AlignedCut: Visual Concepts Discovery on Brain-Guided Universal Feature Space

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the intriguing connection between visual data, deep networks, and the brain. Our method creates a universal channel alignment by using brain voxel fMRI response prediction as the training objective. We discover that deep networks, trained with different objectives, share common feature channels across various models. These channels can be clustered into recurring sets, corresponding to distinct brain regions, indicating the formation of visual concepts. Tracing the clusters of channel responses onto the images, we see semantically meaningful object segments emerge, even without any supervised decoder. Furthermore, the universal feature alignment and the clustering of channels produce a picture and quantification of how visual information is processed through the different network layers, which produces precise comparisons between the networks.

1 Introduction

Introducing a novel approach, Yang et al. (2024) has successfully established a method of computing a mapping between the brain and deep-nets, effectively linking two black boxes. The brain fMRI prediction task allows for visualizing information flow from layer to layer, using the brain as an analysis tool.

If a picture is worth a thousand words, the main idea is that the brain's thousands of voxels can be thought of as alphabets for these words that describe an image. Just as alphabets must be combined to form words and phrases with meanings, we need to find the grouping of brain voxels and their network channel counterparts to understand their meaning (Figure 1).

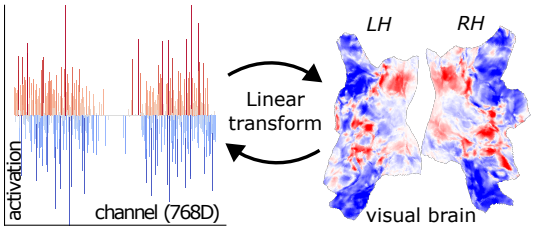

Figure 1: Transform the hidden channel activation of deep-nets into visual brain voxels' response.

Our main discovery is that while the network layer structure differs, channel feature correspondence exists across networks with a shared encoding of reoccurring visual concepts. This paper builds upon the idea of 'Rosetta stone' neurons (Dravid et al., 2023), which find channels across networks that share similar image responses in binary segmentation. If channels are alphabets, 'Rosetta stone' provides an alphabet-level translation between networks.

Individual channel-level analysis could miss feature correspondence across networks at finer and coarser levels. On a finer level, because the channels are invariant up to a linear transformation, we might miss a reconstituted feature constructed from a composition of existing channels. On a coarse level, the channels can be combined and clustered to form a bigger 'Rosetta' concept.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

To address fine-level channel analysis, we use brain voxel response as a reference signal and linearly transform channels for each network into a shared space sufficient for brain fMRI prediction. This process produces a universal feature space that aligns channel features across the layers and models.

To find bigger visual concepts, one can start with Neuroscience knowledge of brain regions (ROIs) with specific brain functionality, i.e., V1, V4, and EBA. While tracing the mapping of the ROIs to channels can produce visual concepts (Figure 2), brain regions don't function in isolation.

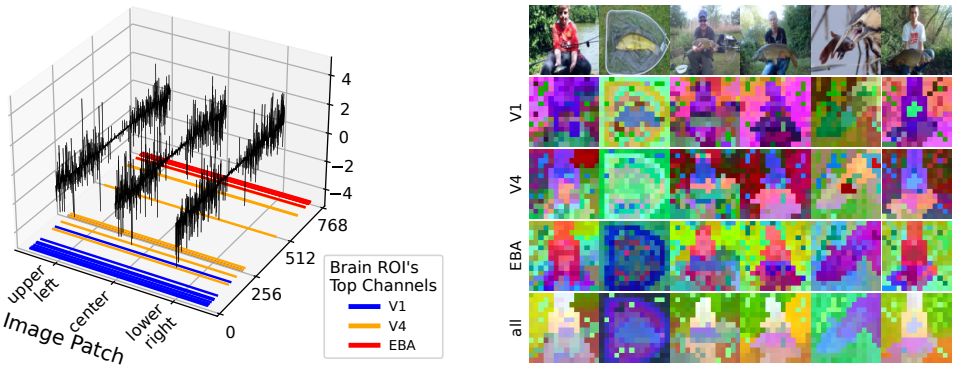

Figure 2: From the 768D feature on CLIP layer-6, we extract different levels of segmentation by restricting the use of a subset of channels. ***Left:*** Channel activation on example image patches. The ordering of channels is sorted from the early brain to the late brain by their weights for brain voxels. ***Right:*** Spectral clustering on each subset of channels filtered by each brain ROI (V1, V4, EBA), image pixels colored by 3D spectral-tSNE of top 10 eigenvectors.

Instead of searching through all possible channel grouping combinations, our first insight is that we can create a channel grouping hypothesis by examining channels from each pixel's perspective. Think of the pixels and channels forming a bipartite graph; each channel produces a per-pixel response (image activation map), defining the graph edge between the pixels and channels. Taking the perspective of pixels, one can collect graph edges incident on each pixel into a vector, which can be thresholded to produce a hypothesis grouping over channels.

Our second insight is that if a channel grouping hypothesis repeats across images, layers, and models, it is highly unlikely to be accidental and, therefore, signals meaningful visual concepts.

We formulate this clustering problem as a graph partition task. The graph nodes are the product space of pixels and layers. We apply spectral clustering to produce k-top eigenvectors. We take advantage of two properties of spectral clustering: it makes 1) soft-cluster embedding space in the form of eigenvectors and 2) hierarchical clustering by varying the number of eigenvectors.

We made the following discoveries. First, shared channel sets, reoccurring across layers and models, predict response in distinct brain regions. By tracing the channel activation to the known brain ROI properties, we observe that the channel cluster encodes visual concepts at various levels of visual abstraction.

Second, meaningful object segments can emerge by tracing the channel cluster responses onto each image. We observed that some channel clusters produce figure/ground separation while others produce fine-grained category classification. Our image segmentation requires no additional segmentation decoder and uses only a simple distance measure over the eigenvectors.

Finally, the universal feature alignment and the spectral clustering of channels produce a picture and quantification of how visual information is processed through the different network layers.

While these discoveries are promising, there are two main technical hurdles to overcome to verify them on a large scale. Our method rests upon a crucial assumption: the channels across the different layers and models can be mapped into a shared space. While brain prediction over thousands of voxels can provide strong guidance for this alignment, an additional constraint would be needed when the shared space has a large dimension (suitable for expressiveness). We use clustering as a constraint, ensuring alignment linear transformation preserves spectral clustering eigenvectors. Furthermore, the graph size is enormous as it is a product space over pixels, layers, images, and models; therefore, computing eigenvectors over their pairwise affinity matrix can be computationally infeasible. We developed a Nystrom-like approximation to ensure efficient computation.

In summary, our key contributions are:

1. We constructed a universal channel-aligned space using brain encoding as supervision and spectral clustering eigenvector constraints to ensure minimal channel signal loss. Brain encoding associates the aligned channel space to brain regions and gives them meanings.

2. Models trained with different objectives learned similar visual concepts: corresponding channel patterns exist across different models. The resulting visual concepts can be validated by unsupervised segmentation benchmarks on ImageNet-segmentation and PASCAL VOC.

3. Models show divergent computation paths over the visual concept space formed by the top-k spectral eigenvectors. Different models differ in trajectories and pace of movement layer-to-layer.

2 Methods: AlignedCut

Just as human languages might consist of distinct alphabets, features across different models appear superficially in embedding spaces as almost mutually orthogonal (Figure 3). However, the underlying information that they represent can be similar. To jointly analyze features across models and layers, we proposed the **channel align transform** that linearly projects features to a universal space.

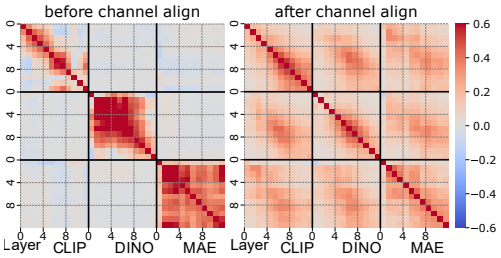

Figure 3: Cosine similarity of channel activation on the same image inputs.

The learning signal for the channel align transform is provided by **brain response prediction**. Learning from brain prediction offers two advantages. First, brain response covers rich representations from all levels of semantics; the channel alignment removes irrelevant information while preserving the necessary and sufficient visual image features. Second, knowledge of brain regions provides an interpretable understanding of their corresponding channels derived from the alignment.

Our visual concept discovery is formulated as a graph partitioning task using **spectral clustering**. We term our approach for this channel align and graph partitioning as **AlignedCut**. Furthermore, a major challenge in applying spectral clustering to large graphs is the complexity scaling issue. To address this, we developed a **Nystrom-like approximation** to reduce the computational complexity.

2.1 Brain-Guided Universal Channel Align

Brain Dataset We used the Algonauts competition (Gifford et al., 2023) release of Nature Scenes Dataset (NSD) (Allen et al., 2022). Briefly, NSD provides an fMRI brain scan when watching COCO images. Each subject viewed 10,000 images over 40 hours of scanning. We used the first subject's publicly shared pre-processed and denoised (Prince et al., 2022) data.

Channel Align Let $\mathcal{V} = \{\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_n | \boldsymbol{V}_i \in \mathbb{R}^{P \times D_i}\}$ be the set of image features, extracted from each layer of pre-trained ViT models, where $P = (H \times W + 1)$ is image patches and class token, D_i is the hidden dimension. In particular, we used the attention layer output for each $\boldsymbol{V_i}$ without adding residual connections from previous layers. Let $\mathcal{V}'$ be the channel-aligned features; the goal of channel alignment is to learn a set of linear transform $\mathcal{W} = \{\boldsymbol{W}_1, \boldsymbol{W}_2, \cdots, \boldsymbol{W}_n | \boldsymbol{W}_i \in \mathbb{R}^{D_i \times D'}\}$. In the new D' dimensional space, channels are aligned.

$$\mathcal{V}' = \mathcal{V} \odot \mathcal{W} = \{\boldsymbol{V}_1\boldsymbol{W}_1, \boldsymbol{V}_2\boldsymbol{W}_2, \cdots, \boldsymbol{V}_n\boldsymbol{W}_n | \boldsymbol{V}_i\boldsymbol{W}_i \in \mathbb{R}^{P \times D'}\} \tag{1}$$

Brain Prediction To produce a learning signal for channel align $\mathcal{W}$, features from $\mathcal{V}'$ are summed (*not concatenated*) to do brain prediction. Let $\boldsymbol{Y} \in \mathbb{R}^{1 \times N}$ be the brain prediction target, where N is the number of flattened 3D brain voxels, and 1 indicates that each voxel's response is a scalar value. Let $F_\theta : \mathbb{R}^{P \times D'} \Rightarrow \mathbb{R}^{1 \times N}$ be the learned brain encoding model; without loss of generalizability, we set F_θ as global average pooling then linear weight $\boldsymbol{\beta}_\theta \in \mathbb{R}^{D' \times N}$ and bias $\boldsymbol{\epsilon}_\theta \in \mathbb{R}^{1 \times N}$:

$$\left[\underset{p \in P}{\texttt{Avg Pool}}\left(\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{V}_i\boldsymbol{W}_i)\right) \times \boldsymbol{\beta}_\theta + \boldsymbol{\epsilon}_\theta\right] \Rightarrow \boldsymbol{Y} \tag{2}$$

Channel in the Brain's Space Let $\mathcal{B} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_n | \boldsymbol{B}_i \in \mathbb{R}^{P \times N}\}$ be the set of channel activations in the brain's space. By defining $\boldsymbol{B}_i := \boldsymbol{V}_i \boldsymbol{W}_i \times \boldsymbol{\beta}_\theta$, we have the brain response prediction $\boldsymbol{Y} = \texttt{Avg Pool}_{p \in P}(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{B}_i) + \boldsymbol{\epsilon}_\theta$ (Eq. (2)). Intuitively, we linearly transformed the activation to the brain's space, such that the activation from all slots sum up to the brain response prediction.

