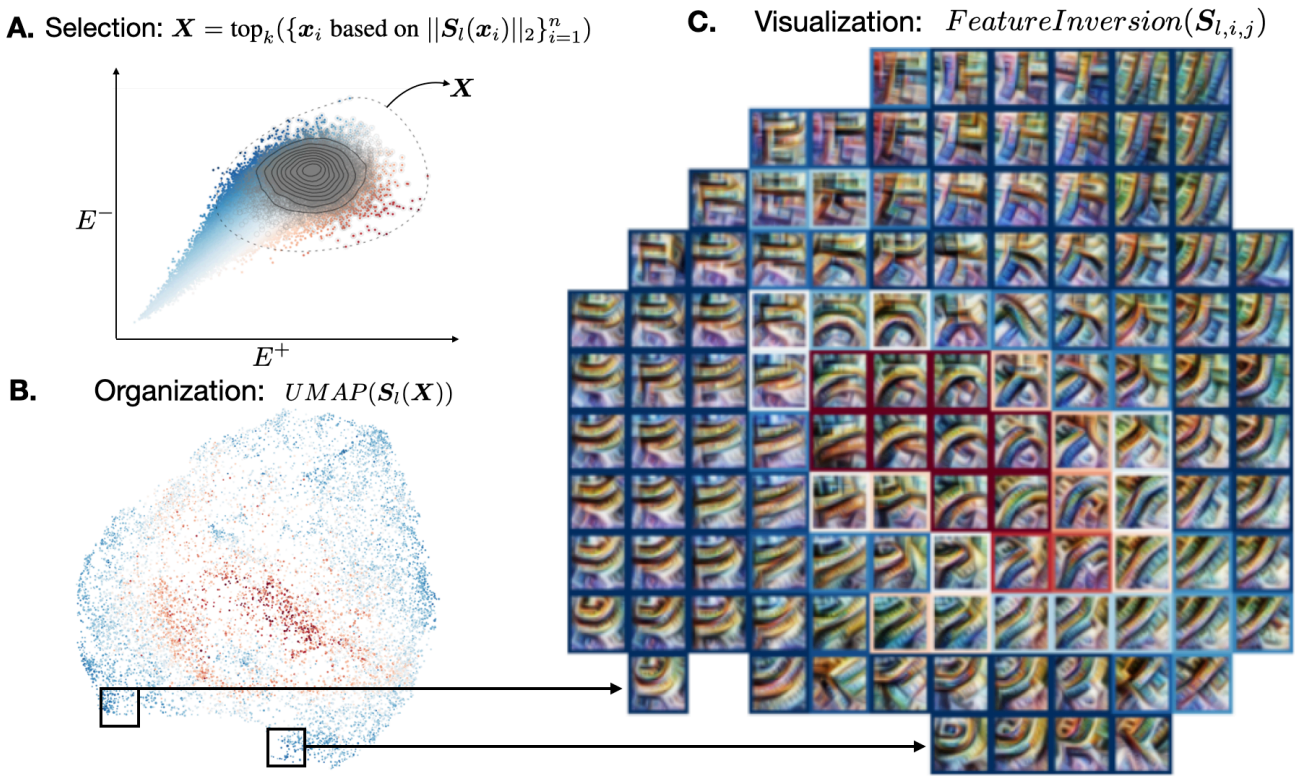


**Figure 6: A.** Selection of large attribution examples from the dataset shown over the plot of $E^{\pm}$ values. Contour lines show the density of selected points in the plot. **B.** Organize the attribution vectors of these selected points using UMAP. **C.** Average the attributions within local regions of the UMAP, then perform feature attribution inversion (equation 4). See appendix 18 for more example atlases on other features.

## 6. Inhibition from Superposition

We have presented a framework for understanding the role of inhibition in the construction of features. The lynch-pin of our approach is the attribution in intermediate layer $l$; inhibition plays a functionally relevant role in how $\boldsymbol{f_v}$ is computed for input $\boldsymbol{x}$ when $E_l^{\pm}(\boldsymbol{x}, \boldsymbol{f_v})$ are both large. Suppose though, that we found an image $\boldsymbol{x}$ expressing only features independent of $\boldsymbol{f_v}$, that nonetheless induced a large $E^{\pm}(\boldsymbol{x}, \boldsymbol{f_v})$. This would present a problem, as such an image expresses no features relevant to the computation of $\boldsymbol{f_v}$, and scrutinizing such an image may only mislead us. Recent work on toy models of superposition(Elhage et al., 2022b) suggests that such attributions are possible. Specifically, when features are sparse the model may represent more features than it has dimensions, Such that independent features are represented with *interference* (non-zero dot product) in a latent layer. It was hypothesized that reading out from features in superposition could necessitate inhibition negating the excitation caused by interference, in which case we would observe large $E_l^{\pm}(\boldsymbol{x}, \boldsymbol{f_v})$ even when all features expressed by $\boldsymbol{x}$ are independent of $\boldsymbol{f_v}$.

**Superposition in toy models.** Following Elhage et al. (2022b), let's test that these deceptive attributions are possible with a toy model computing the absolute value function, which can be computed by a ReLU neural network as $\text{abs}(x) = \text{ReLU}(x) + \text{ReLU}(-x)$. For multivariate input $\boldsymbol{x} \in \mathbb{R}^n$, the network requires $m = 2n$ hidden neurons to compute $\text{abs}(\boldsymbol{x})$ exactly. In the original work, the authors find the network will optimize for very different weight motifs when $m < 2n$ and $\boldsymbol{x}$ is sampled sparsely, such that any given $x_i$ usually takes the value 0. In our replication of these experiments, we train 4 models to compute $\text{abs}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{R}^6$, with hidden dimensions $m = 12, 10, 8, 6$. As $m$ decreases, the 6 input features should be represented with more interference in the hidden layer.

For each trained model we pass 12 inputs, one-hot vectors for each of the features and their negatives, i.e the basis vectors $e_i$ and $-e_i$. For each of these inputs we compute the attribution to each output feature, $f_i$, which we've trained to compute $f_i(\boldsymbol{x}) = \text{abs}(x_i)$. These attribution matrices are shown in Figure 7; element $(i, j)$ shows $E_l^{\pm}(e_i, f_j)$ in the hidden layer, colored according the legend in Figure 4.b. The matrix in

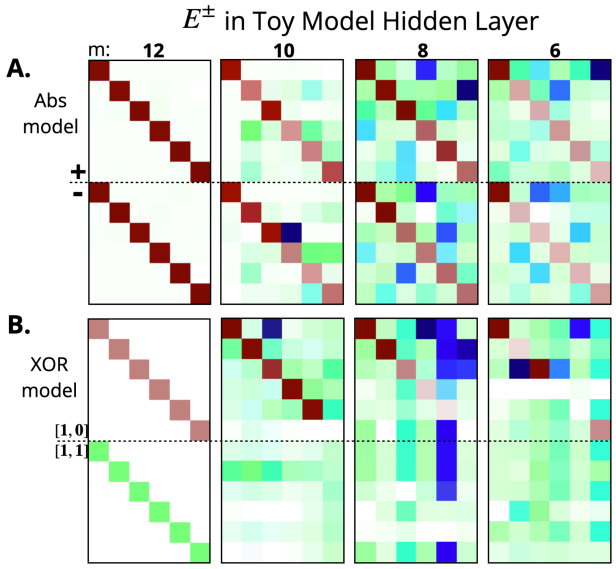


**Figure 7:** Toy models compute the (**A.**) absolute value function and (**B.**) XOR function on 6 independent input features. Attributions are normalized within each column. For a color-scale legend, see Figure 4.b

the first column corresponds to a disentangled model that can compute absolute value exactly without feature interference. The matrix shows + attribution along the diagonal only, and 0 attribution everywhere else, meaning input $e_i$ excites $f_i$, but $f_j$ is insensitive to $e_i$ whenever $i \neq j$, as it should be. Subsequent columns show attribution in models with ever tighter bottlenecks, inducing more interference, and subsequently more off-diagonal attributions. All of this off-diagonal inhibition and excitation is induced by independent features irrelevant to the computation of $f_j$.

