Interpreting ResNet-based CLIP via Neuron-Attention Decomposition

Edmund Bu
UC San Diego
ebu@ucsd.edu

Yossi Gandelsman
UC Berkeley
yossi_gandelsman@berkeley.edu

Abstract

We present a novel technique for interpreting the neurons in CLIP-ResNet by decomposing their contributions to the output into individual computation paths. More specifically, we analyze all pairwise combinations of neurons and the following attention heads of CLIP's attention-pooling layer. We find that these neuron-head pairs can be approximated by a single direction in CLIP-ResNet's image-text embedding space. Leveraging this insight, we interpret each neuron-head pair by associating it with text. Additionally, we find that only a sparse set of the neuron-head pairs have a significant contribution to the output value, and that some neuron-head pairs, while polysemantic, represent sub-concepts of their corresponding neurons. We use these observations for two applications. First, we employ the pairs for training-free semantic segmentation, outperforming previous methods for CLIP-ResNet. Second, we utilize the contributions of neuron-head pairs to monitor dataset distribution shifts. Our results demonstrate that examining individual computation paths in neural networks uncovers interpretable units, and that such units can be utilized for downstream tasks.

1 Introduction

Interpreting the hidden components in pre-trained deep neural networks, by tracing their contribution to the model output, allows us to detect model limitations and useful sub-computations that can be repurposed for multiple downstream tasks (Lindsey et al. [2025], Sharkey et al. [2025]). Recently, such approaches were applied to CLIP - a widely used class of image encoders (Gandelsman et al. [2024b], Bhalla et al. [2024], Gandelsman et al. [2024a]). These approaches unlocked various capabilities - discovery of spurious correlations in the model output, automated generation of adversarial attacks, and even reuse of the interpreted model components for segmentation.

While CLIP interpretability work showed promising results for vision transformer-based variants (CLIP-ViT), such existing methods do not readily extend to the ResNet counterparts (CLIP-ResNet). Most of these existing methods rely on a decomposition of the output into a sum of per-layer contributions, which is not possible for CLIP-ResNet as, despite its additive residual connections, each layer is followed by a non-linearity. Moreover, CLIP-ViT interpretability methods rely on the model attention blocks and a special class token, while CLIP-ResNet models have convolutions and final attention pooling instead. These architectural differences make existing methods not applicable.

To address this gap, we introduce a new approach that provides a fine-grained decomposition of CLIP-ResNet model outputs into individual contributions of neurons in the last layers and the following attention-pooling heads. More specifically, we show that the output of CLIP-ResNet is a sum over computation paths, ranging from individual neurons in the last layer, in parallel through all the attention heads in the attention pooling layer, to the output. As each such neuron-head contribution lives in the joint image-text space, it can be compared and interpreted via text.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Mechanistic Interpretability.

We analyze the contributions of each neuron-head pair to the output and find that, unlike CLIP-ResNet's individual neurons, neuron-head pairs can be approximated by a *single direction* in the joint image-text embedding space. Additionally, we find that a fixed sparse set of neuron-head pairs comprises most of the output value (mean-ablating a fixed bottom 80% of neuron-head pairs decreases ImageNet classification accuracy by only 5%). We run analogous experiments for neuron contributions and find these properties to be unique to neuron-head pairs. Furthermore, we discover that some neuron-head pairs encompass sub-concepts of the concepts that the corresponding neuron represents (e.g., a 'butterfly' neuron can be decomposed into a 'butterfly clothing' neuron-head pair and other sub-concepts).

We leverage our findings for two applications: semantic segmentation and monitoring dataset distribution shifts. Given the approximation of each neuron-head pair by a single direction, we can rank each pair by its similarity to text representations. We apply this to associate each class with a set of k neuron-head pairs, and use this association for dense semantic segmentation. Notably, we achieve a 15% relative improvement in mIoU over previous methods for training-free semantic segmentation using CLIP-ResNet. We also conduct a case study on monitoring dataset distribution shift, in which we show that neuron-head pair contributions closely track the ground truth of the concepts they represent, and are different between the classes.

In summary, we propose neuron-attention decomposition as an improved interpretability method to automatically label the components of CLIP-ResNet's vision encoder with text. We find evidence that our decomposition is favorable in comparison to decomposing solely neurons or solely attention heads, and apply this decomposition to two relevant applications. This demonstrates the viability of examining fine-grained computation paths for studying and enhancing model capabilities.

2 Related Work

Contrastive vision-language models. Models like CLIP (Radford et al. [2021]) and its variants (Jia et al. [2021], Zhai et al. [2023]) are trained on massive web-based datasets of images and their captions to learn meaningful image representations. This pre-training enables zero-shot capabilities for downstream tasks like OCR, geolocalization, and classification (Wortsman [2023]). These models are also used as backbones for other systems such as LLaVA (Liu et al. [2023]), 3-D learning (Zhu et al. [2023]), and image generation (Ramesh et al. [2021], Rombach et al. [2022]).

Interpreting vision models. Mechanistic interpretability is a field of research that seeks to understand the inner workings of neural networks by analyzing fundamental model components and computation paths. Early mechanistic interpretability discoveries for vision models include the attribution of high-level concepts to intermediate model neurons (Bau et al. [2017]), and curve detectors and circuits (Cammarata et al. [2020]). Similar to us, a body of work aims to automatically label vision model components with text (Hernandez et al. [2022], Bills et al. [2023], Oikarinen and Weng [2024]).

Interpreting CLIP. Several recent works investigate CLIP's embedding space with techniques like sparse coding (Bhalla et al. [2024]) and factor rotation (Zhao et al. [2025]), aiming to identify human-interpretable concepts. Most closely to us, previous work analyzing CLIP-ViT's output decomposition (Gandelsman et al. [2024a], Gandelsman et al. [2024b]) by utilizing the additive linearity of the residual stream to examine contributions of individual components to the embedding space, and relied on the joint output image-text space to interpret such components with text. Differently from these methods, we focus on ResNet-based models, for which the contributions of the early layers are not additive to the output (due to a ReLU non-linearity following each residual connection). However, we *are* able to interpret CLIP-ResNet's last convolutional block and its attention pooling, where we propose a more fine-grained view of its inner workings by analyzing neuron-head pairs.