2.2 Graph Spectral Clustering

Spectral Clustering We use spectral clustering for visual concepts discovery and image-channel analysis; it provides 1) soft-cluster embedding space and 2) unsupervised hierarchical image segmentation. Normalized Cut (Shi and Malik, 2000) partitions the graph into sub-graphs with minimal cost of breaking edges. It embeds the graph into a lower dimensional eigenvector representation, where each eigenvector is a hierarchical sub-graph assignment.

Let $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ be the symmetric affinity matrix, where M denotes the total number of image patches. Given channel aligned features $\boldsymbol{V}' \in \mathbb{R}^{M \times D'}$, we define $\boldsymbol{A}_{ij} := \exp(\cos(\boldsymbol{V}'_i, \boldsymbol{V}'_j) - 1)$ such that $\boldsymbol{A}_{ij} > 0$ measures the similarity between data i and j. The spectral clustering embedding $\boldsymbol{X} \in \mathbb{R}^{M \times C}$ is solved by the top C eigenvectors of the following generalized eigenproblem:

$$(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2})\boldsymbol{X} = \boldsymbol{X}\boldsymbol{\Lambda} \tag{3}$$

where $\boldsymbol{D}$ is the diagonal degree matrix $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$, $\boldsymbol{\Lambda}$ is diagonal eigenvalue matrix.

Nystrom-like Approximation Computing eigenvectors for $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ is prohibitively expensive for enormous M with a time complexity of $O(M^3)$. The original Nystrom approximation method (Fowlkes et al., 2004) reduced the time complexity to $O(m^3 + m^2 M)$ by solving eigenvectors on sub-sampled graph $\boldsymbol{A}' \in \mathbb{R}^{m \times m}$, where $m \ll M$. In particular, the orthogonalization step of eigenvectors introduced the time complexity of $O(m^2 M)$. Because our Nystrom-like approximation trades the $O(m^2 M)$ orthogonalization term with the K-nearest neighbor, our Nystrom-like approximation reduced the time complexity to $O(m^3 + mM)$.

Our Nystrom-like Approximation first solves the eigenvector $\boldsymbol{X}' \in \mathbb{R}^{m \times C}$ on a sub-sampled graph $\boldsymbol{A}' \in \mathbb{R}^{m \times m}$ using Equation (3), then propagates the eigenvector from the sub-graph m nodes to the full-graph M nodes. Let $\tilde{\boldsymbol{X}} \in \mathbb{R}^{M \times C}$ be the approximation $\tilde{\boldsymbol{X}} \approx \boldsymbol{X}$. The eigenvector approximation $\tilde{\boldsymbol{X}}_i$ of full-graph node $i \leq M$ is assigned by averaging the top K-nearest neighbors' eigenvector $\boldsymbol{X}'_k$ from the sub-graph nodes $k \leq m$:

$$\begin{aligned} \mathcal{K}_i &= KNN(\boldsymbol{A}_{*i}; m, K) = \arg\max_{k \leq m} \sum_{k=1}^{K} \boldsymbol{A}_{ki} \\ \tilde{\boldsymbol{X}}_i &= \frac{1}{\sum_{k \in \mathcal{K}_i} \boldsymbol{A}_{ki}} \sum_{k \in \mathcal{K}_i} \boldsymbol{A}_{ki} \boldsymbol{X}'_k \end{aligned} \tag{4}$$

where $KNN(\boldsymbol{A}_{*i}; m, K)$ denotes KNN from full-graph node $i \leq M$ to sub-graph nodes $k \leq m$.

2.3 Affinity Eigen-constraints as Regularization for Channel Align

While brain prediction can provide strong supervision for the learned channel align operation, we observed that the quality of unsupervised segmentation dropped after the channel alignment. To address this issue, a regularization term is added:

$$\mathcal{L}_{eigen} = \|\boldsymbol{X}_b \boldsymbol{X}_b^T - \boldsymbol{X}_a \boldsymbol{X}_a^T\| \tag{5}$$

where $\boldsymbol{X}_b$ and $\boldsymbol{X}_a \in \mathbb{R}^{\tilde{m} \times c}$ are affinity matrix eigenvectors before and after channel alignment, respectively; $\tilde{m} = 100$ are randomly sampled nodes in a mini-batch and $c = 6$ are the top eigenvectors. The eigen-constraint preserves spectral clustering eigenvectors in dot-product space, invariant to random rotations in eigenvectors. We found adding eigen-constraints improved both the performance of segmentation (Figure 5) and the brain prediction score (Table 1).

Table 1: Affinity eigen-constraints improved brain score (R^2: variance explained).

	ROI Brain Score R^2 ($\pm$ 0.001)			
λ_{eigen}	V1	V4	EBA	all
1.0	**0.170**	**0.181**	0.295	**0.196**
0.1	0.167	0.179	0.294	0.193
0	0.155	0.166	0.296	0.188

3 Results

Our spectral clustering analysis aims to discover visual concepts that share the same pattern of channel activation across different models and layers. However, implementing spectral clustering analysis comes with two main challenges. First, the models sit in different feature spaces, so direct clustering will not reveal their overlap and similarities. Second, when scaling up to a large graph, spectral clustering is computationally expensive.

To address the first challenge, we developed our channel align transform to align features into a universal space. We extracted features from all 12 layers of the CLIP (ViT-B, OpenAI) (Radford et al., 2021), DINOv2 (ViT-B with registers) (Darcet et al., 2024), and MAE (ViT-B) (He et al., 2022) and then transformed features from each layer into the universal feature space.

To address the second challenge, we developed our Nystrom-like approximation to reduce the computational complexity. We extracted features from 1000 ImageNet (Deng et al., 2009) images, with each image consisting of 197 patches per layer. The entire product space of all images and features totaled $M = 7\text{e}{+6}$ nodes, from which we applied our Nystrom-like approximation with subsampled $m = 5\text{e}{+4}$ nodes and KNN $K = 100$, computing the top 20 eigenvectors.

To visualize the affinity eigenvectors, the top 20 eigenvectors were reduced to a 3-dimensional space by t-SNE, and a color value was assigned to each node by the RGB cube. We call this approach AlignedCut color.

Figure 4: Spectral clustering in the universal channel aligned feature space. The image pixels are colored by our approach AlignedCut, the pixel RGB value is assigned by the 3D spectral-tSNE of the top 20 eigenvectors. The coloring is consistent across all images, layers, and models.

In Figure 4, we displayed the analysis, AlignedCut color, and made the following observations:

1. In CLIP layer-5, DINO layer-6, and MAE layer-8, there is class-agnostic figure-ground separation, with foreground objects from different categories grouped into the same AlignedCut color.

2. In CLIP layer-9, there is a class-specific separation of foreground objects, with foreground objects grouped into AlignedCut colors with associated semantic categories.

3. Before layer-3, CLIP and DINO produce the same AlignedCut color regardless of the image input. From layer-4 onwards, the AlignedCut color smoothly changes over layers.

3.1 Figure-ground representation emerge before categories

In this section, we benchmark each layer in CLIP with unsupervised segmentation. The key findings from this benchmarking are: **1)** The figure-ground representation emerges at CLIP layer-4 and is preserved in subsequent layers; **2)** Categories emerge over layers, peaking at layer-9 and layer-10.

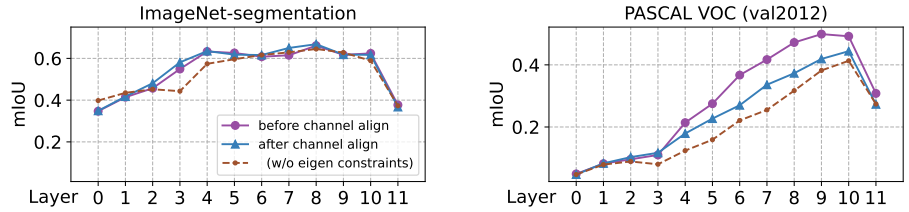

Figure 5: Unsupervised segmentation scores from spectral clustering on each CLIP layer. ImageNet-segmentation dataset is used with binary figure-ground labels, and the mIoU score peaks plateau from layer-4 to layer-10. In PASCAL VOC with 20 class labels, the mIoU score peaks at layer-9.

From which layers did the figure-ground and category representations emerge? We conducted experiments that compared the unsupervised segmentation scores across layers, tracing how well each representation is encoded at each layer. We used two datasets: a) ImageNet-segmentation (Guillaumin et al., 2014) with binary figure-ground labels, and b) PASCAL VOC (Everingham et al., 2010) with 20 category labels. The results are presented in Figure 5. On the ImageNet-segmentation benchmark, the score peaks at layer-4 (mIoU=0.6) and plateaus in subsequent layers, suggesting that the figure-ground representation is encoded and preserved from layer-4 onwards. On the PASCAL VOC benchmark, the score peaks at layer-9 and layer-10 (mIoU=0.5) even though it is low at layer-4 (mIoU=0.2), indicating that category information is encoded at layer-9 and layer-10. Overall, we conclude that the figure-ground representation emerges before the category representation.

3.2 Visual concepts: class-agnostic figure-ground

In this section, we use brain activation heatmaps and image similarity heatmaps to describe figure-ground visual concepts. The key findings from these heatmaps are: **1)** The figure vs. ground pixels activate different channels; **2)** The figure-ground visual concept is class-agnostic; **3)** The figure-ground visual concept is consistent across models.

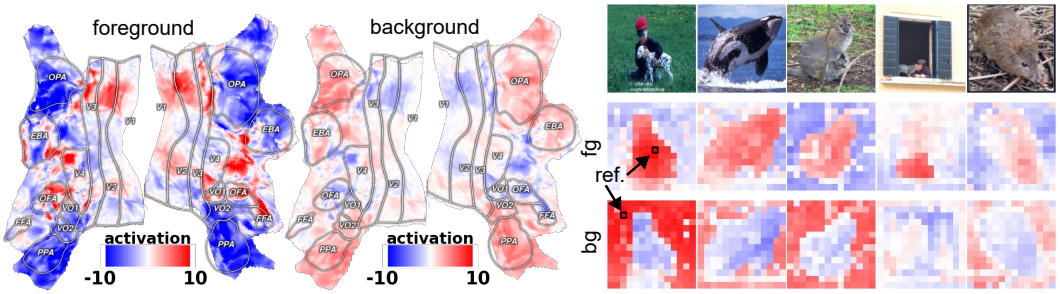

Figure 6: The figure-ground visual concepts in CLIP layer-5. ***Left:*** Mean activation of foreground or background pixels, linearly transformed to the brain's space. ***Right:*** Cosine similarity from ***one*** reference pixel marked. The figure-ground visual concepts are agnostic to image categories.

How can the channel activation patterns of the figure-ground visual concept be described? We averaged the channel activations from foreground and background pixels, using the ground-truth labels from the ImageNet-segmentation dataset. The averaged channel activations were transformed into the brain's space. In Figure 6, foreground pixels exhibit positive activation in early visual brain ROIs (V1 to V4) and the face-selective ROI (FFA), while negatively activating place-selective ROIs (OPA and PPA). Interestingly, background pixels activate the reverse pattern compared to foreground pixels. Overall, the figure and ground pixels activate distinct brain ROIs.

Is the figure-ground visual concept class-agnostic? We manually selected *one* pixel and computed the cosine similarity to all of the other image pixels. In Figure 6, the results demonstrate that one pixel (on the human) could segment out foreground objects from all other classes (shark, dog, cat, rabbit). The same result holds true for one background pixel. We conclude that the figure-ground visual concept is class-agnostic.