It's clear that superposition can induce the +/- attribution motif, however we know that inhibition is not only relevant for compression, as there are functions that provably cannot be approximated without negative weights, such as the xor function(Wang et al., 2023). Figure 7.b shows a similar experiment on a toy model computing the XOR function over 6 independent *pairs* of features, flattened into an input in $\mathbb{R}^{12}$. Instead of testing this model on inputs $\pm e_i$, we pass the inputs $[1,0]_i$ and $[1,1]_i$ – one element relevant to $f_i$ is on, or both – for which the model should return $e_i$ and $\mathbf{0}$ respectively. In this case the disentangled model shows simultaneous inhibition and excitation to $f_i$ from input $[1,1]_i$; excitement from the latent 'or' feature and inhibition from the latent 'and' feature. This is the sort of functionally relevant inhibition we hoped to identify with MTIs, where inhibition cancels the effect of excitation, leading to no activation of the downstream feature. However, when this XOR model is implemented with a latent bottleneck, we see large off diagonal attributions just as before.

**Superposition at scale** How can we distinguish between instances of functionally relevant inhibition – that is necessary for computing $\boldsymbol{f_v}(\boldsymbol{x})$ even in a disentangled model – and instances of inhibition caused by superposition? In toy models its possible to make this distinction, because we know a priori how the features can be computed from the inputs. However, in full-scale object recognition models we don't know the ground-truth for how any feature is computed, that is precisely what we are trying to uncover empirically. One indication that a large attribution norm might be the result of superposition interference is it should affect multiple features. In the limiting case, a maximally superimposed feature in layer $l$ points in the diagonal direction, and *every feature* in layer $l+1$ is excited and inhibited (presuming it has both positive and negative incoming weights) whenever this diagonal feature is expressed. Additionally, it has been theorized that superposition is less likely to occur in the early layers of an object recognition model, because the features represented in such layers, such as edges, are not sparse(Elhage et al., 2022b).

Taking these two observations together, we should expect that in deeper layers of the network, many features show a high attribution norm to the very same images, as these images express highly interfering features. To test this, we consider a uniqueness metric across a sample of features in a layer. Suppose we have $n$ features, and a set of $m$ images, $X_i$, is identified with each feature, $\boldsymbol{f}_i$, by some selection process. We can define a *uniqueness* measure for this process as;

$$U = \frac{|\bigcup_{i=1}^{n} X_i|}{nm} \tag{6}$$

Observe that U is bounded above by 1, when every image selected is unique, and bounded below by $\frac{1}{n}$, when $\forall i, j \in \{1, \ldots, n\}$, $X_i = X_j$.

In Figure 8.a we see how different image selection processes differ in this uniqueness measure. The red solid line shows the uniqueness of *MEI*s across layers of InceptionV1 for 20 random units per layer, for which $U$ is near ceiling across all layers. However, when those images with a large attribution norm in the preceding layer are identified with a unit, rather than those with a large activation, uniqueness decreases substantially in the deeper layers, as we predicted under the superposition hypothesis. Using the $L^2$ norm of the attribution vector (light-green line) rather than the $L^1$

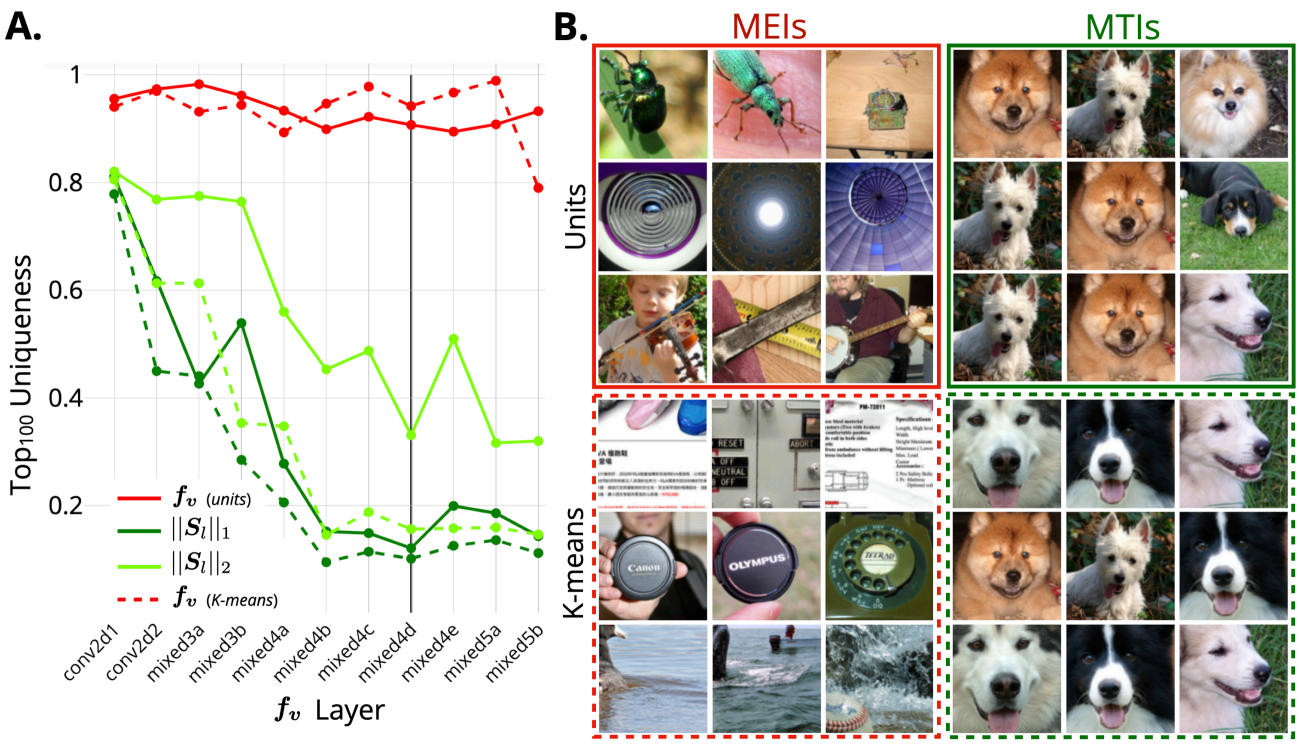


**Figure 8: A.** While features may show large activations for different images (red line), in deeper layers of the model they show large attributions to the same images, (green lines). This is true of unit features and k-means features (dotted lines). **B.** Each row corresponds to a random feature in layer mixed4d, with three MEIs shown on the left, and MTIs on the right. Even when MEIs convey unique semantics across features, MTIs may be shared (in this layer 'dog faces' yield large attribution across all features).