3 Methodology

We start by presenting CLIP-ResNet's architecture and deriving the decomposition of its image representation. We use this decomposition in later sections to interpret the contributions of individual pairs of neurons and attention heads.

3.1 CLIP-ResNet preliminaries

Contrastive pre-training. CLIP is trained via a contrastive loss that aligns the representations of its image encoder M_{image} with the representations of its text encoder M_{text} in a shared image-text latent space $\mathbb{R}^d$. Specifically, over massive web-based datasets, the two encoders are trained together to maximize the cosine similarity for matching image-text pairs (I, t):

$$\text{sim}(I, t) = \frac{\langle M_{\text{image}}(I), M_{\text{text}}(t) \rangle}{\|M_{\text{image}}(I)\|_2 \, \|M_{\text{text}}(t)\|_2}. \tag{1}$$

Zero-shot classification. To perform image classification, each class name c_j is mapped to some template (e.g., "A photo of a {class}") and encoded by the text encoder as $M_{\text{text}}(\text{template}(c_j))$ (for simplicity, we will omit the template notation). The classification prediction for an image I is the class c_j whose text representation $M_{\text{text}}(c_j)$ is most similar to the image representation $M_{\text{image}}(I)$.

CLIP-ResNet. The CLIP-ResNet image encoder is a traditional ResNet network (He et al. [2016]), composed of sequential *residual blocks* with an average pooling replaced by attention pooling (Radford et al. [2021]). While CLIP is often trained with ViT as the image encoder backbone, CLIP-ResNet is competitive in performance across various benchmarks. However, the internal mechanisms of CLIP-ResNet are underexplored in comparison to CLIP-ViT, and existing methods are not applicable due to architectural differences.

CLIP-ResNet architecture. CLIP-ResNet's residual stream is not linear, as a nonlinear ReLU activation follows the additive residual connection within each residual block. Thus, we focus only on the final residual block's input to the attention pooling layer, which we *can* decompose linearly. Formally, given an input image I, let $Z(I)$ be the output of the last convolutional layer in the model, and let $Z'(I)$ be $Z(I)$ after prepending a class token and adding positional embedding. CLIP-ResNet's image representation is the $Z'(I)$ with an attention pooling applied to it:

$$M_{\text{image}}(I) = \text{AttnPool}(Z'(I)). \tag{2}$$

More specifically, the dimensionality of $Z(I)$ is $C \times H' \times W'$ where C is the number of *neurons* (post-ReLU per-location activations in the final convolutional block) and H' and W' are the spatial feature map dimensions. To form $Z'(I) \in \mathbb{R}^{(K+1)\times C}$, $Z(I)$ is first flattened into $K = H'W'$ image tokens $\{z_i\}_{i\in\{1,\dots,K\}}$, $z_i \in \mathbb{R}^C$, then a *class token* $z_0 = \frac{1}{K}\sum_{i=1}^{K} z_i$ is prepended to this sequence, and finally, a learned positional embedding is added to all $K+1$ tokens. Attention pooling is implemented as a standard transformer multi-head attention module, with the exception being that the class token is used as the sole output $M_{\text{image}}(I)$.

3.2 Decomposition into neurons, heads, and tokens

Decomposition into attention heads and tokens. Following Elhage et al. [2021] and leveraging the fact that only the class token is returned by attention pooling, we can write the image representation as a sum over H attention heads of the attention-pooling layer and $K+1$ tokens:

$$M_{\text{image}}(I) = \text{AttnPool}([z_0, \dots, z_K]) = \sum_{h=1}^{H}\sum_{i=0}^{K} a_i^h(I)\, z_i\, W_{VO}^h \tag{3}$$

where $W_{VO}^h \in \mathbb{R}^{C\times d}$ are transition matrices (the OV matrices) and a_i^h is a weight that denotes how much the class token attends to the i-th token ($\sum_{i=0}^{K} a_i^h = 1$).

Decomposition into neuron-head pairs. Each row of the W_{VO}^h matrix corresponds to one neuron. That means, for any given head and token, we can rewrite:

$$z_i\, W_{VO}^h = \sum_{n=1}^{C} z_i\, W_{VO}^{n,h} \tag{4}$$

where the superscript n denotes the n-th row of W_{VO}^h. Substituting into (3) and swapping the sum yields:

$$M_{\text{image}}(I) = \sum_{n=1}^{C}\sum_{h=1}^{H}\sum_{i=0}^{K} r_i^{n,h}(I), \quad r_i^{n,h} = a_i^h(I)\, z_i\, W_{VO}^{n,h}. \tag{5}$$

PC(s)	Accuracy (%)
(Baseline)	70.7
$\{\hat{r}^{n,h}\}$	70.7
$\{\hat{r}^{n}_{1}\}$	66.9
$\{\hat{r}^{n}_{1}, \hat{r}^{n}_{2}\}$	69.0
$\{\hat{r}^{n}_{1}, \hat{r}^{n}_{2}, \hat{r}^{n}_{3}\}$	69.7
$\{\hat{r}^{n}_{1}, \hat{r}^{n}_{2}, \hat{r}^{n}_{3}, \hat{r}^{n}_{4}\}$	70.0

Table 1: **Accuracy for reconstruction from principal components.** Unlike neuron representations, neuron-head representations match the baseline accuracy using only one direction for reconstruction.

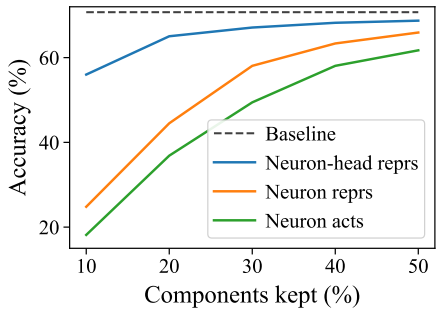

Figure 1: **Mean-ablation accuracy** Mean-ablating all but a fixed set of components shows there is a sparse set of neuron-head pairs that is very significant.