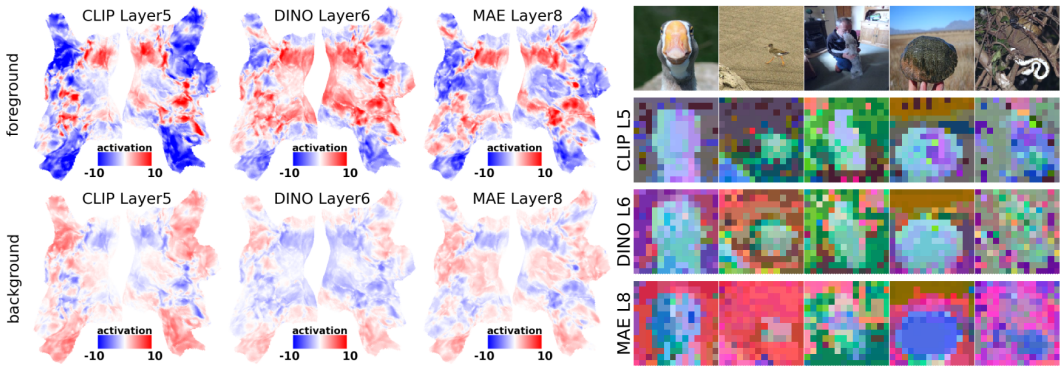

Figure 7: The same figure-ground visual concepts are found in CLIP, DINO and MAE. ***Left:*** Mean activation of all foreground (top) and background (bottom) pixels; the three models exhibit similar activation patterns. ***Right:*** AlignedCut, pixels colored by 3D spectral-tSNE of the top 20 eigenvectors; the three models show similar grouping colors for foreground pixels.

Is the figure-ground visual concept consistent across models? We performed the channel analysis for CLIP, DINO, and MAE. In Figure 7, the foreground or background pixels activates similar brain ROIs across the three models. Additionally, spectral clustering grouped the representations of foreground objects into similar colors for CLIP and DINO (light blue), the grouping for MAE is less similar (dark blue). Overall, the figure-ground visual concept is consistent across models.

3.3 Visual concepts: categories

In this section we use AlignedCut to discover category visual concepts. The key findings from the category visual concepts are: **1)** Class-specific visual concepts activate diverse brain regions; **2)** Visual concepts with higher channel activation values are more consistent.

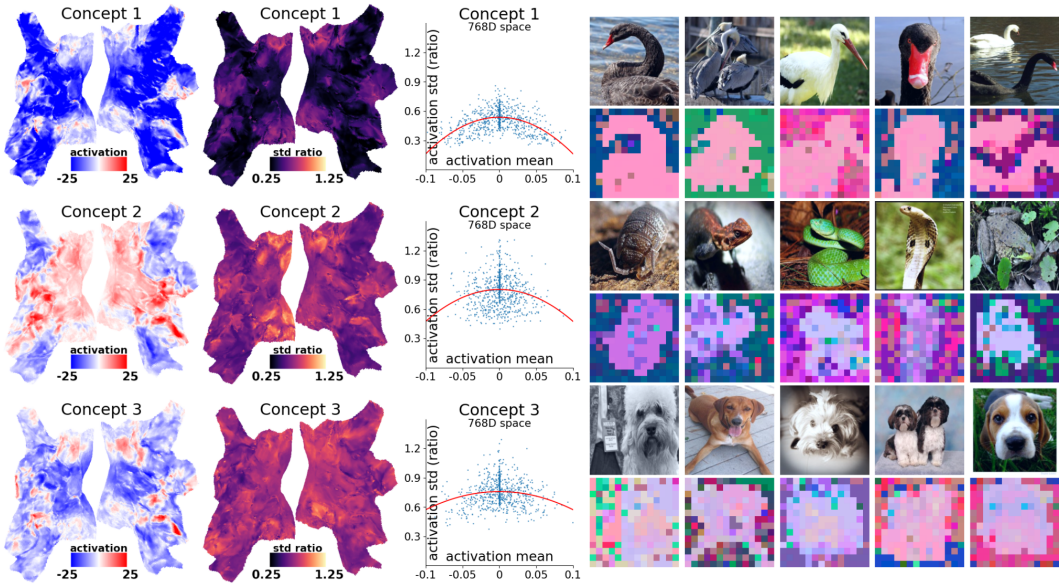

Figure 8: Category visual concepts in CLIP layer-9. ***Left:*** Mean activation of all pixels within an Euclidean sphere centered at the visual concept in the 3D spectral-tSNE space; the concepts activate different brain regions. ***Middle:*** The standard deviation negatively correlates with absolute mean activations. ***Right:*** AlignedCut, pixels colored by 3D spectral-tSNE of the top 20 eigenvectors.

How does each class-specific concept activate the channels? To answer this question, we sampled class-specific concepts from CLIP layer-9. First, we used farthest point sampling to identify candidate centers in the 3D spectral-tSNE space. Then, each candidate center was grouped with its neighboring pixels within an Euclidean sphere in the spectral-tSNE space. Finally, the channel activations of the grouped pixels were averaged to produce the mean channel activation for each visual concept. In Figure 8, Concept 1 (duck, goose) negatively activates late brain regions; Concept 2

(snake, turtle) positively activates early brain regions and also FFA; Concept 3 (dog) negatively activates early brain regions. Overall, category-specific visual concepts activate diverse brain regions.

How do we quantify the consistency of each visual concept? Qualitatively, Concept 1 exhibits more consistent coloring (Figure 8, pink) than Concept 3 (purple). To further quantify this observation, we computed the mean and standard deviation of channel activations for each Euclidean sphere centered on a concept. In Figure 8, there is a reverse U-shape relation between magnitude and standard deviation. The reverse U-shape implies that larger absolute mean channel activation corresponds to lower standard deviation. Overall, higher channel activation magnitudes suggest more consistent visual concepts.

3.4 Transition of visual concepts over layers

In this section, instead of using 3D spectral-tSNE, we use 2D spectral-tSNE to trace the layer-to-layer feature computation. The key findings of spectral-tSNE in 2D are: **1)** The figure vs. ground pixels are encoded in separate spaces in late layers; **2)** The representations for foreground and background bifurcate at CLIP layer-4 and DINO layer-5.

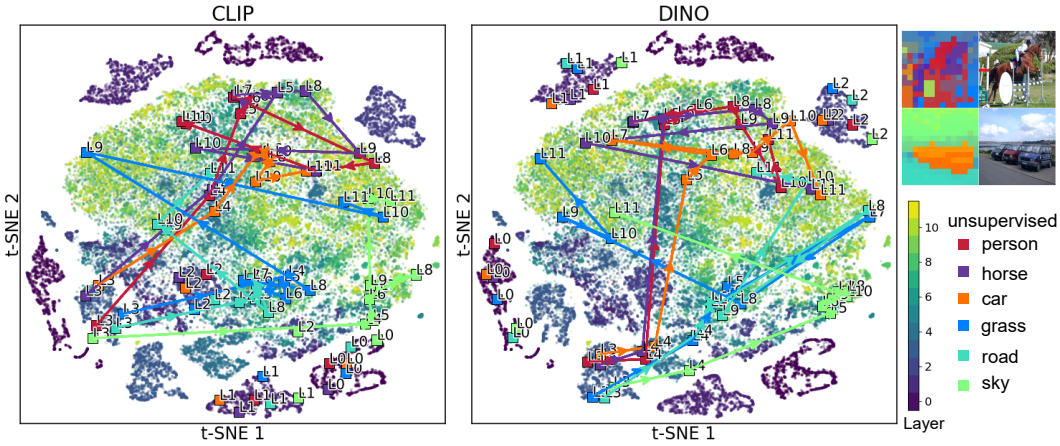

Figure 9: Trajectory of feature progression in layers for six example pixels. ***Left:*** 2D spectral-tSNE plot of the top 20 eigenvectors, jointly clustered across all models; the foreground and background pixels bifurcate at CLIP layer-4 and DINO layer-5. ***Right:*** Pixels colored by unsupervised segmentation.

How does the network encode figure and ground pixels in each layer? We performed spectral clustering and 2D t-SNE on the top 20 eigenvectors to project all layers into a 2D spectral-tSNE space. In Figure 9, we found that all foreground and background pixels are grouped together in each early layer. Each early layer (dark dots) forms an isolated cluster separate from other layers, while late layers (bright dots) are grouped in the center. In the late layers, there is a separation where foreground pixels occupy the upper part of 2D spectral-tSNE space, while background pixels occupy the middle part. Overall, foreground and background pixels are encoded in separate spaces in late layers.

How does the network process each pixel from layer to layer? In the 2D spectral-tSNE plot, we traced the trajectory for each pixel from layer-3 to the last layer. In Figure 9, we found that the trajectories for foreground and background pixels bifurcate: foreground pixels (person, horse, car) traverse to the upper side and remain within the upper side; background pixels (grass, road, sky) jump between the middle right and left sides. The same bifurcation is consistently observed for CLIP from layer-3 to layer-4 and DINO from layer-4 to layer-5. Furthermore, to quantify the bifurcation for foreground and background pixels, we first sampled 5 visual concepts from CLIP layer-3 and layer-4. Then, we measured the transition probability between visual concepts, defined as the proportion of pixels that transited from an Euclidean circle around concept A to a circle around concept B. In Figure 10, the transition probability of foreground pixels to the upper side (A1 to B0) is higher than that of background pixels (0.44 vs. 0.16), while the transition probability of background pixels to the right side (A4 to B4) is higher than that of foreground pixels (0.36 vs. 0.06). Overall, this suggests a bifurcation of figure and ground pixel representations at the middle layers of both CLIP and DINO.

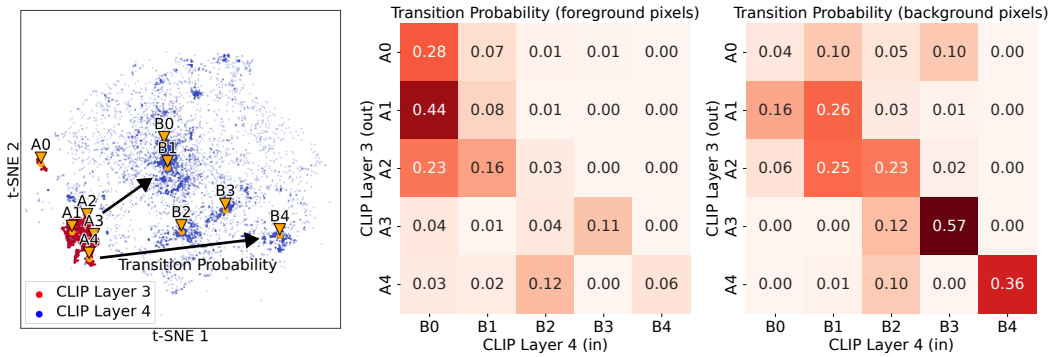

Figure 10: Transition probability of visual concepts from CLIP layer-3 to layer-4. ***Left:*** Five visual concepts sampled from CLIP layer-3 and layer-4. ***Right:*** Transition probability measured separately for foreground and background pixels; a bifurcation occurs where foreground pixels have more traffic to concept B0, while background pixels have more traffic to concepts B3 and B4.

4 Related Work

Mechanistic Interpretability is a field of study that intends to understand and explain the inner working mechanisms of deep networks. One approach is to interpret individual neurons (Bau et al., 2017; Dravid et al., 2023) and circuit connections between neurons (Olah et al., 2020). Another approach is to interpret transformer attention heads (Gandelsman et al., 2024) and circuit connections between attention heads (Wang et al., 2023a). Other approaches also looked into the role of patch tokens (Sun et al., 2024). These approaches made the assumption that channels are aligned within the same model; we compare across models by actively aligning the channels to a universal space.