(dark-green line), mitigates this somewhat, as the $L^2$ norm grows less quickly than the $L^1$ in off-basis directions(Xu et al., 2021; Yin et al., 2014), making $L^2$ potentially less likely to select for superimposed features in the attribution layer. In each row of Figure 8.b we show the top-3 MEIs and MTIs (largest $||\boldsymbol{S}||_1$) in the Imagenet validation set for random features in layer Mixed4d. It's qualitatively clear that even when MEIs convey unique semantics for a feature, the MTIs may convey a global, superimposed concept – a 'dog face' in the case of layer Mixed4d. See the appendix H for examples of those images that yield non-unique attributions in different layers of the model.

## 7. Limitations & Conclusion

Inhibition and excitation in ReLU neural networks do not function symmetrically, but currently the few interpretability tools that target inhibition do not account for this. We introduce several new techniques to the toolbox that respect this asymmetry, by conditioning our understanding of inhibition on excitation through the analysis of *maximally tense images*. Our novel visualization techniques reveal how inhibitory connections prevent erroneous activation in response to MTIs, by isolating the inhibitory and excitatory attributes simultaneously present in such images. However, we also show that superposition currently introduces a major obstacle for these kinds of analyses, as networks use negative weights to facilitate a compression algorithm, representing more features than units. Given this, a 'clean' understanding of inhibition in deep layers of vision models will likely require the development of techniques for 'monosemantic disentanglement', as is being pursued for large language models(Bricken et al., 2023; Cunningham et al., 2023). Additionally, the inhibitory mechanisms described in this work need not be the only ones through which features are suppressed. For example, the final softmax layer of a classifier constitutes a different mechanism by which one 'class' feature inhibits another. Other model architectures could invoke similar suppressive mechanisms throughout, such as those using Top-k (Makhzani & Frey, 2013; Ahmad & Scheinkman, 2019) or SoLU (Elhage et al., 2022a) activation functions. In conclusion, we hope this work prompts more exploration into the inhibitory mechanisms latent in vision models, and their role in feature construction.

## Impact Statement

This paper presents work whose goal is to make neural networks more interpretable, a basic research goal we expect to have only positive social impact.

## References


Ahmad, S. and Scheinkman, L. How can we be so dense? the benefits of using highly sparse representations. *arXiv preprint arXiv:1903.11257*, 2019.

Araujo, A., Norris, W., and Sim, J. Computing receptive fields of convolutional neural networks. *Distill*, 2019. doi: 10.23915/distill.00021. https://distill.pub/2019/computing-receptive-fields.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *Public Library of Science (PloS One)*, 2015.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Borowski, J., Zimmermann, R. S., Schepers, J., Geirhos, R., Wallis, T. S., Bethge, M., and Brendel, W. Exemplary natural images explain cnn activations better than state-of-the-art feature visualization. *arXiv preprint arXiv:2010.12606*, 2020.

Bricken, T., Templeton, A., Batson, J., Chen, B., Jermyn, A., Conerly, T., Turner, N., Anil, C., Denison, C., Askell, A., Lasenby, R., Wu, Y., Kravec, S., Schiefer, N., Maxwell, T., Joseph, N., Hatfield-Dodds, Z., Tamkin, A., Nguyen, K., McLean, B., Burke, J. E., Hume, T., Carter, S., Henighan, T., and Olah, C. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Cammarata, N., Goh, G., Carter, S., Schubert, L., Petrov, M., and Olah, C. Curve detectors. *Distill*, 2020. doi: 10.23915/distill.00024.003. https://distill.pub/2020/circuits/curve-detectors.

Cammarata, N., Goh, G., Carter, S., Voss, C., Schubert, L., and Olah, C. Curve circuits. *Distill*, 2021. doi: 10.23915/distill.00024.006. https://distill.pub/2020/circuits/curve-circuits.

Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. Activation atlas. *Distill*, 4(3):e15, 2019.

Cun, Y. L., Denker, J. S., and Solla, S. A. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pp. 598–605. Morgan Kaufmann, 1990.

Cunningham, H., Ewart, A., Riggs, L., Huben, R., and Sharkey, L. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Elhage, N., Hume, T., Olsson, C., Nanda, N., Henighan, T., Johnston, S., ElShowk, S., Joseph, N., DasSarma, N., Mann, B., Hernandez, D., Askell, A., Ndousse, K., Jones, A., Drain, D., Chen, A., Bai, Y., Ganguli, D., Lovitt, L., Hatfield-Dodds, Z., Kernion, J., Conerly, T., Kravec, S., Fort, S., Kadavath, S., Jacobson, J., Tran-Johnson, E., Kaplan, J., Clark, J., Brown, T., McCandlish, S., Amodei, D., and Olah, C. Softmax linear units. *Transformer Circuits Thread*, 2022a. https://transformer-circuits.pub/2022/solu/index.html.

Elhage, N., Hume, T., Olsson, C., Schiefer, N., Henighan, T., Kravec, S., Hatfield-Dodds, Z., Lasenby, R., Drain, D., Chen, C., Grosse, R., McCandlish, S., Kaplan, J., Amodei, D., Wattenberg, M., and Olah, C. Toy models of superposition. *Transformer Circuits Thread*, 2022b.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., and Madry, A. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

Fel, T., Cadene, R., Chalvidal, M., Cord, M., Vigouroux, D., and Serre, T. Look at the variance! efficient black-box explanations with sobol-based sensitivity analysis. 2021.

Fel, T., Boissin, T., Boutin, V., Picard, A., Novello, P., Colin, J., Linsley, D., Rousseau, T., Cadène, R., Gardes, L., et al. Unlocking feature visualization for deeper networks with magnitude constrained optimization. *arXiv preprint arXiv:2306.06805*, 2023a.

Fel, T., Boutin, V., Moayeri, M., Cadène, R., Bethune, L., andéol, L., Chalvidal, M., and Serre, T. A holistic approach to unifying automatic concept extraction and concept importance estimation, 2023b.

Fel, T., Ducoffe, M., Vigouroux, D., Cadène, R., Capelle, M., Nicodème, C., and Serre, T. Don't lie to me! robust and efficient explainability with verified perturbation analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16153–16163, 2023c.

Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., and Serre, T. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2711–2721, 2023d.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, 2017.

Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. Towards automatic concept-based explanations. 2019.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Klindt, D., Sanborn, S., Acosta, F., Poitevin, F., and Miolane, N. Identifying interpretable visual features in artificial and biological neural systems. *arXiv preprint arXiv:2310.11431*, 2023.

Lee, N., Ajanthan, T., and Torr, P. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2018.

Lundberg, S. and Lee, S.-I. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5188–5196, 2015.

Makhzani, A. and Frey, B. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL `http://arxiv.org/abs/1802.03426`. cite arxiv:1802.03426Comment: Reference implementation available at http://github.com/lmcinnes/umap.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017-Conference Track Proceedings*, 2019.

Mordvintsev, A., Olah, C., and Tyka, M. Inceptionism: Going deeper into neural networks. 2015.

Mordvintsev, A., Pezzotti, N., Schubert, L., and Olah, C. Differentiable image parameterizations. *Distill*, 2018.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., and Clune, J. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *Advances in neural information processing systems*, 29, 2016a.

Nguyen, A., Yosinski, J., and Clune, J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *Visualization for Deep Learning workshop, Int. Conf. Machine Learning*, 2016b.

Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., and Yosinski, J. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4467–4477, 2017.