$r^{n,h}_{i}$ denotes the d-dimensional contribution to the output from the class token's attention to token i through head h, projected by the row of W^{h}_{VO} that corresponds to neuron n.

The image representation M_{image} lives in the joint image-text embedding space. Therefore, each neuron-head contribution $r^{n,h}$ obtained by summing over tokens, and each neuron contribution r^{n}, obtained by summing over tokens *and* heads, lives in the same image-text space. This allows us to compare them to text. We will use this property for automatically labeling neurons and neuron-head pairs in later sections. Summing over the neuron dimension, instead, gives the decomposition r^{h}_{i}, which corresponds to the spatial patches at locations $i \in \{0, \ldots, K\}$. This allows us to compute the similarity to text for each image location represented in the decomposed class token. We use this property to form our per-token segmentation map in Section 5.1.

Comparison to existing decompositions. Previous work on CLIP-ViT has studied the direct effect of decomposition across attention heads and tokens (Gandelsman et al. [2024b]), analogous to our decompositions r^{h}_{i} and r^{h}, as well as the second-order effect of decomposition into neurons (Gandelsman et al. [2024a]), analogous to our decomposition r^{n}. We show that our approach overcomes two main limitations of these methods, when applied to CLIP-ResNet: First, there is a small number of attention heads relative to possible concepts that CLIP learns, meaning that each head encodes multiple concepts, ultimately making them less interpretable (Lecomte et al. [2024]). Second, the second-order effect of neurons disregards the independent nature of attention heads in the multi-head attention layer, meaning r^{n} encompasses all paths through the network – which, as we will show next, captures multi-dimensional conceptual structure that is yet again difficult to interpret.

4 Analysis of individual components

We quantitatively evaluate contributions of individual components in our neuron-attention decomposition, and show their benefits over other decompositions – these contributions are approximately rank-1 and sparse. Additionally, we qualitatively investigate the top-activating images for selected components and discover that neuron-head pairs tend to represent sub-concepts of the concepts represented by the corresponding neuron. We later use these observations to produce semantic segmentation (Section 5.1) and monitoring distribution shift (Section 5.2).

4.1 Quantitative analysis

We study the properties of $r^{n,h}$ and r^{n}. We show that individual $r^{n,h}(I)$ can be approximated by a single direction in the image-text space, while approximating $r^{n}(I)$ with only one linear direction causes a significant drop in reconstruction fidelity. Moreover, we find that neuron-head pairs are contributing more sparsely than neurons-only – mean-ablating all but a subset of each component shows a sparser set of top-contributing neuron-head pairs in comparison to neurons.

Experimental setting. We measure the performance of zero-shot classification on the ImageNet (Deng et al. [2009]) validation dataset after various ablations to quantify the resulting change in the representation. We collect $r^{n,h}(I)$ and $r^{n}(I)$ contributions over the set $\mathcal{D}$, which is comprised

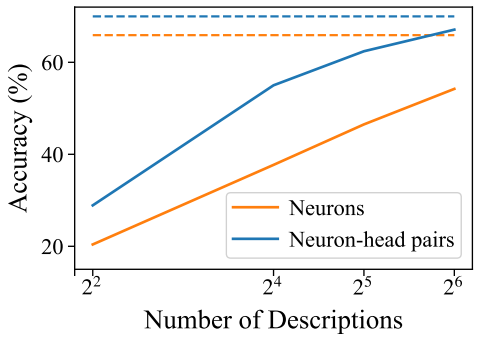

Figure 2: **Reconstruction from sparse text representations.** We compare sparse text decompositions for neurons and neuron-head pairs while varying description set sizes.

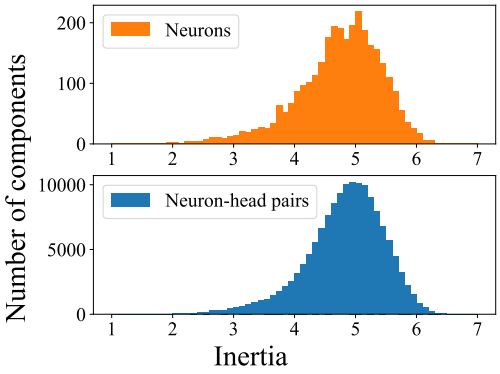

Figure 3: **Inertia as a proxy to compare polysemanticity**. We display bins at 0.1 intervals and compute cluster metrics on each component's top ten images by contribution norm from $\mathcal{D}$.

of 1000 images from the ImageNet test dataset. We use our collected representations to compute singular value decomposition, from which we obtain our principal components, and to compute mean contributions, for our mean-ablation (Nanda et al. [2023]) experiments. Additionally, after obtaining each neuron-head pair's principal component, we further decompose it using a sparse coding technique and evaluate its sparsity-reconstruction tradeoff. We perform all experiments using OpenAI's CLIP-RN50x16, and present additional results for RN50x64 in Appendix A.1.

Neuron-head contributions are one-dimensional. We find that $r^{n,h}(I)$ can be approximated by a single direction $\hat{r}^{n,h}$ in the joint embedding space. We approximate $r^{n,h}(I)$ for any given image I with $x^{n,h}(I)\,\hat{r}^{n,h} + b^{n,h}$, where $x^{n,h}(I)$ is the coefficient obtained from the projection norm of $r^{n,h}(I)$ onto $\hat{r}^{n,h}$, and $b^{n,h}$ is the mean of all $r^{n,h}(I)$ over all images $I \in \mathcal{D}$. As shown in Table 1, replacing $r^{n,h}(I)$ with this approximation results in no decrease in reconstruction quality, as measured by downstream ImageNet classification accuracy.

Neuron-head contributions are sparse. Keeping only 20% of neuron-head contributions, computed over ImageNet (while mean-ablating the rest), results in only a $\sim 5\%$ decrease in classification accuracy. We sort the $C \times H$ neuron-head pairs by the mean of their top percentile norms over $\mathcal{D}$, and observe the same high-scoring pairs $\mathcal{P}^*$ tend to be consistently important across the dataset. Decomposing $M_{\text{image}}(I)$ into individual $r^{n,h}(I)$ contributions, we keep only the $r^{n,h}(I)$ whose neuron-head pairs (n,h) are in $\mathcal{P}^*$ and mean-ablate the rest. We construct $\mathcal{P}^*$ with varying top percentage norms (at 10% increments) and measure the resulting classification accuracy on ImageNet validation in Figure 1. As shown, most of the output value can be recovered from a sparse set of neuron-head pairs.