Spectral Clustering is a graphical method to analyze data grouping in the eigenvector space. Spectral methods have been widely used for unsupervised image segmentation (Shi and Malik, 2000; von Luxburg, 2007; Wu et al., 2018; Wang et al., 2023b). One major challenge for applying spectral clustering to large graphs is the complexity scaling issue. To solve the scaling issue, the Nystrom approximation (Fowlkes et al., 2004) approaches solve eigenvectors on sub-sampled graphs and then propagate to the full graph. Another approach is the gradient-based eigenvector solver (Zhang et al., 2023), which solves the eigenvectors in mini-batches. Our proposed Nystrom-like approximation achieves a computational speedup over the original Nystrom approximation, albeit at the expense of weakened orthogonality of the eigenvectors.

Brain Encoding Model is widely used by the computational neuroscience community (Kriegeskorte and Douglas, 2018). They have been using deep nets to explain the brain's function. One approach is to use the gradient of the brain encoding model to find the most salient image features (Sarch et al., 2023). Another approach generate text caption for brain activation (Luo et al., 2024). Other approaches compare brain prediction performance for different models (Schrimpf et al., 2020). The field focused on using deep nets as a tool to explain the brain's function; we go in the opposite direction by using the brain to explain deep nets.

5 Conclusion and Limitations

We present a novel approach to interpreting deep neural networks by leveraging brain data. Our fundamental innovation is twofold: First, we use brain prediction as guidance to align channels from different models into a universal feature space; Second, we developed a Nystrom-like approximation to scale up the spectral clustering analysis. Our key discovery is that recurring visual concepts exist across networks and layers; such concepts correspond to different levels of objects, ranging from figure-ground to categories. Additionally, we quantified the information flow from layer to layer, where we found a bifurcation of figure-ground visual concepts.

Limitations. While the learned channel align transformation projects all features onto a universal feature space, the nature of learned transformation does not preserve all the information. There is a small drop in unsupervised segmentation performance after channel alignment, which is not fully addressed by our proposed eigen-constraint regularization. Secondly, as a trade-off for faster computation, our Nystrom-like approximation does not produce strictly orthogonal eigenvectors. To produce expressive eigenvectors, our approximation relies on using larger sub-sample sizes than the original Nystrom method.

References

Allen, E. J., St-Yves, G., Wu, Y., Breedlove, J. L., Prince, J. S., Dowdle, L. T., Nau, M., Caron, B., Pestilli, F., Charest, I., Hutchinson, J. B., Naselaris, T., and Kay, K. (2022). A massive 7T fMRI dataset to bridge cognitive neuroscience and artificial intelligence. *Nature Neuroscience*, 25(1):116–126. Number: 1 Publisher: Nature Publishing Group. 3

Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *Computer Vision and Pattern Recognition*. 9

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. (2024). Vision Transformers Need Registers. In *The Twelfth International Conference on Learning Representations*. 5

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. 5

Dravid, A., Gandelsman, Y., Efros, A. A., and Shocher, A. (2023). Rosetta Neurons: Mining the Common Units in a Model Zoo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1934–1943. 1, 9

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338. 6

Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the Nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225. 4, 9

Gandelsman, Y., Efros, A. A., and Steinhardt, J. (2024). Interpreting CLIP's Image Representation via Text-Based Decomposition. In *The Twelfth International Conference on Learning Representations*. 9

Gifford, A. T., Lahner, B., Saba-Sadiya, S., Vilas, M. G., Lascelles, A., Oliva, A., Kay, K., Roig, G., and Cichy, R. M. (2023). The Algonauts Project 2023 Challenge: How the Human Brain Makes Sense of Natural Scenes. arXiv:2301.03198 [cs, q-bio]. 3

Guillaumin, M., Küttel, D., and Ferrari, V. (2014). ImageNet Auto-Annotation with Segmentation Propagation. *International Journal of Computer Vision*, 110(3):328–348. 6

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009. 5

Kriegeskorte, N. and Douglas, P. K. (2018). Cognitive computational neuroscience. *Nature Neuroscience*, 21(9):1148–1160. Publisher: Nature Publishing Group. 9

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2017). Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 14

Luo, A., Henderson, M. M., Tarr, M. J., and Wehbe, L. (2024). BrainSCUBA: Fine-Grained Natural Language Captions of Visual Cortex Selectivity. In *The Twelfth International Conference on Learning Representations*. 9

Olah, C., Cammarata, N., Schubert, L., Goh, G., Petrov, M., and Carter, S. (2020). Zoom In: An Introduction to Circuits. *Distill*, 5(3):e00024.001. 9

Prince, J. S., Charest, I., Kurzawski, J. W., Pyles, J. A., Tarr, M. J., and Kay, K. N. (2022). Improving the accuracy of single-trial fMRI response estimates using GLMsingle. *eLife*, 11:e77599. Publisher: eLife Sciences Publications, Ltd. 3

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning Transferable Visual Models From Natural Language Supervision. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR. 5

Sarch, G. H., Tarr, M. J., Fragkiadaki, K., and Wehbe, L. (2023). Brain Dissection: fMRI-trained Networks Reveal Spatial Selectivity in the Processing of Natural Images. In *Thirty-seventh Conference on Neural Information Processing Systems*. 9

Schrimpf, M., Kubilius, J., Lee, M. J., Murty, N. A. R., Ajemian, R., and DiCarlo, J. J. (2020). Integrative Benchmarking to Advance Neurally Mechanistic Models of Human Intelligence. *Neuron*. 9

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. 4, 9

Sun, M., Chen, X., Kolter, J. Z., and Liu, Z. (2024). Massive Activations in Large Language Models. arXiv:2402.17762 [cs]. 9

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416. 9

Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. (2023a). Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. In *The Eleventh International Conference on Learning Representations*. 9

Wang, X., Girdhar, R., Yu, S. X., and Misra, I. (2023b). Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134. 9

Wu, Z., Xiong, Y., Stella, X. Y., and Lin, D. (2018). Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9

Yang, H., Gee, J., and Shi, J. (2024). Brain Decodes Deep Nets. arXiv:2312.01280 [cs]. 1

Zhang, X., Yunis, D., and Maire, M. (2023). Deciphering 'What' and 'Where' Visual Pathways from Spectral Clustering of Layer-Distributed Neural Representations. arXiv:2312.06716 [cs]. 9

A Appendix overview

1. Appendix B summarizes background of brain ROIs.
2. Appendix C is implementation details
    - 2.1. Additional regularization terms
    - 2.2. Brain encoding model training loss function
    - 2.3. Unsupervised segmentation evaluation pipeline
    - 2.4. Nystrom-like approximation for t-SNE
3. Appendix D lists more image examples from the 3D spectral-tSNE.
4. Appendix E lists figure-ground channel activation for every model and layer.
5. Appendix F lists more example category-specific visual concepts.
6. Appendix G lists more example pixels from the 2D spectral-tSNE information flow.

B Brain Region Background Knowledge

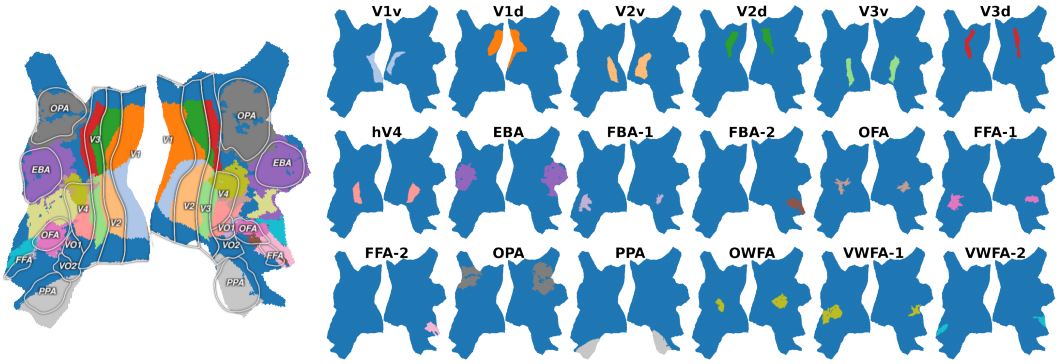

Figure 11: **Brain Region of Interests (ROIs)**. V1v: ventral stream, V1d: dorsal stream.

Table 2: Known function and selectivity of brain region of interests (ROIs).

ROI name	V1 V2 V3	V4	EBA FBA	OFA FFA	OPA	PPA	OWFA VWFA
Known Function/Selectivity	primary visual	mid-level	body	face	navigation	scene	words

This section briefly summarizes the known functions of key brain regions of interest (ROIs). Figure 11 provides an overview of these brain ROIs. Table 2 lists the known functions and selectivities for each ROI.

In brief, V1 to V3 are the primary visual stream, which is further divided into ventral (lower) and dorsal (upper) streams. V4 is a mid-level visual area. EBA (extrastriate body area) and FBA (fusiform body area) are selectively responsive to bodies, while FFA (fusiform face area) and OFA (occipital face area) show selectivity for faces. OWFA (occipital word form area) and VWFA (visual word form area) are selective for written words. PPA (parahippocampal place area) exhibits selectivity for scenes and places, and OPA (occipital place area) is involved in navigation and spatial reasoning.

Visual information processing in the brain follows a hierarchical, feedforward organization. Beginning in the primary visual cortex (V1) and progressing through higher visual areas like V2, V3, and V4, neurons exhibit increasingly large receptive fields and represent increasingly abstract visual concepts. While neurons in V1 encode low-level features like edges and orientations within a small portion of the visual field, neurons in V4 synthesize more complex patterns and object representations across a larger area of the visual input.

C Implementation Details

C.1 Additional Regularization for Channel Align Transformation

Additional Regularization are added to the channel align transform to ensure good properties of the aligned features: 1) zero-centered, 2) small covariance between channels, and 3) focal loss.

Zero-centered regularization. We did not apply z-score normalization to the extracted features; instead, we added a regularization term to ensure the transformed features are zero-centered. Recall that the channel-aligned transformed feature $\boldsymbol{V}' \in \mathbb{R}^{M \times D'}$, where M is the number of data points and D' is the hidden dimension. The zero-center loss is defined as:

$$\mathcal{L}_{\text{zero}} = \frac{1}{D'} \frac{1}{M} \sum_{i \leq M, j \leq D'} v'_{ij} \tag{6}$$

Covariance regularization. We used the covariance loss to minimize the off-diagonal elements in the covariance matrix of the transformed feature $C(\boldsymbol{V}')$, aiming to bring them close to $\mathbf{0}$. Recall that channel align transformed feature $\boldsymbol{V}' \in \mathbb{R}^{M \times D'}$, where M is number of data, D' is the hidden dimension. The covariance loss is defined as:

$$\mathcal{L}_{\text{cov}} = \frac{1}{D'} \sum_{i \neq j} [C(\boldsymbol{V}')]_{i,j}^2, \text{ where } C(\boldsymbol{V}') = \frac{1}{M-1} \sum_{i=1}^{M} \left(v'_i - \bar{v'}\right) \left(v'_i - \bar{v'}\right)^T, \bar{v'} = \frac{1}{M} \sum_{i=1}^{M} v'_i. \tag{7}$$

Focal Loss. Lin et al. (2017) introduced focal loss, which dynamically assigns smaller weights to the loss function for hard-to-classify classes. In our scenario, we apply spectral clustering on the affinity matrix $\boldsymbol{A}_a \in \mathbb{R}^{M \times M}$ after performing the channel alignment transform, where M represents the number of data points. Due to the characteristics of spectral clustering, disconnected edges play a more critical role than connected edges. Adding an edge between disconnected clusters significantly reshapes the eigenvectors, while adding edges to connected clusters has only a minor impact. Therefore, we aim to assign larger weights to disconnected edges in the loss function:

$$\mathcal{L}_{eigen} = \|(\boldsymbol{X}_b \boldsymbol{X}_b^T - \boldsymbol{X}_a \boldsymbol{X}_a^T) * \exp(-\boldsymbol{A}_b)\| \tag{8}$$

where $\boldsymbol{A}_b \in \mathbb{R}^{M \times M}$ is the affinity matrix before the channel alignment transform, element wise dot-product to $\exp(-\boldsymbol{A}_b)$ assigned larger wights for disconnected edges. $\boldsymbol{X}_b \in \mathbb{R}^{M \times C}, \boldsymbol{X}_a \in \mathbb{R}^{M \times C}$ are eigenvectors before and after channel align transform, respectively.

