

PURE: Turning Polysemantic Neurons Into Pure Features by Identifying Relevant Circuits

— Supplementary Material —

Maximilian Dreyer¹, Erblina Purelku¹, Johanna Vielhaben¹,
Wojciech Samek^{1,2,3,†}, Sebastian Lapuschkin^{1,†}

¹ Fraunhofer Heinrich Hertz Institute, ² Technical University of Berlin,

³ BIFOLD – Berlin Institute for the Foundations of Learning and Data

[†]corresponding authors: {wojciech.samek | sebastian.lapuschkin}@hhi.fraunhofer.de

A. Appendix

In the appendix, we provide additional results for the main manuscript. Concretely, we give details on how we compute feature visualizations in Appendix A.1. Secondly, in Appendix A.2, the distribution of polysemanticity throughout neurons of ResNet [3] models is shown in higher detail. Further, we provide additional examples of resulting disentangled representations when applying Purifying Representations (PURE) in Appendix A.3. Lastly, Appendix A.4 provides more results for evaluating feature interpretability after purification of neurons.

A.1. Feature Visualizations

An important part in understanding neurons are feature visualizations that aim to communicate the underlying semantics or concept of a neuron. In literature, either real data samples, or synthetically generated samples are used for visualization purposes [5]. Throughout our experiments, we utilize reference samples from the original dataset to render visualizations that are as natural as possible, ideally staying in-distribution w.r.t. to the foundational models of CLIP [8] and DINO [7] in evaluations.

When using reference samples from the dataset, it is crucial to crop images to the actually important part for a neuron, as semantics can be very localized, as, *e.g.*, visible in Fig. A.1 when comparing “full” against “cropped” samples. In order to detect this “relevant” part, Achibat *et al.* [1] propose to explain neuron activations using LRP [2] in a first step, which results in neuron-specific input feature attributions. Specifically for convolutional layers, the maximum activation of a feature map is explained. In a second step, the attributions are smoothed using a Gaussian filter with kernel size K , normalized to a maximum value of one, and the image cropped and masked to include only attributions above a threshold of T . For the masked part, black color is

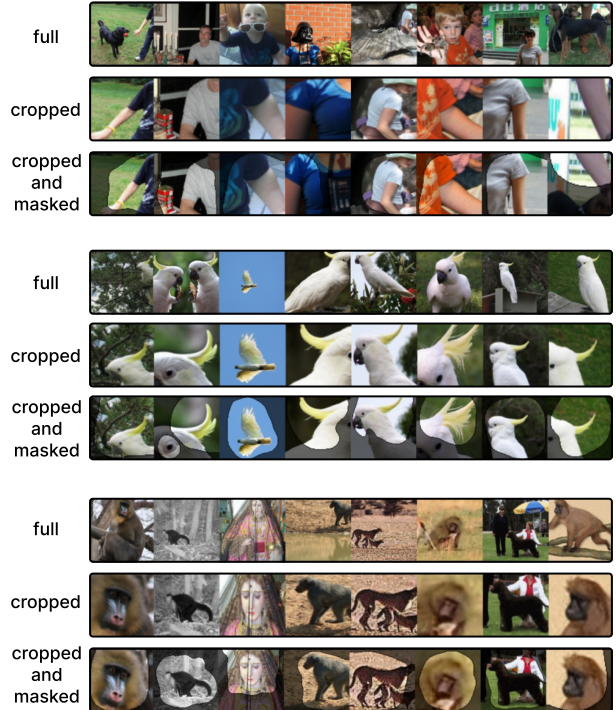


Figure A.1. Feature visualizations for neurons #1028 (*top*), #1029 (*middle*) and #1030 (*bottom*) for full, cropped-only and cropped as well as masked reference samples. It is visible that cropping improves visualizations by removing irrelevant and distracting image parts not relevant for a neuron semantics.

overlaid with 40 % transparency.

When evaluating visualizations with CLIP and DINO, we use cropped samples with $K = 5$ and $T = 0.01$. We refrain from masking samples, as masks could introduce out-of-distribution data. For visualizations in plots, we include masks as proposed by [1] using $K = 51$ and $T = 0.01$.