Sparse text-based decomposition of neuron-head directions. Following Gandelsman et al. [2024a], we use orthogonal matching pursuit (Pati et al. [1993]) to further decompose each $\hat{r}^{n,h}$ direction into sparse text components. Formally, we use a sparse set of text components $\{t_j\}_{j\in\{1,m\}}$ to approximate $\hat{r}^{n,h} \approx \sum_{j=1}^{m} \gamma_j^{n,h}\, M_{\text{text}}(t_j)$, where $\gamma_j^{n,h}$ is a non-zero scalar coefficient. In our experiments, the initial pool of text descriptions is composed of the 30,000 most common English words. We vary the sparse set size m and present classification accuracy performance on ImageNet, after replacing $\hat{r}^{n,h}$ with our sparse approximation in Figure 2. As shown, using $m = 64$ text descriptions for neuron-head pairs surpasses the neuron baseline in reconstruction accuracy, exemplifying the benefits of our fine-grained decomposition.

Comparison to neuron-only contributions. We repeat the two experiments above for neuron-only decomposition. To include the variance explained by multiple principal components, we simply reconstruct from a set $\{\hat{r}_1^n, \hat{r}_2^n, \ldots\}$, where $\hat{r}_k^n$ is the k-th principal component. We use $\hat{r}_1^n$ for all text-based sparse decomposition experiments. Compared to our findings above for neuron-head pairs, the first principle component $\hat{r}_1^n$ explains much less variance – it reconstructs with a $\sim 4\%$ drop in accuracy from the baseline. Additionally, the contributions are less sparse – our mean-ablation experiment shows a nearly 38% accuracy difference between top 10% neurons and top 10% neuron-

Figure 4: **Images with largest contribution norm for attention heads, neurons, and neuron-head pairs.** We present the top images from ImageNet validation set. Neuron-head pairs correspond to specific subcategory concepts of neurons (e.g., 'butterfly *clothing*' in row 3) and similar concepts to their neurons (e.g., 'router' for neuron $\#2384$ and 'people' for neuron $\#1300$)

head pairs (Figure 1). We also show results for mean-ablation of neuron activations, which performs even worse. Finally, as shown in Figure 2, neurons show less sparsity in reconstruction from text components.

4.2 Qualitative analysis

We qualitatively analyze the images I across the ImageNet validation dataset that provide the highest contribution $r^h(I)$, $r^n(I)$, or $r^{n,h}(I)$ in norm. In Figure 4, we select pair $\#(624, 21)$ as the pair with the top cosine similarity to $M_{\text{text}}(\text{“}butterfly\text{''})$, and choose the other two pairs from four randomly selected pairs. We note that, as shown in the figure, the top-activating images for a given neuron-head pair appear similar to those of its corresponding neuron, but not its corresponding head. Notably, the neuron $\#624$ is most active on images that represent the 'butterfly' concept, and the neuron-head pair $\#(624, 21)$ is most active for the *sub-concept* 'butterfly clothing'. These examples highlight the ability of neuron-attention decomposition to isolate semantically meaningful directions in CLIP's embedding space.

Components	Text descriptions
Attention head #21	“slr", “vantage", “jetta"
Neuron #624	“wings" (+1.73), “butterfly" (+1.69), “kite" (+1.20)
Neuron-head #(624, 21)	“butterfly" (+2.64), “roses" (−1.17), “grizzlies" (+0.91)
Attention head #35	“musicians", “motorcycles", “archery"
Neuron #2384	“kobe" (+1.00), “alfa" (−0.98), “redmond" (+0.91)
Neuron-head #(2384,35)	“routers" (+2.10), “tutorials" (−0.99), “spur" (−0.94)
Attention head #46	“salford", “jcpenney", “chattanooga"
Neuron #1300	“smileys" (+1.60), “masterpieces" (+1.31), “affiliated" (+1.16)
Neuron-head #(1300,46)	“cnn" (+1.26), “cto" (−1.25),“varsity" (−0.99)

Table 2: **Sparse text-based decomposition examples for Figure 4 components.** We select from the descriptions detailed in Section 4.1 with sparsity $m = 64$ and use TextSpan (Gandelsman et al. [2024b]) to obtain descriptions for each attention head.