C.2 Brain Encoding Model Training Loss

Let $\boldsymbol{Y} \in \mathbb{R}^{1 \times N}$ represent the brain prediction target, where N is the number of flattened 3D brain voxels, and the 1 indicates that each voxel's response is a scalar value. $\hat{\boldsymbol{Y}}$ is the model's predicted brain response. The brain encoding model training loss is the L1 loss:

$$\mathcal{L}_{brain} = \|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\| \tag{9}$$

C.3 Total Training Loss

The total training loss is a combination of the following components: 1) brain encoding model loss, 2) eigen-constraint regularization, 3) zero-centered regularization, and 4) covariance regularization:

$$\mathcal{L} = \mathcal{L}_{brain} + \lambda_{eigen}\mathcal{L}_{eigen} + \lambda_{zero}\mathcal{L}_{zero} + \lambda_{cov}\mathcal{L}_{cov} \tag{10}$$

where we set $\lambda_{eigen} = 1$, $\lambda_{zero} = 0.01$, $\lambda_{cov} = 0.01$.

C.4 Oracle-based Unsupervised Segmentation Evaluation Pipeline

Our unsupervised segmentation pipeline aims to benchmark and compare the performance across each single layer of the CLIP model. The evaluation pipeline is oracle-based:

1. Apply spectral clustering jointly across all images, taking the top 10 eigenvectors.

2. For each class of object (plus one background class), use ground-truth labels from the dataset to mask out the pixels and their eigenvectors, and then use the mean of the eigenvectors to define a center for each class.

3. Compute the cosine similarity of each pixel to all class centers.

4. For each pixel, if the maximum similarity to all classes is less than a threshold value, assign this pixel to the background class.

5. Assign pixels (with a similarity greater than the threshold value) to the class with the maximum similarity.

There's one hyper-parameter, the threshold value that requires different optimal value for each layer of CLIP. To ensure a fair comparison across all layers, the threshold value is grid-searched from 10 evenly spaced values between 0 and 1, the maximum mIoU score in the grid search is taken for each layer.

C.5 Nystrom-like approximation for t-SNE

To visualize the eigenvectors, we applied t-SNE to the eigenvectors $\boldsymbol{X} \in \mathbb{R}^{M \times C}$, where the number of data points M span the product space of models, layers, pixels, and images. Due to the enormous size of $M = 7\text{e}{+6}$ nodes, t-SNE suffered from complexity scaling issues. We again applied our Nystrom-like approximation to t-SNE, with sub-sampled $m = 10\text{e}{+4}$ nodes and KNN $K = 1$.

It's worth noting that, since the non-linear distance adjustment in t-SNE, it's crucial to use only one nearest neighbor $K = 1$ for t-SNE.

C.6 Computation Resource

All of our experiments are performed on one consumer-grade RTX 4090 GPU. The brain encoding model training took 3 hours on 4GB of VRAM, spectral clustering eigen-decomposition on large graph took 10 minutes on 10GB of VRAM and 60GB of CPU RAM.

C.7 Code Release

Our code will be publicly released upon publication.

D 3D spectral-tSNE

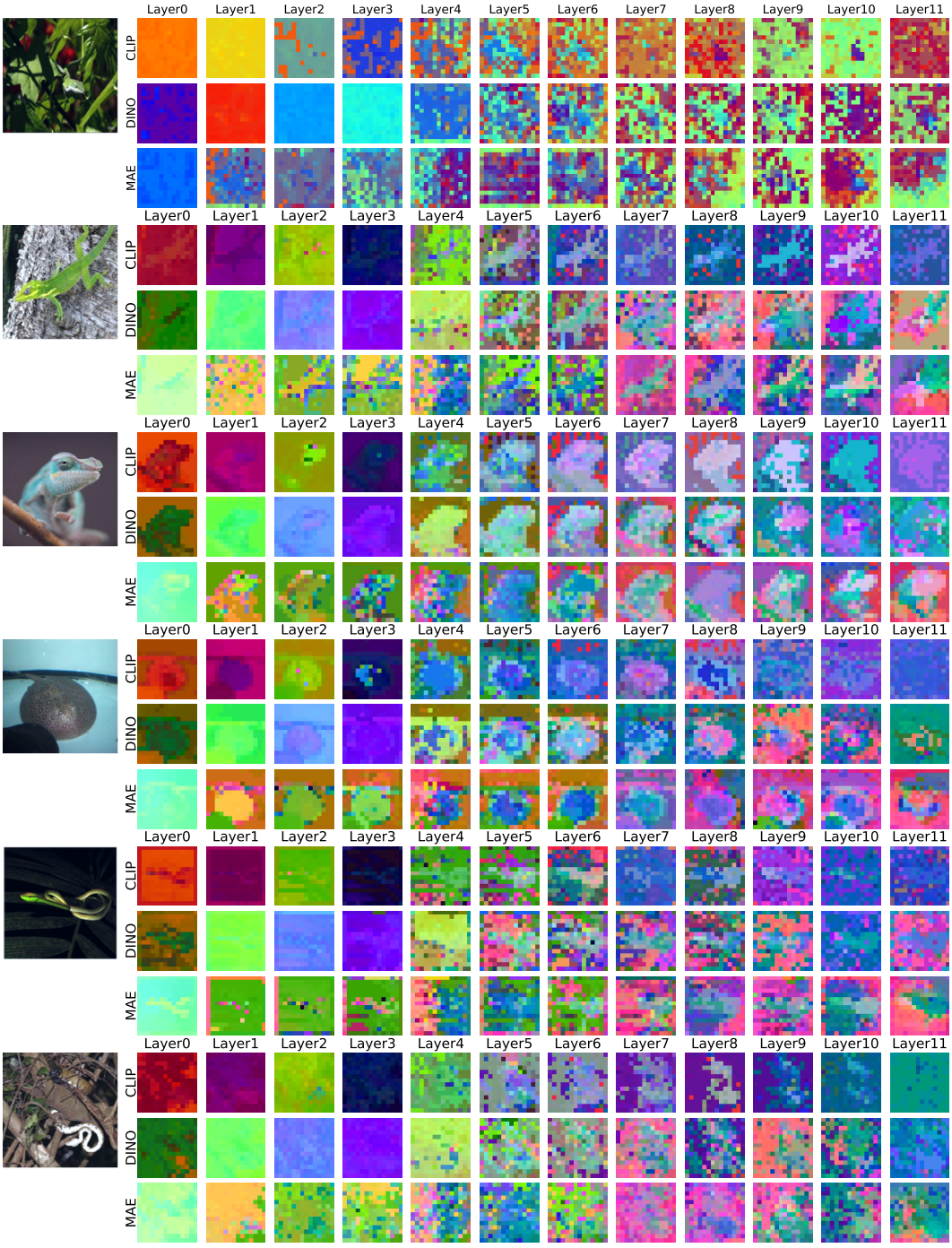

Figure 12: Spectral clustering in the universal channel aligned feature space. The image pixels are colored by our approach AlignedCut, the pixel RGB value is assigned by the 3D spectral-tSNE of the top 20 eigenvectors. The coloring is consistent across all images, layers, and models.

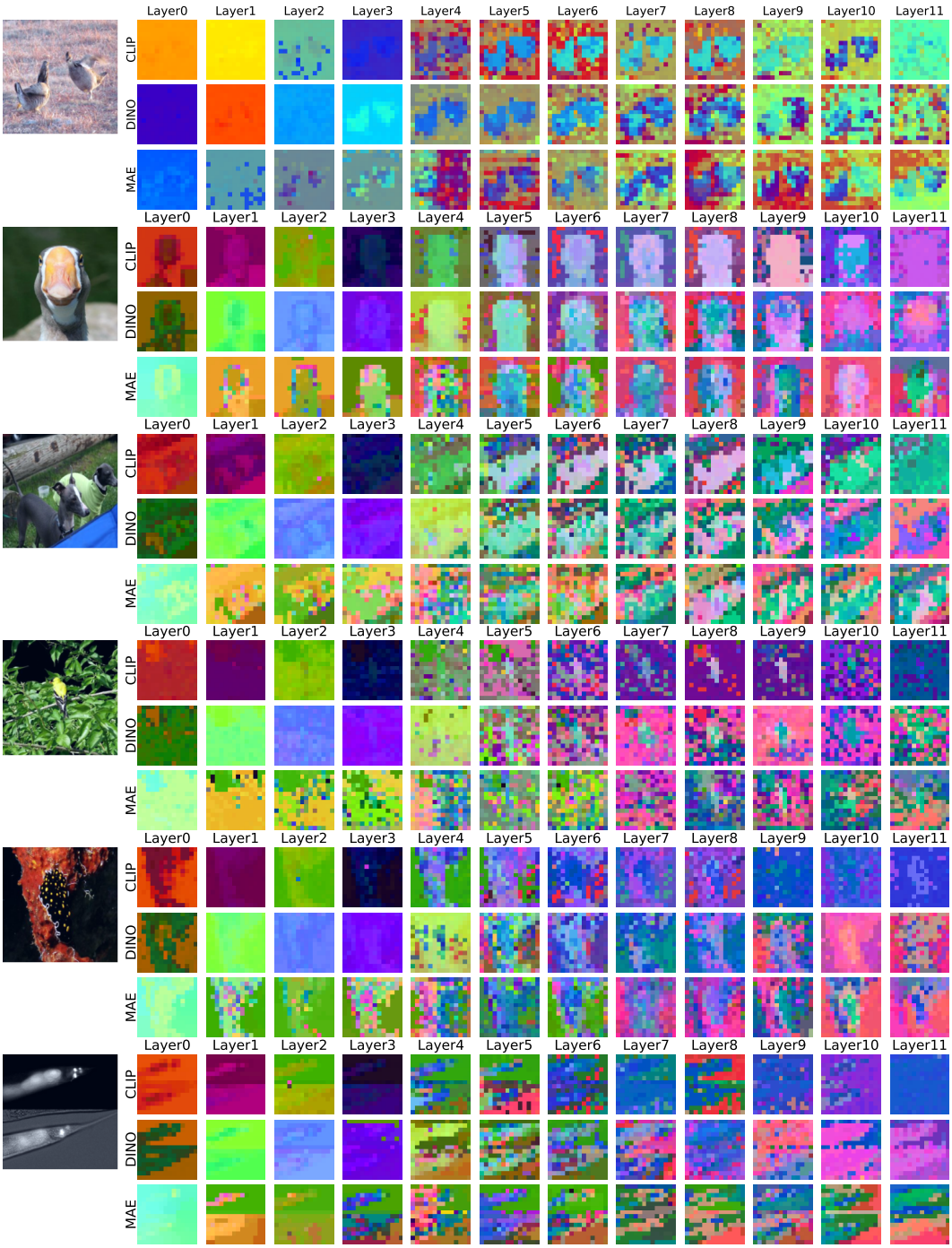

Figure 13: Spectral clustering in the universal channel aligned feature space. The image pixels are colored by our approach AlignedCut, the pixel RGB value is assigned by the 3D spectral-tSNE of the top 20 eigenvectors. The coloring is consistent across all images, layers, and models.