Novello, P., Fel, T., and Vigouroux, D. Making sense of dependence: Efficient black-box explanations using dependence measure. 2022.

Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. Zoom in: An introduction to circuits. *Distill*, 2020a. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Olah, C., Cammarata, N., Voss, C., Schubert, L., and Goh, G. Naturally occurring equivariance in neural networks. *Distill*, 2020b. doi: 10.23915/distill.00024.004. https://distill.pub/2020/circuits/equivariance.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *BMVC*, 2018.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Knowledge Discovery and Data Mining (KDD)*, 2016.

Santurkar, S., Ilyas, A., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Image synthesis with a single (robust) classifier. *Advances in Neural Information Processing Systems*, 32, 2019.

Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., and Batra, D. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *IJCV*, abs/1610.02391, 2019. URL `http://arxiv.org/abs/1610.02391`.

Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not just a black box: Learning important features through propagating activation differences, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop, ICLR*, 2013.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, Int. Conf. Machine Learning*, 2017.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017a.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *Int. Conf. Machine Learning*, 2017b.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

Tyka, M. Class visualization with bilateral filters. 2016. *URL: https://mtyka. github. io/deepdream/2016/02/05/bilateral-class-vis. html*, 2(3).

Wang, Q., Powell, M. A., Geisa, A., Bridgeford, E., Priebe, C. E., and Vogelstein, J. T. Why do networks have inhibitory/negative connections? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22551–22559, 2023.

Wei, D., Zhou, B., Torrabla, A., and Freeman, W. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

Xu, Y., Narayan, A., Tran, H., and Webster, C. G. Analysis of the ratio of 1 and 2 norms in compressed sensing. *Applied and Computational Harmonic Analysis*, 55:486–511, 2021.

Yin, P., Esser, E., and Xin, J. Ratio and difference of $l\_1$ and $l\_2$ norms and sparse representation with coherent dictionaries. *Communications in Information and Systems*, 14(2):87–109, 2014.

Zhang, R., Madumal, P., Miller, T., Ehinger, K. A., and Rubinstein, B. I. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. *arXiv preprint arXiv:2006.15417*, 2020.

Zimmermann, R. S., Borowski, J., Geirhos, R., Bethge, M., Wallis, T., and Brendel, W. How well do feature visualizations support causal understanding of cnn activations? *Advances in Neural Information Processing Systems*, 34:11730–11744, 2021.

Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.

# Appendix

## A. Code

Code for this project is available at this github repository.

## B. Human Experiment

Here we explain a human experiment in which we endeavor to identify if/when people can relate the MEIs and MIIs of a feature. Given visual inspection of such exciting/inhibiting images, like those shown in figure 2, we hypothesize that early on in the model humans will be able to conceive of a feature's 'opposite' given example MEIs, but wont be able to do so in later layers. To test this we conducted an experiment in which participants are shown a set of MEIs for a feature, then must use this information to identify the feature's MII from another set.

In this experiment we test 15 features per an early, mid, and late layer of InceptionV1 – Conv2d0, Mixed4a, and Mixed5a. We use the centroids of k-means clusters as our feature directions, following the observation in Klindt et al. (2023) that shows such k-means features are particularly interpretable for humans in this experimental paradigm. In particular, we first specify 1000 clusters per layer using the SKlearn k-means clustering algorithm with a cosine distance metric. The clustering was applied to each layers' hidden vectors in response to the ImageNet validation set, with each image sampled at a random position in the activation map for each hidden vector. We chose 15 random features from these centroids per layer for use in this experiment. We defined the activation for such a feature in response to an image as the cos*dot (see section C) of the hidden vector for the image in the corresponding pre-relu layer, computed in the channel dimension. We use $p = 2$ cosine power, which ensures cosine and dot product terms do not cancel out negatives when multiplied together.

For each feature identified this way we construct a trial of the experiment, which consists of 15 image (crops). First, we compute activations for the feature in response to the ImageNet validation set. The image that induces the largest activation we specify as the trial's 'target MEI', which is cropped to the effective receptive field (Araujo et al., 2019) that induced the large activation. Similarly the smallest activation is specified as the 'target MII'. The 2nd-7th largest activations are specified as the 'example MEIs'. Finally 7 'distractor' image crops are selected, which each satify the simultaneous constraints of yielding only modest activation for the feature – within 2 standard deviations of the mean – and having large over all activation – the hidden vector for the crop in the feature's layer has an $L^1$-norm in the 90th percentile. This second constraint ensures distractors are not different in kind from the target MEI and MII, conveying salient objects rather than awkward crops or background elements. Across trials of features in a given layer, we apply the additional constraints that crops cannot be repeated for any MII or distractor, and crops cannot come from different locations of the same image.

In a given trial of the experiment, the participant is shown the set of 6 'example MEIs' for the feature on the left. They are then shown a set of 9 images on the right; the 7 'distractors', the 'target MEI', and the 'target MII'. The participant is first asked to select which image on the right they believe to be the MEI. Next, they are asked to select the image they believe to be the MII. They are given feedback as to the correct choices after every trial. Each participant completes the 15 trials corresponding to those features in a single layer before moving on the next layer. An example trial showing what the participant sees when selecting the MEI and MII is shown in Figure B. We recruited 48 participants for this experiment through Prolific (www.prolific.com). They were paid $3 to complete the ~15 minute experiment.

The results of this experiment are in agreement with our hypothesis, and can be seen in Figure B. Specifically, participants were able to extrapolate from the set of MEIs to the new MEI well across all layers. However, participants could not easily extrapolate from MEIs to their 'opposite' MII. Participants performed significantly better ($p \approx 1e-5$) when identifying MIIs in the first layer, Conv20, than the middle and late layers, Mixed4a and Mixed5a.

## C. Dot*Cosine Objective

Figure C shows visually the motivation for the dot*cosine objective used for attribution visualizations. The dot product can optimize for the hidden vector $\boldsymbol{h}$ to simply have a large magnitude, but not really point in the direction $\boldsymbol{v}$. Conversely, cosine similarity can be maximized by inputs with a very low magnitude, which aren't salient to the network. The cosine*dot objective optimizing for $\boldsymbol{h}$ to have a large magnitude and point in the direction $\boldsymbol{v}$.

## D. model weight distribution

A slight majority of weights are negative consistently across Imagenet trained models, but the function of these weights has received significantly less treatment in the literature.

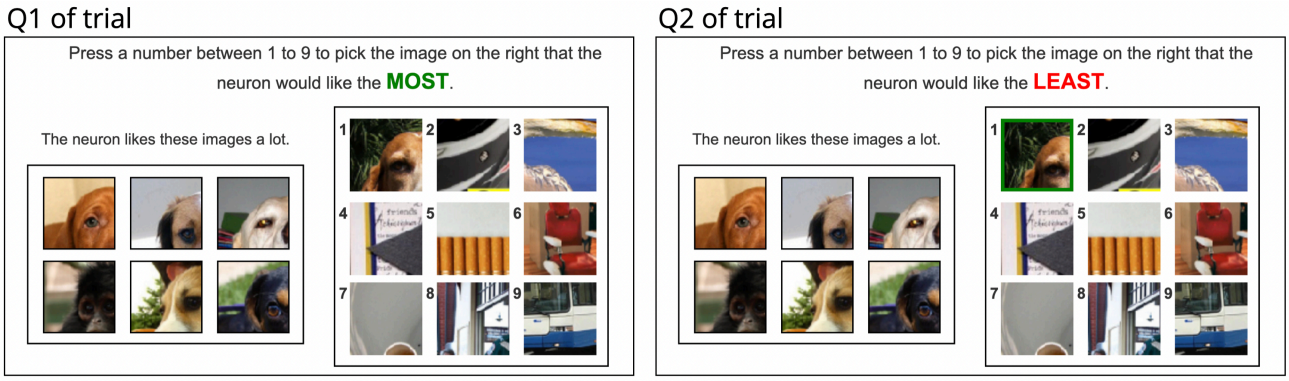


**Figure 9:** An example trial of our human experiment, in which participants must extrapolate from a set of MEIs to both a new MEI and an MII. In Q2 of the trial, the green outline for choice 1 indicates that it was selected as the new MEI. Participants could not choose the same image as the MEI and MII.