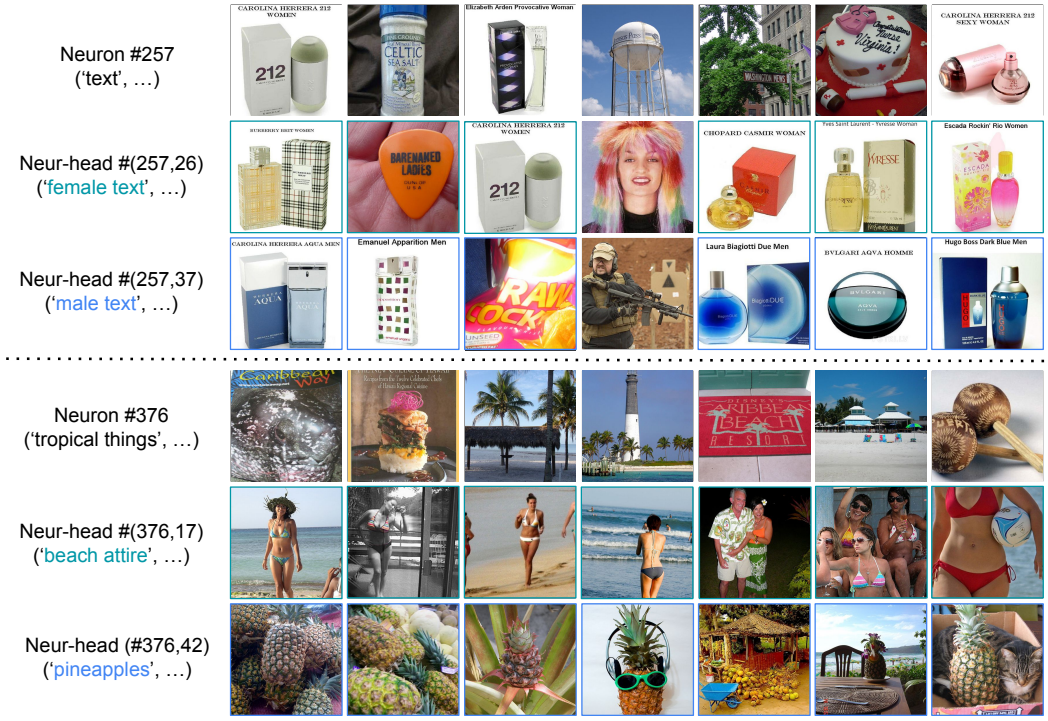

Figure 5: **Sub-concept neuron-head pairs for select neurons.** We again present top images by contribution norm and depict sub-concept relationships for two different neurons.

Neuron-head pairs remain polysemantic. We observe that neuron-head pairs remain polysemantic (for instance, in Figure 4, the 6th image for pair $\#(624, 21)$ and the 9th image for pair $\#(2384, 35)$ show polysemanticity). We collect the top ten images by contribution norm, from $\mathcal{D}$, for each neuron and neuron-head pair, and present the inertia of the normalized image embeddings (see Figure 3). Lower inertia implies a tighter cluster and less polysemanticity. By raw count, there are far more less-polysemantic neuron-head pairs than neurons, an advantage brought about by neuron-attention decomposition.

Neuron-head concepts are sub-concepts of neuron concepts. Inspired by the ‘butterfly clothing’ pair in Figure 4, we search for more image retrieval examples of sub-concept neuron-head pairs. We detail this process in Appendix A.3 and present two examples in Figure 5 and the rest in the appendix.

Figure 6: **Class segmentation maps for 'dog'.** Our method of multiplying two heatmaps together mitigates the failure modes of either one. Here, those failure modes are 1) the focus on the cat shown by $\sum_{r=1}^{k} Z^{n_r}(I)$ and 2) the negative localization shown by $\sum_{r=1}^{k} L_{\text{sim}}^{h_r}(I)$.

We find several sub-concept pairs in all of the randomly selected neurons, demonstrating that our neuron-attention decomposition captures fine-grained sub-concepts.

5 Applications

5.1 Semantic segmentation

We use the observation from above to repurpose CLIP for training-free semantic segmentation. Existing interpretability-based approaches used for segmentation (Gandelsman et al. [2024b], Gandelsman et al. [2024a], Helbling et al. [2025]) focus on ImageNet-Segmentation Guillaumin et al. [2014] or other single-class tasks. We aim to show the viability of interpretability for multi-class segmentation tasks, validating our approach in the paradigm of training-free CLIP semantic segmentation (Appendix A.4). Unlike related works in this area, we do not alter any computation within CLIP (i.e., we do not use self-self attention). We use CLIP's actual output decomposition and its similarity to text.

Method. We collect two separate features from the model: the final-layer activation map $Z(I)$ with dimensionality $C \times H' \times W'$, and the per-head contributions from each of the $K = H'W'$ image patches $\{r_i^h(I)\}_{i \in \{1,\dots,K\}}$. We compute the per-head segmentation heatmap $L_{\text{sim}}(I)$ for class name t_j by calculating $\langle r_i^h(I),\ M_{\text{text}}(t_j)\rangle$ individually for all pairs of tokens i and heads h, and then aggregating these similarity maps such that $L_{\text{sim}}(I) \in \mathbb{R}^{H \times H' \times W'}$.

Next, we refine $Z(I)$ and $L_{\text{sim}}(I)$ using our neuron-attention decomposition. We select the top-k neuron-head pairs by cosine similarity to class t_j and denote the ordering of these pairs by $(n_1, h_1), (n_2, h_2), \dots, (n_k, h_k)$. Then, the segmentation logits for class t_j are:

$$\hat{L}(I) = \sum_{r=1}^{k} Z^{n_r}(I) \circ L_{\text{sim}}^{h_r}(I) \tag{6}$$

where the superscripts n_r and h_r denote the top r-th neuron or head.

Implementation details. We evaluate our method on the PASCAL Context dataset (Mottaghi et al. [2014]). Similar to related works, we adopt a slide inference image pre-processing approach, where we specifically resize images to have a shorter side of 512 and then use a 384×384 window with a 192 stride. Additionally, we do not modify the class names in any way, both to select our top-k neuron-head pairs and to compute cosine similarity.

We report all results by selecting the top $k = 20000$ neuron-head pairs to text. We defer the effects of varying k, as well as the effect of intervening on register neurons (Darcet et al. [2023], Jiang et al. [2025]), to Appendix A.5.

Main results. As shown in Table 3, our method outperforms previous methods for semantic segmentation using CLIP-ResNet. To ensure fair comparison, we evaluate MaskCLIP on the same slide inference setup detailed in the previous paragraph, and use the ResNet50x16 backbone across all our experiments. We also present the results reported by SC-CLIP (Bai et al. [2024]), which is the current state-of-the-art method that uses CLIP-ViT. However, we stress that SC-CLIP leverages self-self attention, which, as shown, performs poorly when applied to CLIP-ResNet.

Method	mIoU(%)	Backbone
Self-self	22.2	RN50x16
MaskCLIP	22.8	RN50x16
Ours	26.2	RN50x16
SC-CLIP	40.1	ViT-B/16

Table 3: **Semantic segmentation performance on PASCAL Context.** For fair comparison, we implement MaskCLIP (Zhou et al. [2022]) and $QQ^T + KK^T$ attention on the same slide inference setup we use. SC-CLIP, the current state-of-the-art, uses a ViT backbone.

Method	mIoU (%)
Neuron maps only	16.5
Head maps only	24.7
Both (multiplied)	26.2

Table 4: **mIoU comparison of decomposition-based methods.** Segmentation by multiplying neuron features by head features performs better than by solely heads or solely neurons.