A.2. Distribution of Polysemanticity

In Sec. 4.1, we measure the degree of monosemanticity by using CLIP embeddings and computing distances in embedding space for feature visualization pairs of a neuron. Small distances between embeddings presumably correspond to visually similar feature visualizations.

For polysemantic neurons, where, *e.g.*, two monosemantic features superimpose, we would see two distinct clusters for each set of reference samples (of each pure feature) in a UMAP [4] embedding. The inter-cluster distance, in this case, will be high, whereas intra-cluster distance will be low. In the following, we apply k -means clustering ($k = 2$) on CLIP embeddings for all neurons w.r.t. the 100 most activating reference samples, and measure the overall as well as inter- and intra-cluster distances as given by Eq. (5).

In Fig. A.2, we show the distribution of distances (intra- and inter-cluster distance difference, and overall distance) for ResNet-50. It is apparent, that for most neurons, clustering improves the visual similarity of feature visualizations according to CLIP. However, for some (a few hundred of the 2048 neurons), a larger improvement can be seen, indicating more polysemantic neurons.

Examples with UMAP embeddings and clustered reference samples for neurons of varying degrees of polysemanticity are shown in Fig. A.2 (bottom). Neurons #1028 and #1984 have low overall CLIP embedding distance and correspond to monosemantic features, *e.g.*, “human arms” and “human crowds”, respectively. Further, neurons #696 and #107 have large improvement in embedding distance when clustered, indicating polysemanticity, *e.g.*, “dog face” vs. “text/horizontal lines” and “shark under water”, respectively. Lastly, we show neuron #1177 with high overall distances, where clustering does not strongly decrease distances. This is apparently due to three existing semantics in the neuron, where clustering using two clusters is not optimal for disentanglement.

Additionally, we provide distribution plots for ResNet-34 and ResNet-101 in Fig. A.3. It is to note, that ResNet-34 only consists of 512 instead of 2048 neurons in the penultimate layer. Comparing ResNet-34 and ResNet-100 distributions, improvements in CLIP embedding distances through clusters seem to be lower for ResNet-34, potentially indicating a smaller degree of polysemanticity. However, we leave comparison across architectures for future work.

A.3. Examples for Applying PURE

In the following, we present additional examples when applying PURE to neurons with different separability levels, which, for neurons with high separability score, leads to multiple virtual (ideally more disentangled) neurons.

As in Sec. 4.1, we create two virtual neurons using k -means for each neuron of a ResNet-50 in the penultimate layer. We then visualize the UMAP embeddings of the

PURE attributions given by Eq. (3) and the resulting clustered feature visualizations.

In Fig. A.4 we show five *randomly sampled* neurons which exhibit different levels of separability; we see that both monosemantic (#162 and #1804) and polysemantic (#141, #1657, and #310) neurons can be found. Regarding the polysemantic units #1657 and #310, PURE leads to well separated reference samples. For #141, three semantics seem to exist, where clustering with two clusters does not optimally disentangle the unit.

In Fig. A.5, we focus on neurons with higher degree of polysemanticity (indicated by a large inter-cluster distance and low intra-cluster distance on PURE embeddings), which can be meaningfully disentangled into multiple virtual monosemantic neurons using PURE. For instance, *e.g.* neuron #1381 encodes both for “printed letters” and “orca”, which PURE effectively disentangled. Similarly, neurons #916 and #915 refers to semantics such as “two dogs” and “bird wings” and of “badger” and “ketchup and mustard in hot dog” features, respectively.

Fig. A.6 illustrates examples of neurons encoding for pure features with small differences in inter- and intra-cluster distances (with a low level of separability). Noteworthy, instances include neurons such as #1, which represents “chain”, or neuron #160, encoding “seashore/shore”.