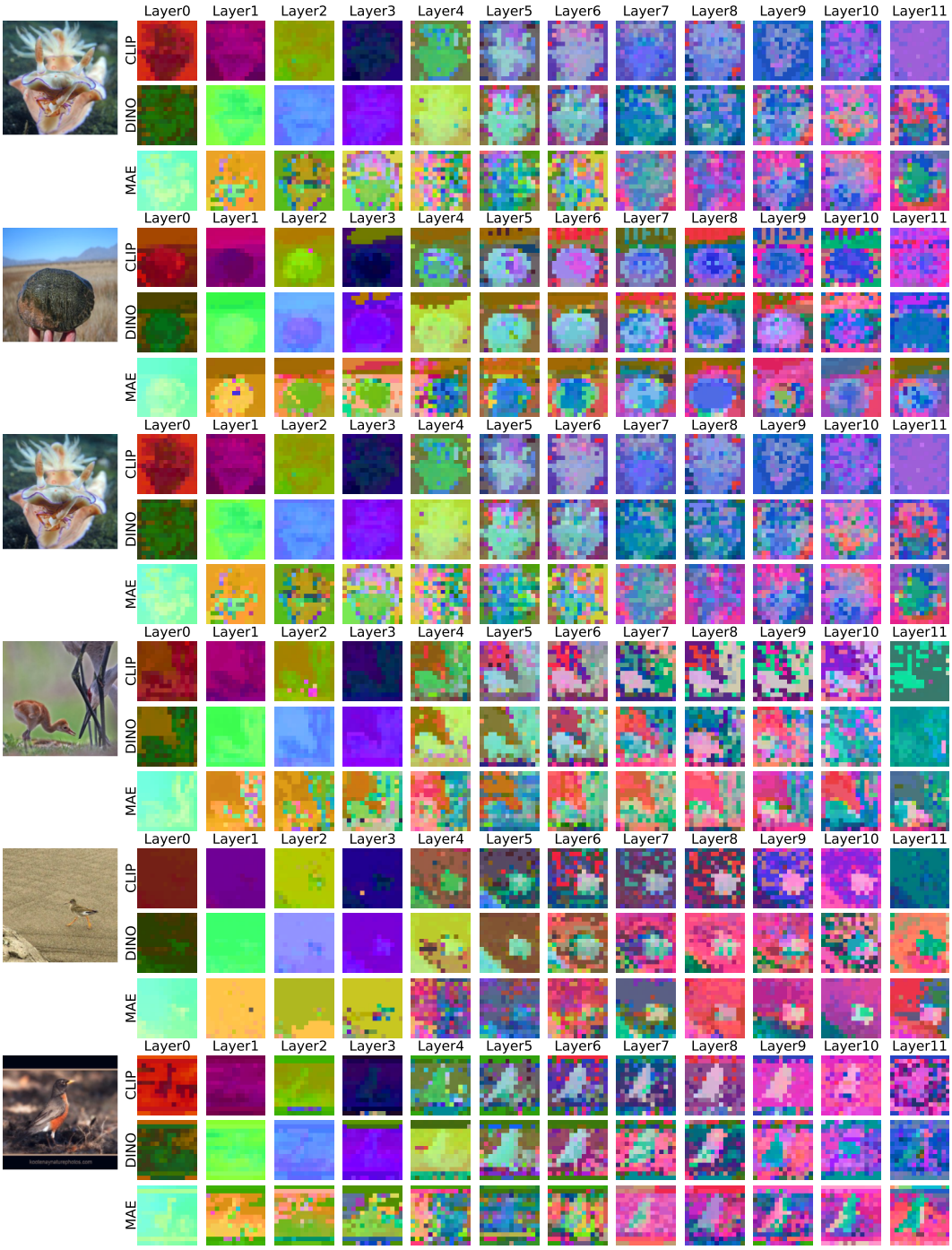

Figure 14: Spectral clustering in the universal channel aligned feature space. The image pixels are colored by our approach AlignedCut, the pixel RGB value is assigned by the 3D spectral-tSNE of the top 20 eigenvectors. The coloring is consistent across all images, layers, and models.

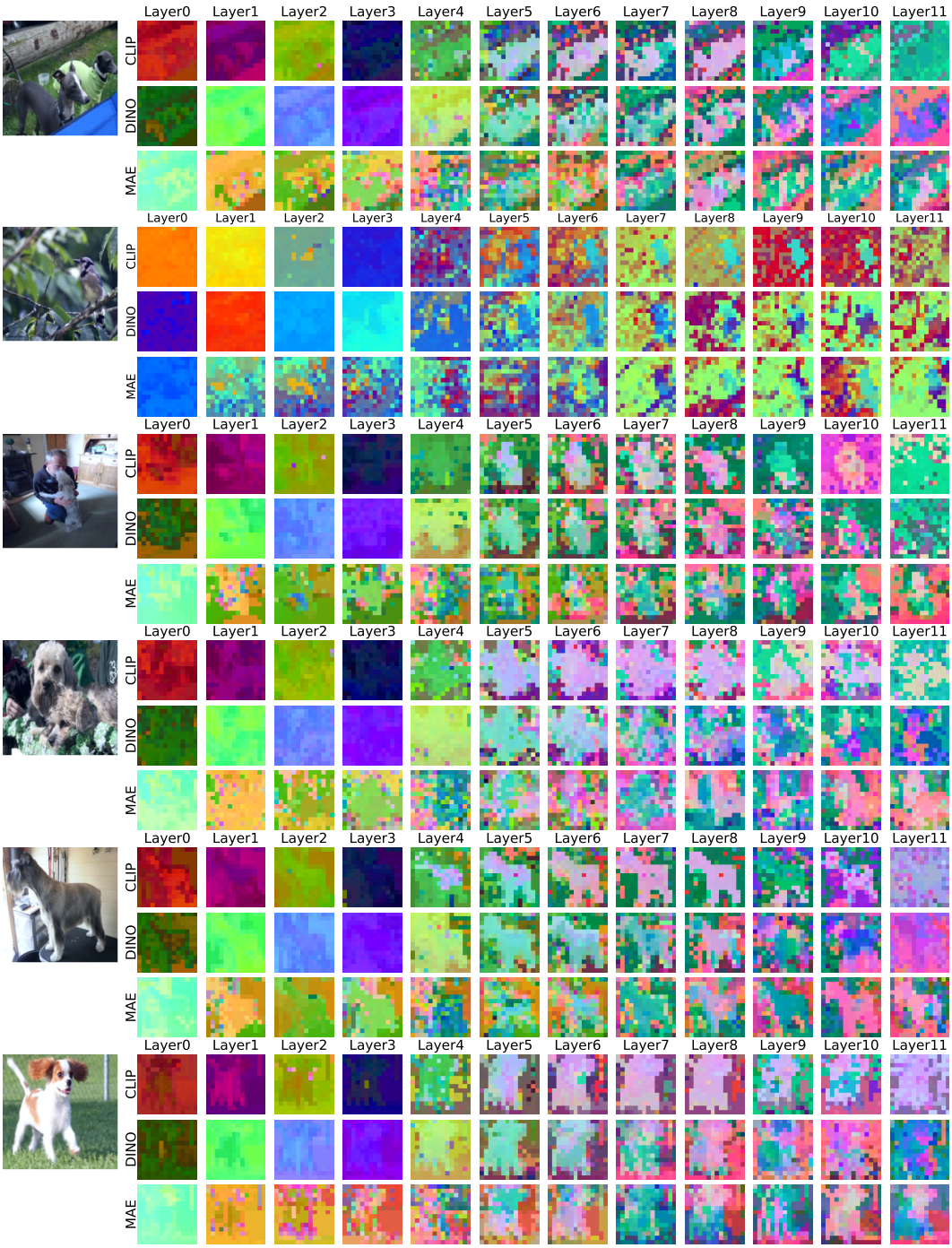

Figure 15: Spectral clustering in the universal channel aligned feature space. The image pixels are colored by our approach AlignedCut, the pixel RGB value is assigned by the 3D spectral-tSNE of the top 20 eigenvectors. The coloring is consistent across all images, layers, and models.

E Figure-ground Channel Activation from All Layers and Models

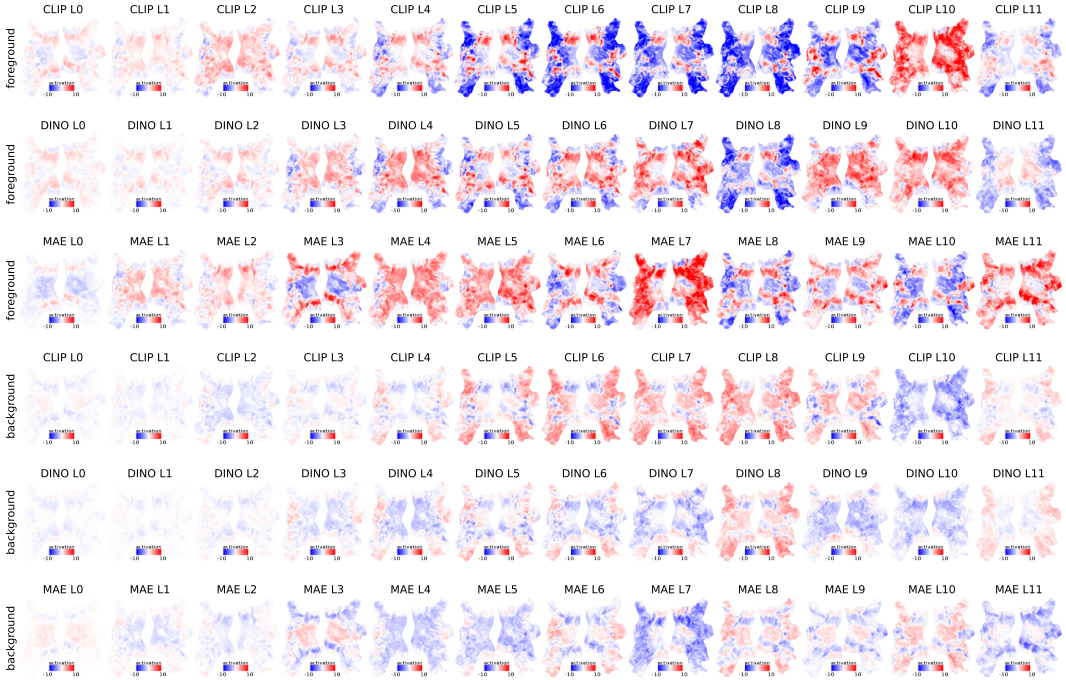

Figure 16: Mean activation of foreground or background pixels at each layer of CLIP, DINO and MAE. Channel activations are linearly transformed to the brain's space. Large absolute activation value means more consistent visual concepts.