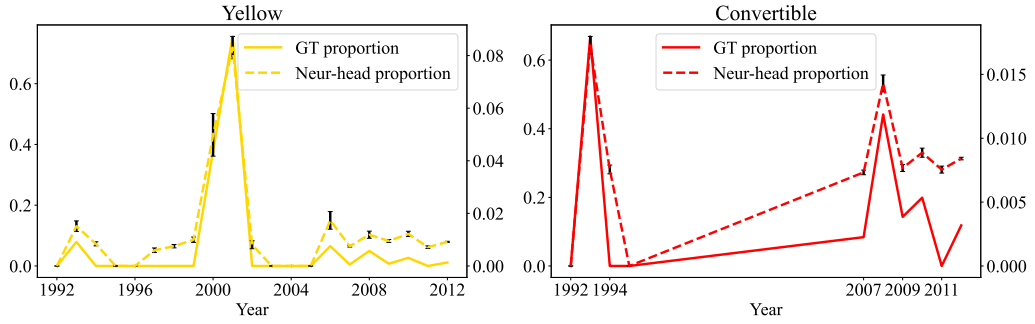

Figure 7: **Monitoring the distribution shift of Stanford Cars.** We compare the ground truth concept prevalence (left y-axis) to the mean proportional contribution (in norm) of top neuron-head pairs for a given concept (right y-axis).

We present qualitative heatmaps from which we compute the segmentation maps in Table 4. Our neuron-attention decomposition method performs better a single aggregated segmentation map, verifying the efficacy of our method in reducing noise and focusing on the correct class.

5.2 Monitoring distribution shift

As an additional application, we utilize the interpreted neurons for monitoring distribution shift between datasets. Following Bhalla et al. [2024], we consider the Stanford Cars dataset (Krause et al. [2013]) and track the neuron-head contributions for different categories over time (see Figure 7). Specifically, we choose the top $k = 5$ neuron-head pairs for a concept by cosine similarity, and then compute these pairs' contribution norm divided by the contribution norm of the model output itself. We compare these scores to the ground truth proportion of concepts per-year. As shown, the norms qualitatively follow similar trends to the actual distribution shift, which allows us to monitor it and summarize with text. Quantitatively, the average point-biserial correlation coefficient between the two proportions is 0.85 for 'yellow' and 0.71 for 'convertible', computed across applicable years. Notably, these concept contributions (along with tens of thousands more we do not analyze) are collected in a single forward pass per image, which makes this approach suitable for large datasets.

6 Limitations, discussion, and future work

We conclude by presenting two limitations of our approach and discussing future work.

Analyzing previous layers. Our approach is only applicable to the last layer of the ResNet. Earlier convolutional blocks could give us a more complete understanding of the model's computation. This is especially relevant for semantic segmentation, where related CLIP-ViT methods leverage the spatial consistency of intermediate layers for improved performance. Nevertheless, the neurons of the last layer are still useful for various downstream tasks, as shown above.

Neuron-head pairs remain polysemantic. Existing literature (Yuksekgonul et al. [2023], Park et al. [2024]) shows evidence that models like CLIP encode concepts additively in their embedding space.

While some neuron-head pairs appear less polysemantic in image retrieval examples, we qualitatively observe polysemanticity in other examples and Figure 3.

Discussion and future work. We presented a method to analyze specific, relatively interpretable, circuits in CLIP – neuron-head pairs. Extending and finding the correct minimal component that will be the most useful is an ongoing research question. We seek to scale and automate the assignment of fine-grained labels to such components in future work.

Acknowledgements. EB is grateful to XLab's Summer Research Fellowship at the University of Chicago for making this work possible. YG is supported by the Google Fellowship.

References

Sule Bai, Yong Liu, Yifei Han, Haoji Zhang, and Yansong Tang. Self-calibrated clip for training-free open-vocabulary segmentation. *arXiv preprint arXiv:2411.15869*, 2024.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice). In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 84298–84328. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/996bef37d8a638f37bdfcac2789e835d-Paper-Conference.pdf.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html, 2023.

Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. *arXiv preprint arXiv:2312.00878*, 2023.

Nick Cammarata, Shan Carter, Gabriel Goh, Chris Olah, Michael Petrov, Ludwig Schubert, Chelsea Voss, Ben Egan, and Swee Kiat Lim. Thread: Circuits. *Distill*, 2020. doi: 10.23915/distill.00024. https://distill.pub/2020/circuits.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting the second-order effects of neurons in clip. *ArXiv*, abs/2406.04341, 2024a. URL https://api.semanticscholar.org/CorpusID:270285780.

Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting clip's image representation via text-based decomposition, 2024b. URL https://arxiv.org/abs/2310.05916.

Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459.

Alec Helbling, Tuna Han Salih Meral, Ben Hoover, Pinar Yanardag, and Duen Horng Chau. Conceptattention: Diffusion transformers learn highly interpretable features. *arXiv preprint arXiv:2502.04320*, 2025.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. *CoRR*, abs/2201.11114, 2022. URL https://arxiv.org/abs/2201.11114.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. URL https://api.semanticscholar.org/CorpusID:231879586.

Nick Jiang, Amil Dravid, Alexei Efros, and Yossi Gandelsman. Vision transformers don't need trained registers, 2025. URL https://arxiv.org/abs/2506.08010.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, June 2013.

Victor Lecomte, Kushal Thaman, Rylan Schaeffer, Naomi Bashkansky, Trevor Chow, and Sanmi Koyejo. What causes polysemanticity? an alternative origin story of mixed selectivity from incidental causes, 2024. URL https://arxiv.org/abs/2312.03096.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. On the biology of a large language model. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/attribution-graphs/biology.html.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

Tuomas Oikarinen and Tsui-Wei Weng. Linear explanations for individual neurons. *arXiv preprint arXiv:2405.06855*, 2024.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024. URL https://arxiv.org/abs/2311.03658.