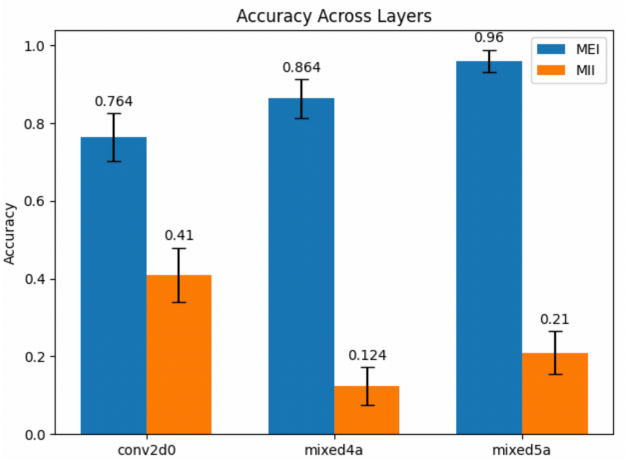


**Figure 10:** Human accuracy in predicting MEI/MIIs across model layers.

## E. Other measures of Attribution Completeness

In section 2 we show a ceiling Pearson correlation between model logits and the summed attribution through many preceding layers of the model latent space. Pearson correlation is a good metric as the logit activations are approximately normally distributed, and we dont care about the overall scale of the attributions for our purposes of selecting MTIs. That said, here we report some other distance metrics between the attributions and logit activations; **A.** Spearman correlations and **B.** absolute (L1) distance. Additionally in **C.** we show the original Pearson metric, but measured on models with batch normalization layers. Passing through batch normalization layers causes the correlation between attribution and logit activation to drop significantly. For all these plots, and that in Figure 3, we show a standardized 'model depth' on the X axis in the range $[0, 1]$, which corresponds to the ratio of ReLU non-linearities preceding the layer over the total in the model.

## F. Do Attribution Inversions Actually Excite/Inhibit?

Attribution inversion is suppose to tell us something about feature $\boldsymbol{f_v}$, but optimizes an objective based on the attribution feature vector in an earlier layer $\boldsymbol{S}_l^{\pm}$. Here we conduct a simple sanity check to confirm optimizing for $\boldsymbol{S}_l^{\pm}$ has the expected effect of exciting/inhibiting $\boldsymbol{f_v}$. To test this, for 5 layers of InceptionV1 we choose 20 random units, then compute inhibitory and excitatory attribution inversion through the preceding layer across for 10 of the unit's MTIs. Figure 14 shows the activation these inversions induce in the target feature across optimization steps. The shaded region corresponds to the full range of results, and the lines correspond to the median. We plot activation on the y-axis in standard deviations for the target unit. The figure shows that attribution inversions indeed have the desired effect.

## G. Toy model details

Architecturally our absolute value toy model and xor toy model are nearly identical, differing only in their input dimensions – 6 for the abs model but 12 the xor model, as each of the 6 output features is a function of a pair of inputs. From these inputs $\boldsymbol{x}$, a hidden vector $\boldsymbol{h}$ is computed, and then the output $\boldsymbol{f}'$ as follows;

$$\boldsymbol{h} = \text{ReLU}(W_1\boldsymbol{x} + b_1) \tag{7}$$
$$\boldsymbol{f}' = \text{ReLU}(W_2\boldsymbol{h} + b_2) \tag{8}$$

These models were trained in a manner similar to that

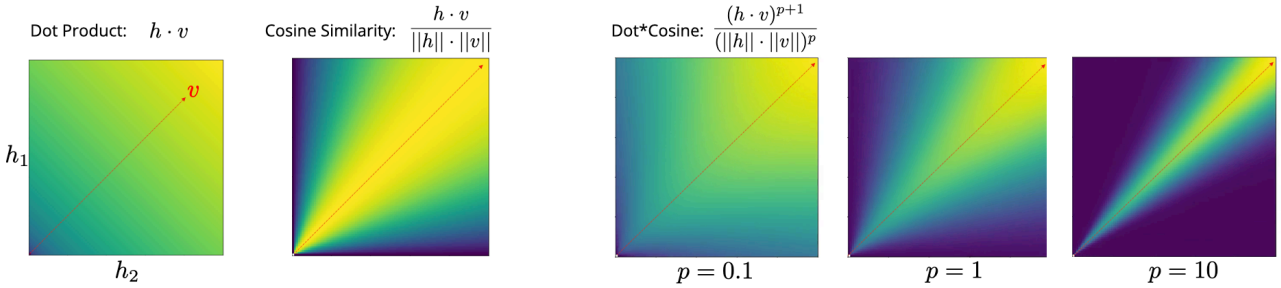


**Figure 11:** A visual intuition for the dot*cosine loss function

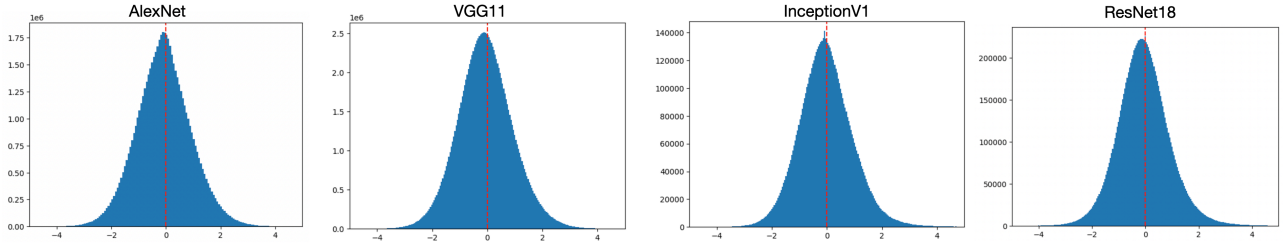


**Figure 12:** Across Imagenet trained models, we see a very similar weight distribution, with a slight majority of weights being negative in all cases. weights are standardized to $\sigma = 0$ in each layer before being aggregated in each histogram.