F Visual Concepts: Categories

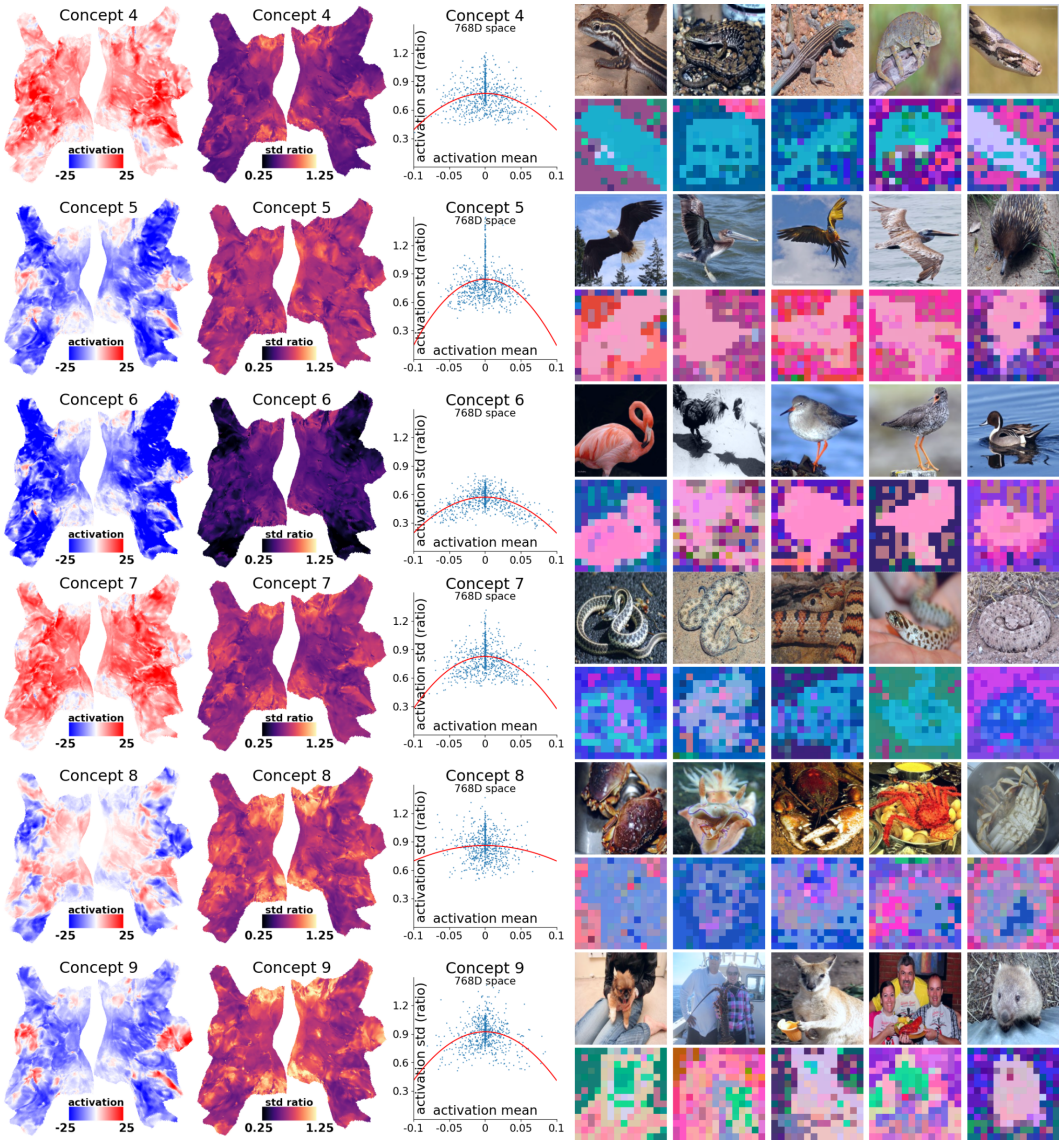

Figure 17: Category visual concepts in CLIP Layer 9. ***Left:*** Mean activation of all pixels within an Euclidean sphere centered at the visual concept in the 3D spectral-tSNE space; the concepts activate different brain regions. ***Middle:*** The standard deviation negatively correlates with absolute mean activations. ***Right:*** Spectral clustering, colored by 3D spectral-tSNE of the top 20 eigenvectors.

G Layer-to-Layer Feature Computation Flow in 2D spectral-tSNE space

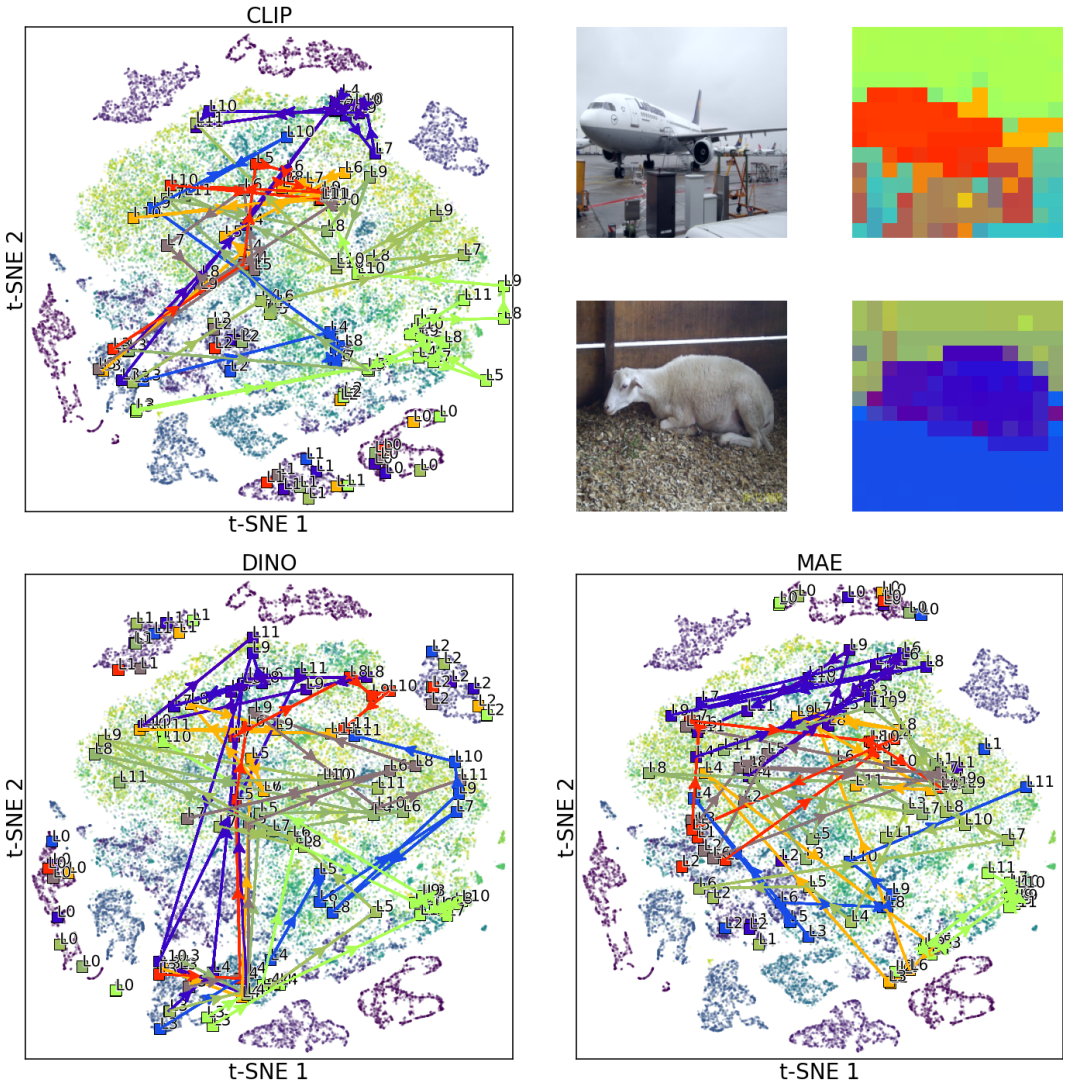

Figure 18: Trajectory of feature progression in from layer to layer, in the 2D spectral-tSNE space. Arrows displayed for 10 randomly sampled example pixels. ***Top Right:*** Pixels are colored by unsupervised segmentation.

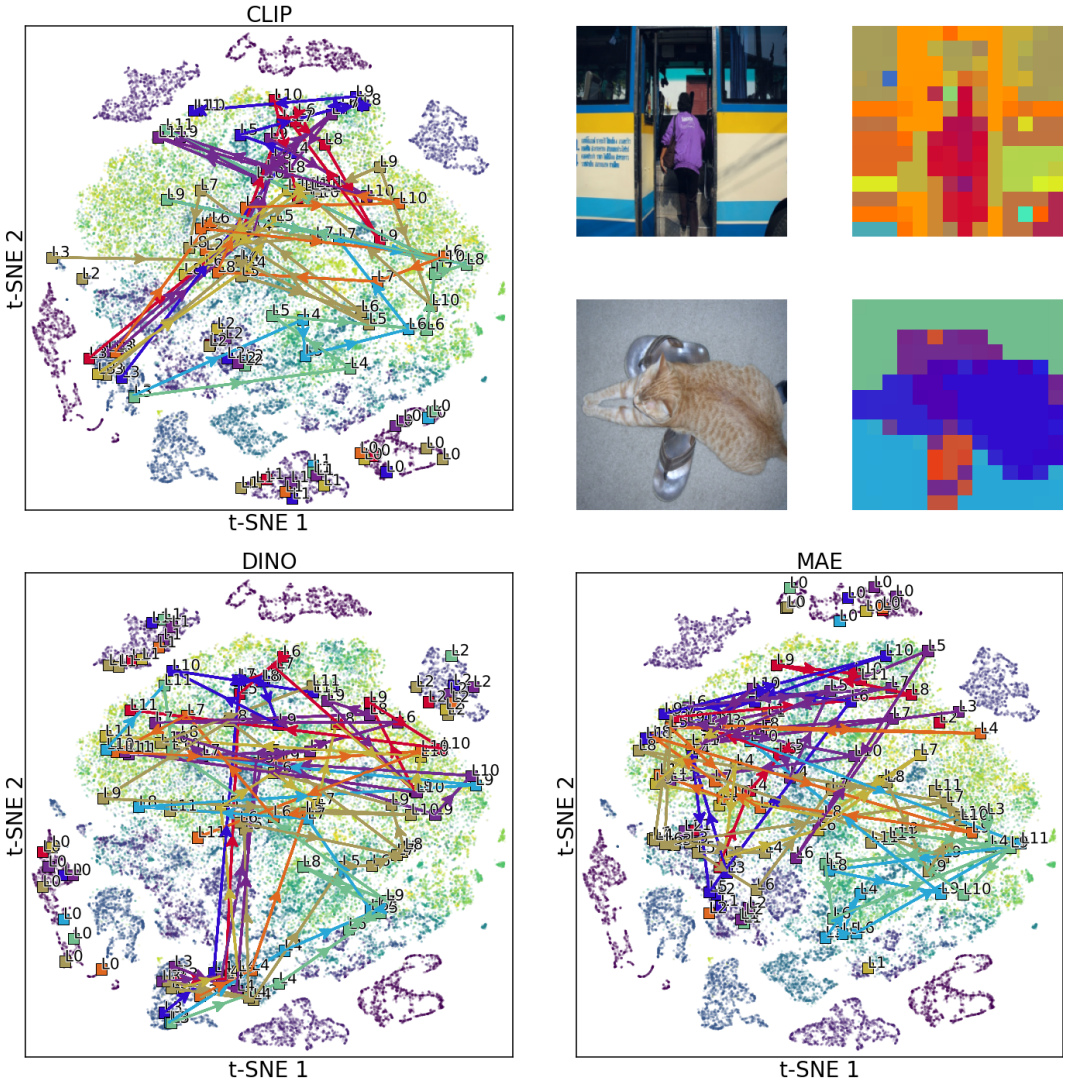

Figure 19: Trajectory of feature progression in from layer to layer, in the 2D spectral-tSNE space. Arrows displayed for 10 randomly sampled example pixels. ***Top Right:*** Pixels are colored by unsupervised segmentation.