Yagyensh C. Pati, Ramin Rezaiifar, and Perinkulam S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44 vol.1, 1993. URL https://api.semanticscholar.org/CorpusID:16513805.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763. PMLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. URL https://arxiv.org/abs/2102.12092.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Tong Shao, Zhuotao Tian, Hang Zhao, and Jingyong Su. Explore the potential of clip for training-free open vocabulary semantic segmentation. In *European Conference on Computer Vision*, pages 139–156. Springer, 2024.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXI*, page 315–332, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-72663-7. doi: 10.1007/978-3-031-72664-4_18. URL https://doi.org/10.1007/978-3-031-72664-4_18.

Mitchell Wortsman. Reaching 80% zero-shot accuracy with openclip: Vit-g/14 trained on laion-2b. https://laion.ai/blog/giant-openclip/, 2023. Accessed: 2025-08-12.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024. URL https://arxiv.org/abs/2309.17453.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. URL https://arxiv.org/abs/2210.01936.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

Jitian Zhao, Chenghui Li, Frederic Sala, and Karl Rohe. Quantifying structure in clip embeddings: A statistical framework for concept interpretation, 2025. URL https://arxiv.org/abs/2506.13831.

Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European conference on computer vision*, pages 696–712. Springer, 2022.

Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2639–2650, 2023. doi: 10.1109/ICCV51070.2023.00249.

PC(s)	Accuracy (%)
(Baseline)	73.9
$\{\hat{r}^{n,h}\}$	73.9
$\{\hat{r}^{n}_1\}$	71.0
$\{\hat{r}^{n}_1, \hat{r}^{n}_2\}$	72.8
$\{\hat{r}^{n}_1, \hat{r}^{n}_2, \hat{r}^{n}_3\}$	73.3

Table 5: **Accuracy for reconstruction from principal components (RN50x64).** It remains true that neuron-head pair contributions are rank-1 in the embedding space while the representations of individual neurons are not.

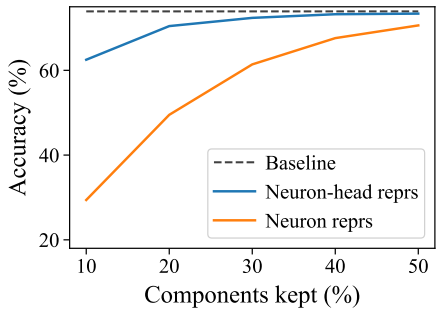

Figure 8: **Mean-ablation accuracy (RN50x64).** The same trend as Figure 1 is shown, in which there is a sparse set of neuron-head pairs that contribute very significantly to the output value.

A Appendix

A.1 Experimental results for RN50x64

We repeat experiments from Section 4.1 using OpenAI's CLIP-RN50x64 instead of RN50x16. As shown in Table 5 and Figure 8, neuron-head pairs and neurons display the same behavior described in the main text. Notably, the neuron representations of RN50x64 reconstruct with higher fidelity while using the same number of principal components as their RN50x16 counterparts. We also note that previous work shows that CLIP-ViT neurons are rank-1 in embedding space (Gandelsman et al. [2024a]). However, for CLIP-ResNet, it is still the case that only neuron-head pairs can be approximated by a single direction in the joint embedding space.

A.2 Reconstruction details

Specifically, we take $\hat{r}^{n,h}$ to be the first principal component computed from the top 50 $r^{n,h}(I)$ samples of $\mathcal{D}$ by norm. We compute $\{\hat{r}^{n}_1, \hat{r}^{n}_2, \ldots\}$ analogously. Additionally, we note that we add the positional embedding in this process.

In Figure 2 and Table 2, rather than decomposing the $\hat{r}^{n}_1$ we obtain from $\mathcal{D}$, we decompose the first principle component obtained from the *top 100* neuron contribution samples by norm selected from *5000* images from ImageNet test.

A.3 Finding sub-concept neuron-head pairs.

We observe that our text-based decomposition doesn't capture fine-grained sub-concepts (e.g., there is no 'clothing'-like description in the decomposition of the 'butterfly clothing' pair $\#(624, 21)$), and that these sub-concepts are difficult to identify via cosine similarity; for instance, $\langle \hat{r}^{624,21}, M_{\text{text}}(\text{"butterfly"}) \rangle >> \langle \hat{r}^{624,21}, M_{\text{text}}(\text{"butterfly clothing"}) \rangle$.

Instead, we manually inspect six different neurons that are randomly selected. We narrow the search space to a pool of thirty neurons whose directions $\hat{r}^{n}_1$ have cosine similarity to any text embedding from the top 30k English words above a certain threshold. We automate this by prompting ChatGPT to compare top images for a pair (n, h) and the top images for neuron n and to output sub-concept decisions (if the sub-concept relationship exists) and descriptions of the sub-concept. We then inspect ~ 5 of 48 attention head pairs based on the output descriptions and present findings in Figure 10.

A.4 Related work: semantic segmentation using CLIP

Many recent works use the inherent semantic alignment of CLIP's image and text encoders for dense segmentation tasks, in which CLIP features are either processed as pseudo-labels to train another network or are used themselves for segmentation predictions (Zhou et al. [2022]). We focus on the latter training-free paradigm, where state-of-the-art methods incorporate a *self-self attention* mechanism (Bousselham et al. [2023], Wang et al. [2024], Shao et al. [2024], Bai et al.

[2024]). In contrast to self-self attention methods that discard the class token and rewrite attention, our proposed segmentation method uses the (decomposed) class token and does not rearrange any internal mechanisms.

In comparison to CLIP-ViT, CLIP-ResNet has disadvantages for segmentation purposes: it has more aggressive spatial downsampling (a factor of 32 compared to 16) and, as shown earlier, is not conducive to self-self attention. However, CLIP-ResNet is uniquely able to process arbitrarily-sized images without resizing, can be better at processing larger images, and its attention pooling structure makes our findings relevant to CLIP-ViT.

A.5 Semantic segmentation using neuron-attention decomposition

Effect of varying k. We observe that cosine similarity to text decreases drastically as k increases, and the segmentation maps become noisier and more similar in range as k increases. We base this observation from experiments conducted for $k \in \{1, 5, 10, 100, 500, 1000, 5000, 10000, 20000, 25000\}$. Intuitively, this finding should mean that smaller k performs better on segmentation, but this is not the case: $k = 100$ gives only 20.0 mIoU while $k = 20000$ gives 26.2. Future work on thresholding techniques will likely bring improvement.