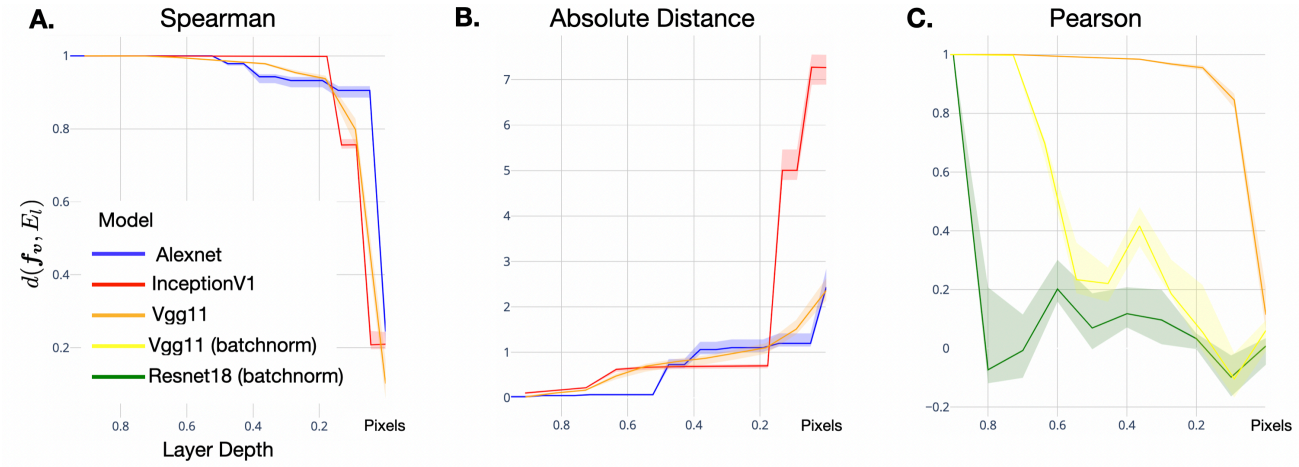


**Figure 13:** **A.** The Spearman correlation between logits and total attribution across layers, which is indistinguishable from the Pearson correlation (Figure 3). **B.** The absolute distance (L1) between the attribution and logit activation. **C.** The Pearson correlation measured on models with batch normalization layers. VGG11 (no batch normalization) is displayed again in this plot for reference to VGG11 (with batch normalization).

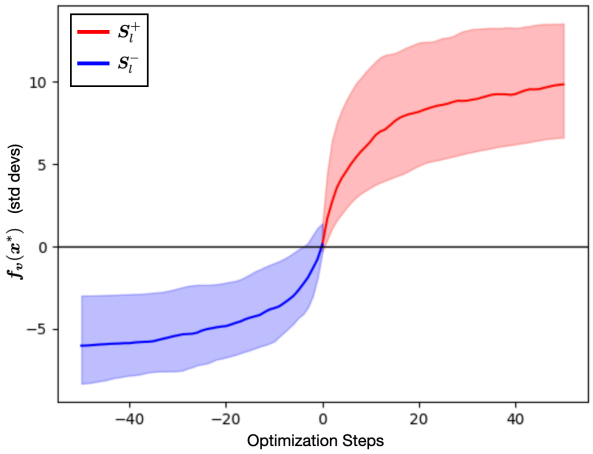


**Figure 14:** Activations induced in the target feature by optimizing towards $S_l^-$ and $S_l^+$

in Elhage et. al. (2022)(Elhage et al., 2022b). First, for the absolute value model, $\boldsymbol{x} is sampled such that x_i$ has a .99 probability of being 0 (it is sparse), otherwise it is sampled uniformly from $[0, 1]$. Each feature receives an 'importance' $I_i = .9^i$, so the loss function can weight important features more heavily. We train the model using the mean squared error from the target function, $f_i = \text{abs}(x_i)$;

$$\mathcal{L} = \sum_{i=1}^{6} I_i (f_i - f_i')^2 \tag{9}$$

We train on batches of 600 inputs for 20000 iterations using the Adam optimizer (Kingma & Ba, 2017), with learning rate .001. As in the original work we find the model is hard to train on the sparse input signal, and thus train from 1000 different random seeds, picking the top performing model. We repeat this process for the 4 hidden dimensionalities tested; $m = 12, 10, 8, 6$.

For the XOR toy model the procedure is largely the same as above, the main difference is how we sample the input. As before, each feature $f_i$ is sampled with some probability $S$ of being 'off', or 0 in the input. However, in this case when a feature passes this sampling filter and is 'on', it corresponds to a pair of elements in the input, $[x_{2i}, x_{2i-1}]$, which are each sampled independently from a Bernoulli distribution, with $p = .5$. This sampling procedure introduces additional sparsity over the absolute value features, so we use a $S = .95$ for sampling XOR features. In this case our target function is $f_i = \text{XOR}(x_{2i}, x_{2i-1})$, and loss is computed as before (equation 9. All other training details are identical to the absolute value toy models.

It is important we ensure our toy models are actually performant, if we are to take anything from the results in Figure 7. To test this, for each model, we pass inputs that span the domain and compute the resultant loss. For the absolute values models these inputs are every combination of elements $\{0, -1, 1\}$ in 6 dimensions. For the XOR models, we test every combination of elements $\{0, 1\}$ in 12 dimensions. We show these input-wise losses in Figure G as box plots, organized by the toy model hidden dimensions and the number of 'active features'– i.e. the number of 1s in the output computed by the ground-truth absolute value and XOR functions. We see the the disentangled models, with 12 hidden units, are perfect, incurring no loss for any of the inputs. As the hidden dimensionality decreases however, both models incur loss as they cannot faithfully represent the target function. Of note, this loss monotonically increases with the number of active features, in agreement with the theory that features in superposition can be faithfully represented when either is present in the input, but not simultaneously present. The inputs used for Figure 7 all have 1 or 0 active features, and the models are performant over these inputs.

## H. Uniqueness Experiment

K-means directions in this experiment were defined as the cluster centroids determined by the SKlearn k-means clustering algorithm using the cosine distance metric. The clustering was applied to each layers' hidden vectors in response to the ImageNet validation set, each sampled at a random position in the layer's activation map. We first used k=1000 centroids, to define a general basis of many features. To get our 20 sampled features for the experiment, we ran k-means again on these centroid vectors using k=20, then selected a random vector from the original 1000 from each of these new clusters. We did this to ensure the features were sampled from a reasonably sized basis (larger than 20), but also not too close to each other in direction. Gradients in this experiment were computed with respect to the central position in each activation map.

We found in this uniqueness experiment that across features in later layers, the very same images yielded high attributions. We argued that these images expressed features represented in superposition in the attribution layer. Viewing the corresponding images, which are shown in figure 16 helps lend some credence to this view. Each (receptive field cropped) image was among the top-100 largest attributions for at least 15 out of the 20 features sampled. These images show single salient concepts, like 'dog-face', 'bird', or 'emergency vehicle', rather than many concepts. This motivates our hypothesis that these images load onto many units because they express features the model has placed in