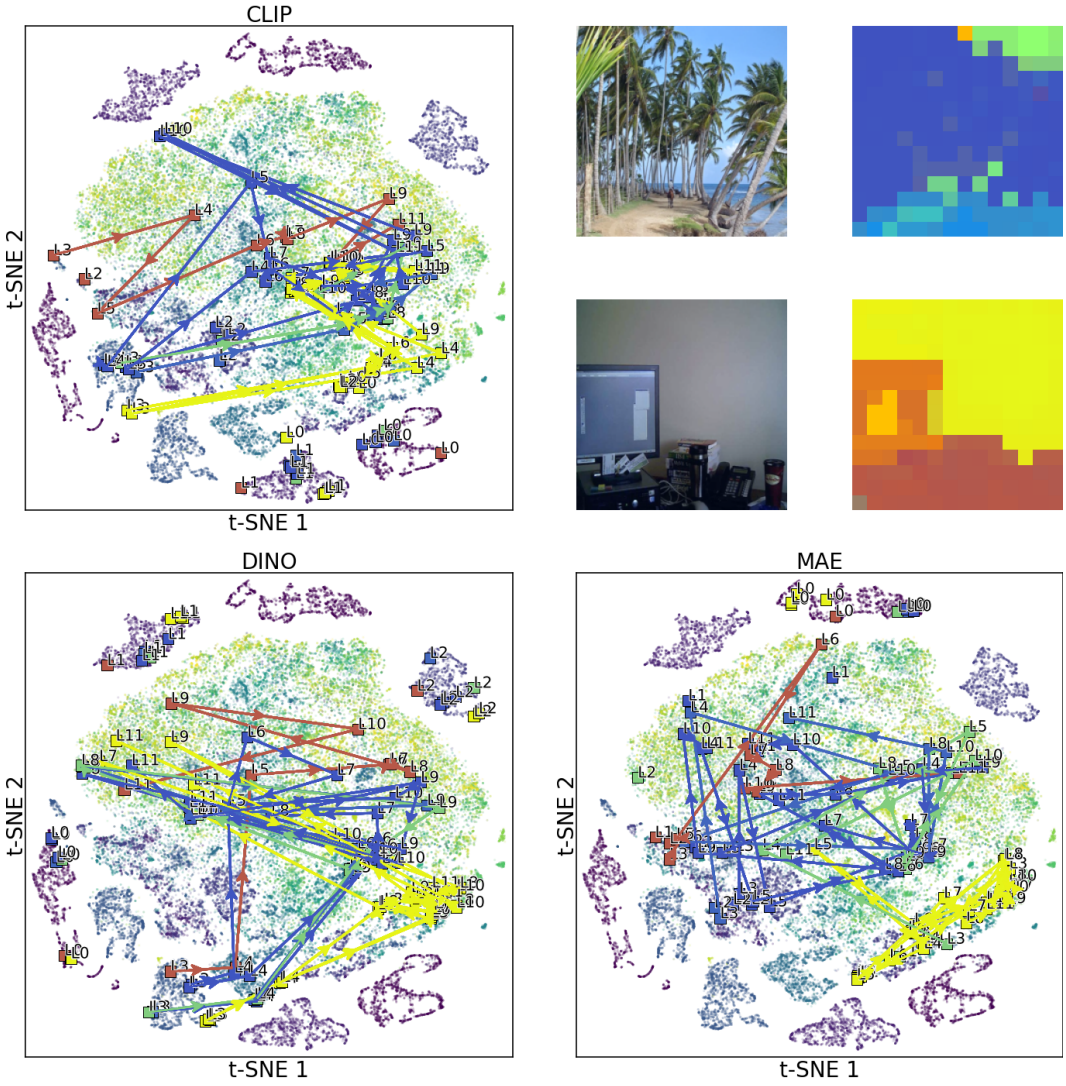

Figure 20: Trajectory of feature progression in from layer to layer, in the 2D spectral-tSNE space. Arrows displayed for 10 randomly sampled example pixels. ***Top Right:*** Pixels are colored by unsupervised segmentation.

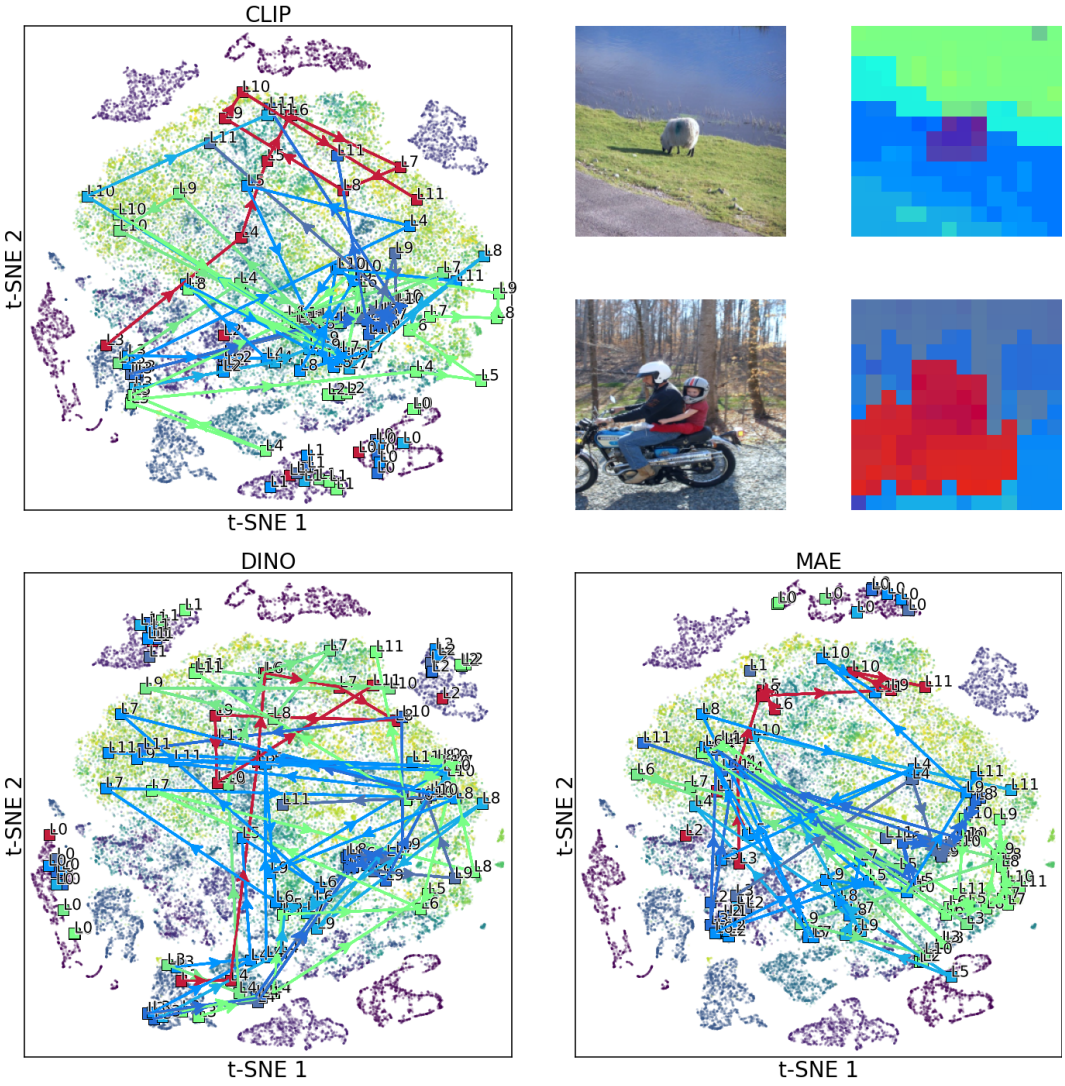

Figure 21: Trajectory of feature progression in from layer to layer, in the 2D spectral-tSNE space. Arrows displayed for 10 randomly sampled example pixels. ***Top Right:*** Pixels are colored by unsupervised segmentation.

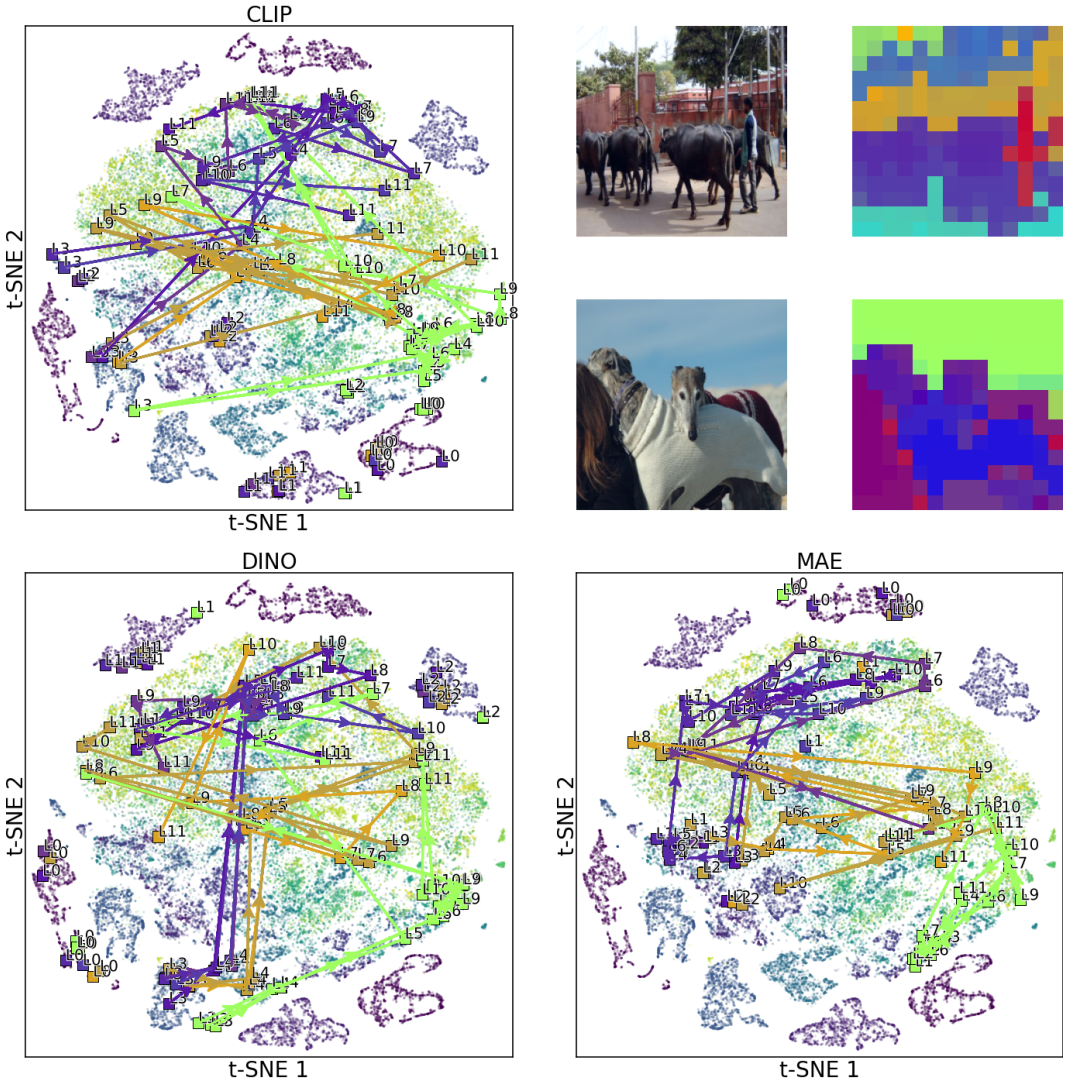

Figure 22: Trajectory of feature progression in from layer to layer, in the 2D spectral-tSNE space. Arrows displayed for 10 randomly sampled example pixels. ***Top Right:*** Pixels are colored by unsupervised segmentation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details are in the appendix

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The data is provided as open-source from another study; our code is not yet released, it will be released upon publication.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Experimental details are in the appendix

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provided standard deviation in Table 1, measured over training with 3 random seed.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.