Effect of register neurons. Modifying select *register neurons* at test time was recently proposed (Jiang et al. [2025]) as an alternative to trained registers (Darcet et al. [2023]) that mitigate irregular attention patterns. We find a sparse set of register neurons for CLIP-ResNet and adopt the intervention detailed by Jiang et al. [2025] – this intervention improves mIoU by 0.36% for our method using $k = 100$, but the advantage becomes negligible or even detrimental with higher k. We note that our method of selecting CLIP-ResNet's register neurons hinges on the empirical observation, discussed in the next section, that CLIP-ResNet's attention sink invariably appears in the last image token – a better register neuron selection method may show further improvements.

A.6 CLIP-ResNet attention sink

We find that CLIP-ResNet's attention sink (Xiao et al. [2024]) appears in the last token, such that the class token attends extremely significantly to it, as shown in Figure 9.

Finding register neurons. Unlike previous works on DINO and CLIP-ViT, we do not notice outlier features in the input tokens that signal an attention sink. However, we find that CLIP-ResNet's attention sink always appears in the last token. Therefore, over 1000 images from ImageNet test, we sort the neurons by the absolute value difference in the attention sink caused by intervening on each neuron (zeroing its activation). We find a sparse set of register neurons (< 18) that causally affect the magnitude of the attention sink.

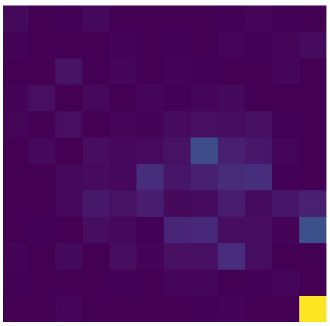

Figure 9: **An example of CLIP-ResNet's attention sink.** We show the class token attention weights (i.e., the first row of the attention matrix), averaged over each head for an arbitary input.

A.7 Compute

All experiments were run on a single A100 GPU. We note that the high amount of neuron-head pairs (for CLIP-RN50x16, $48 \cdot 3072 \approx 176000$) forces us to be memory-conscious (for instance, in using only 1000 images for $\mathcal{D}$).

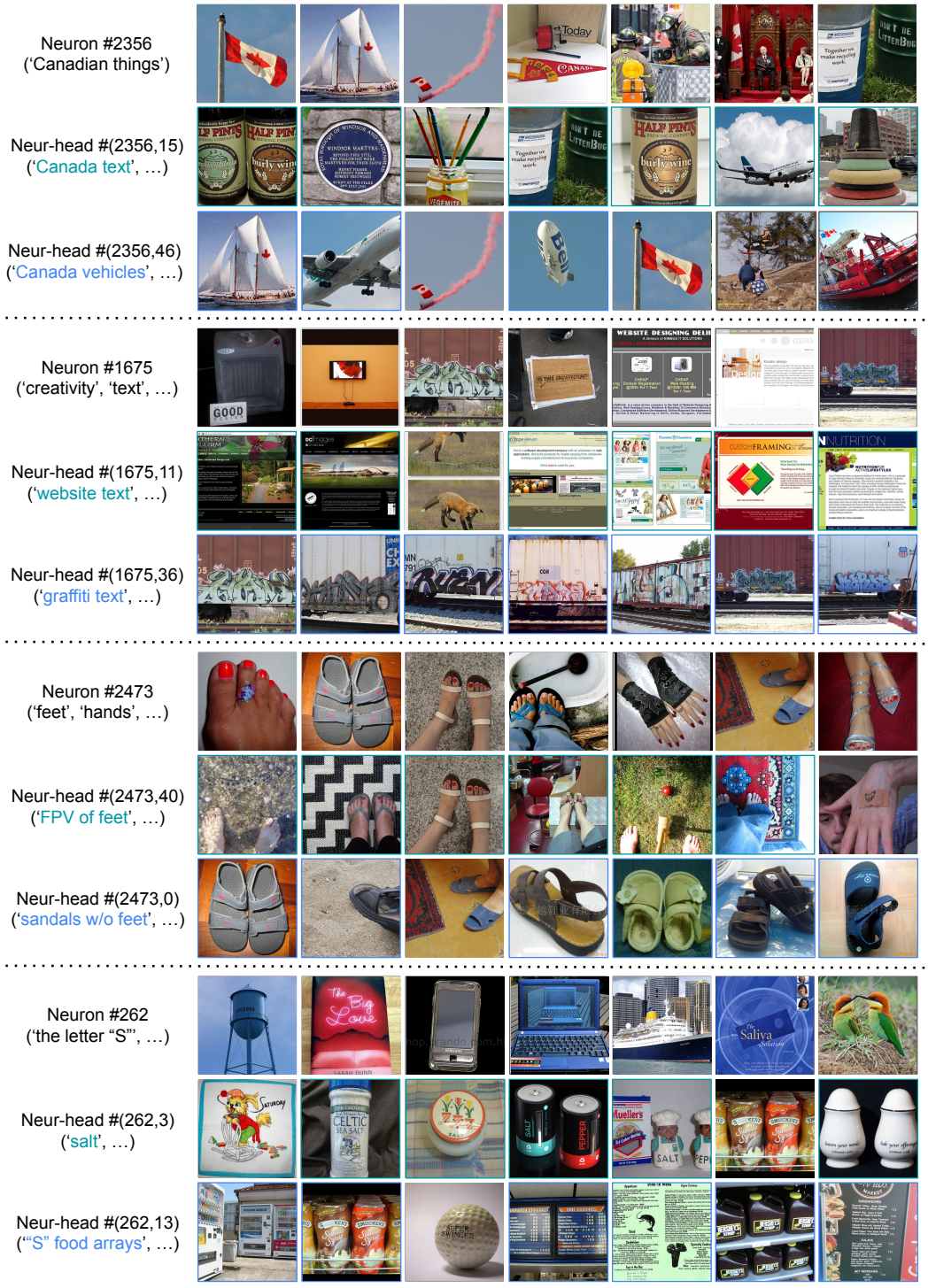

Figure 10: **Remaining sub-concept examples.**