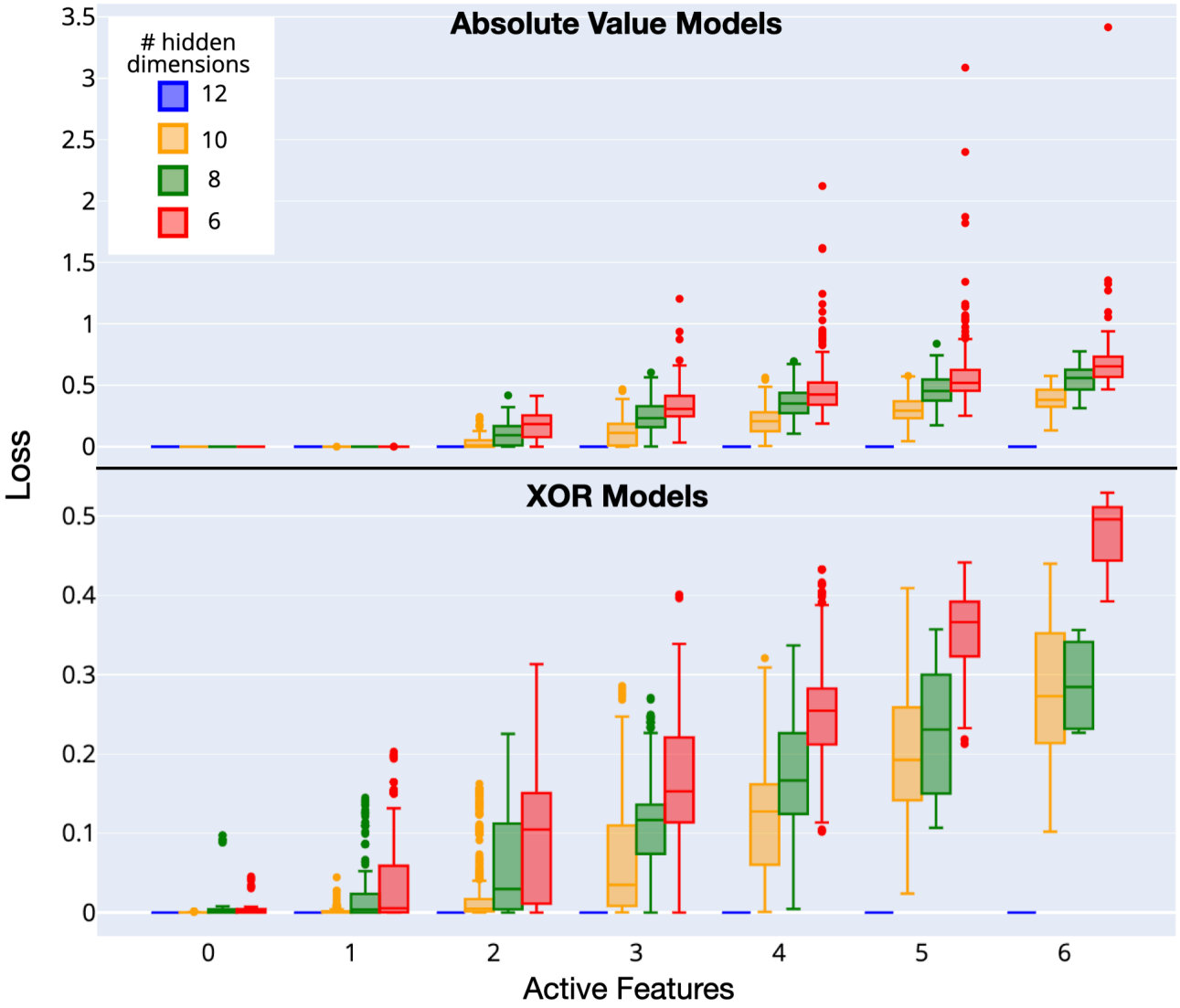


**Figure 15:** Losses for the Absolute Values and XOR models across inputs with differing number of active features. Where disentangled models $m = 12$ are perfect for all inputs, decreasing the hidden dimensions increases the loss. In particular, loss is greater the more features are present in the input.

superposition, rather than the alternative, that they simply express many different features at once.

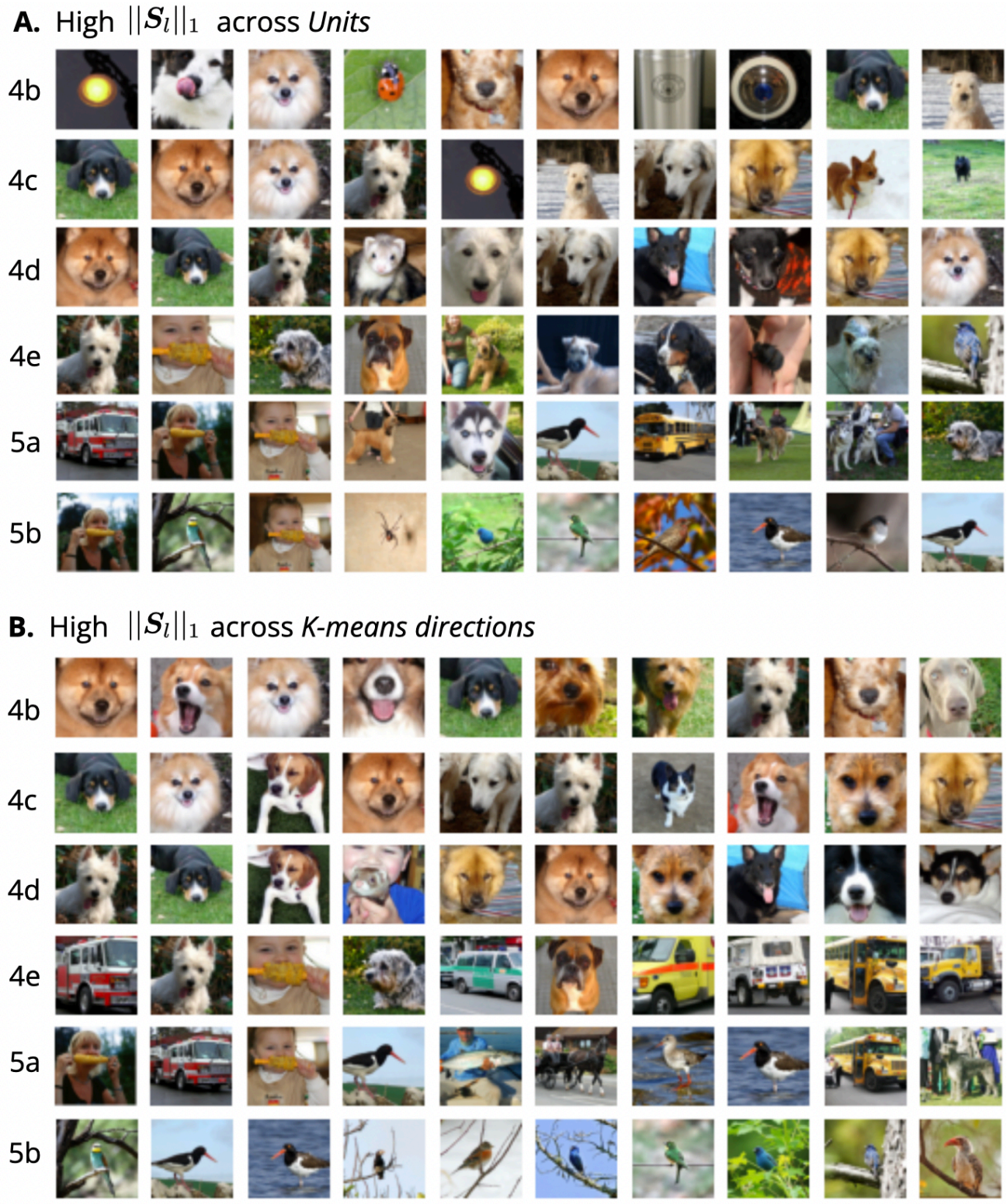


**Figure 16:** Images with large attributions across **A.** unit, and **B.** k-means features.

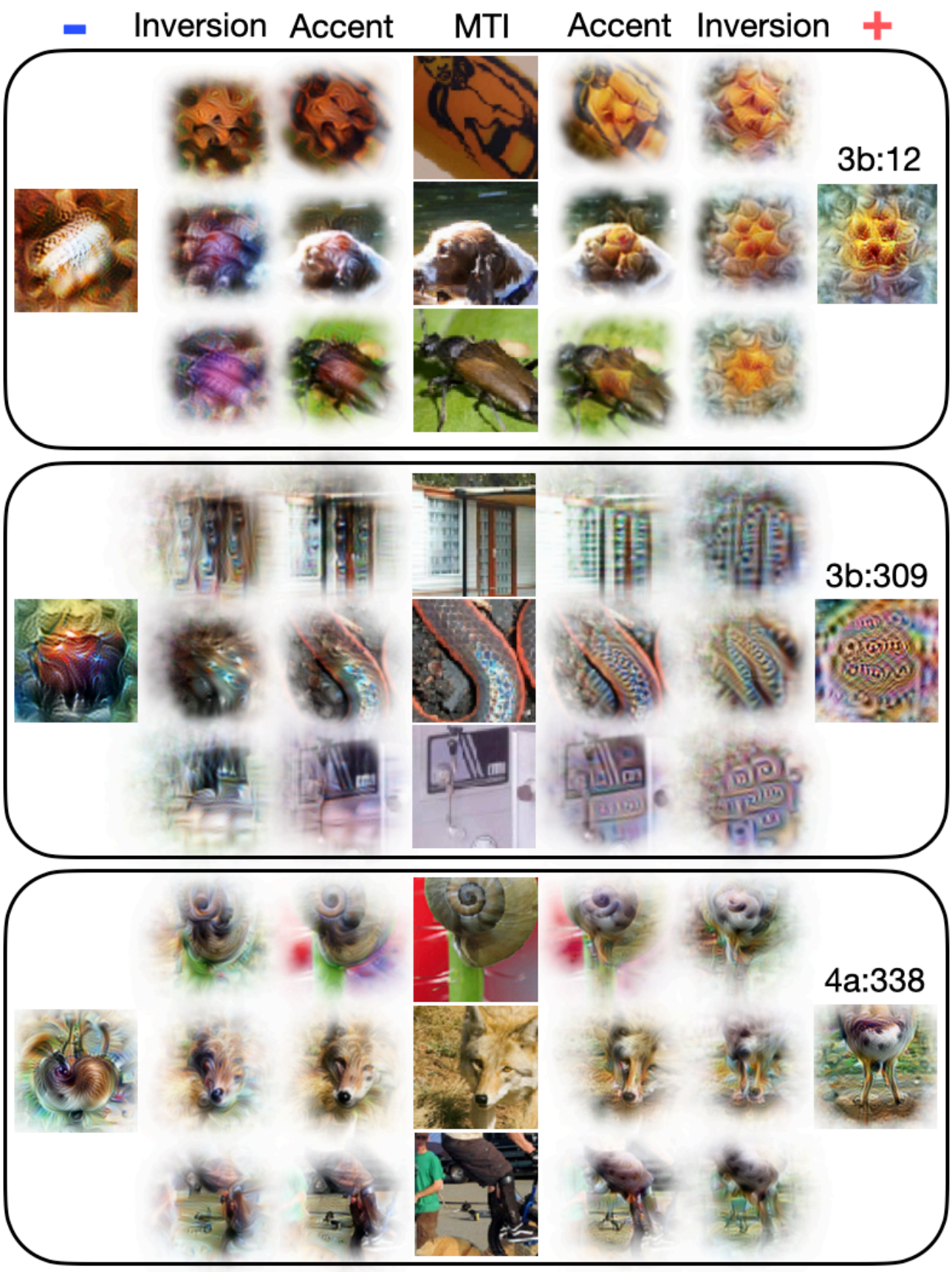


**Figure 17**

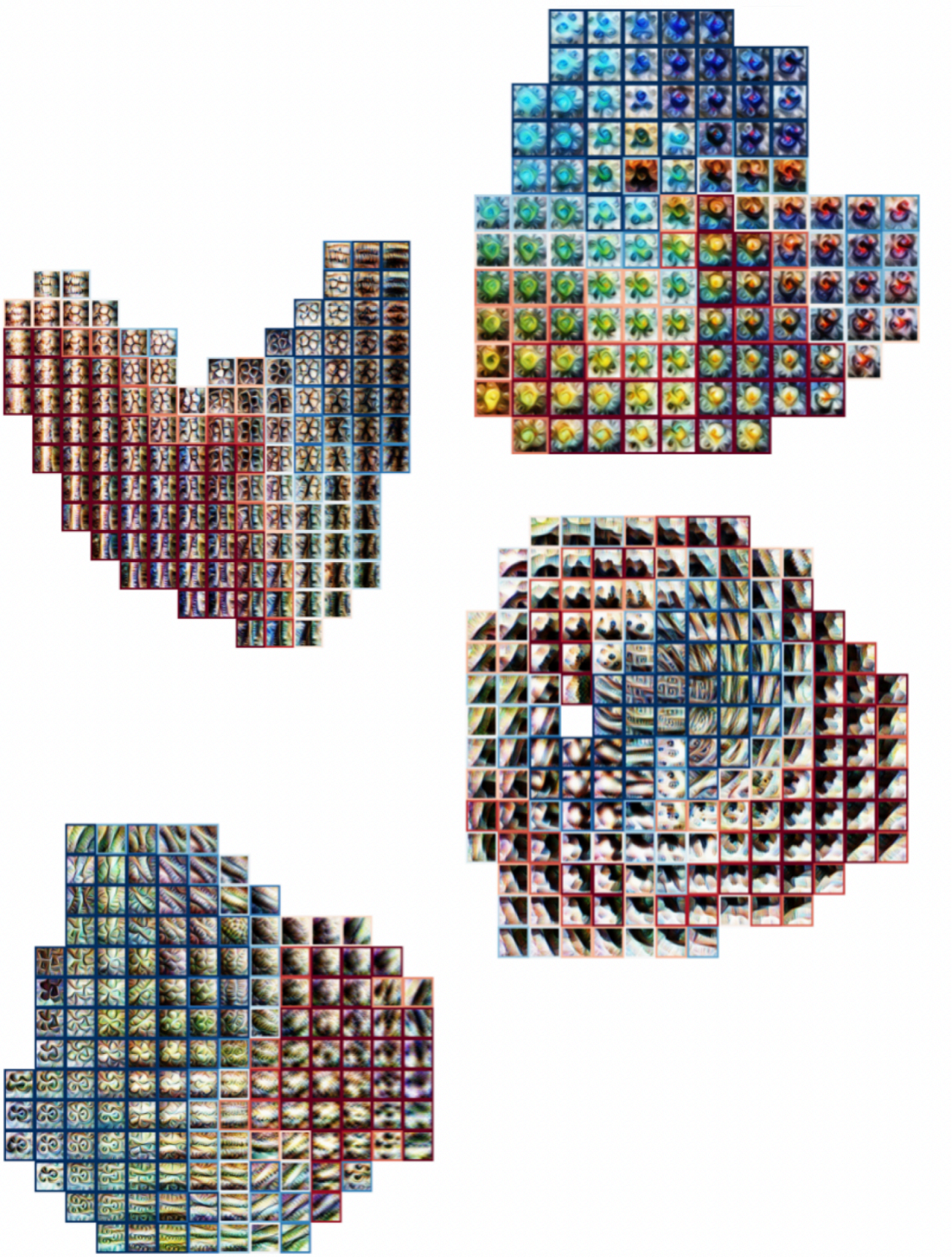

**Figure 18**

**Figure 19**