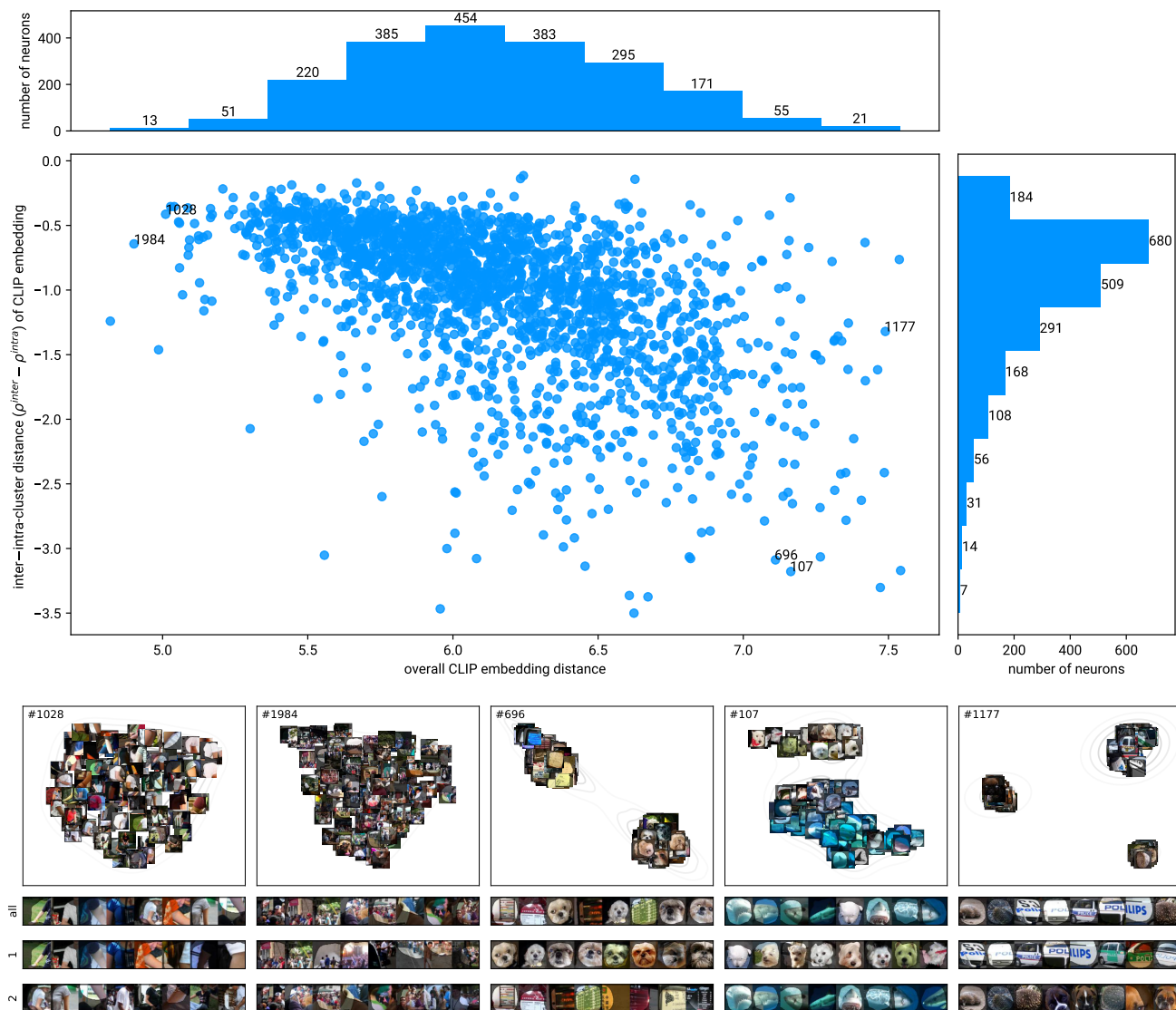


Figure A.2. Distribution of CLIP embeddings distances for all neurons of the ResNet-50 model. We show the overall distances between feature visualizations of a neuron on the horizontal axis, and the difference between inter- and intra-cluster distance after clustering visualizations into two clusters on the vertical axis. (*Bottom*): Examples are given for neurons with various degrees of polysemanticity. UMAP embeddings for PURE attributions as well as reference samples for the original and two virtual neurons are shown when applying PURE.

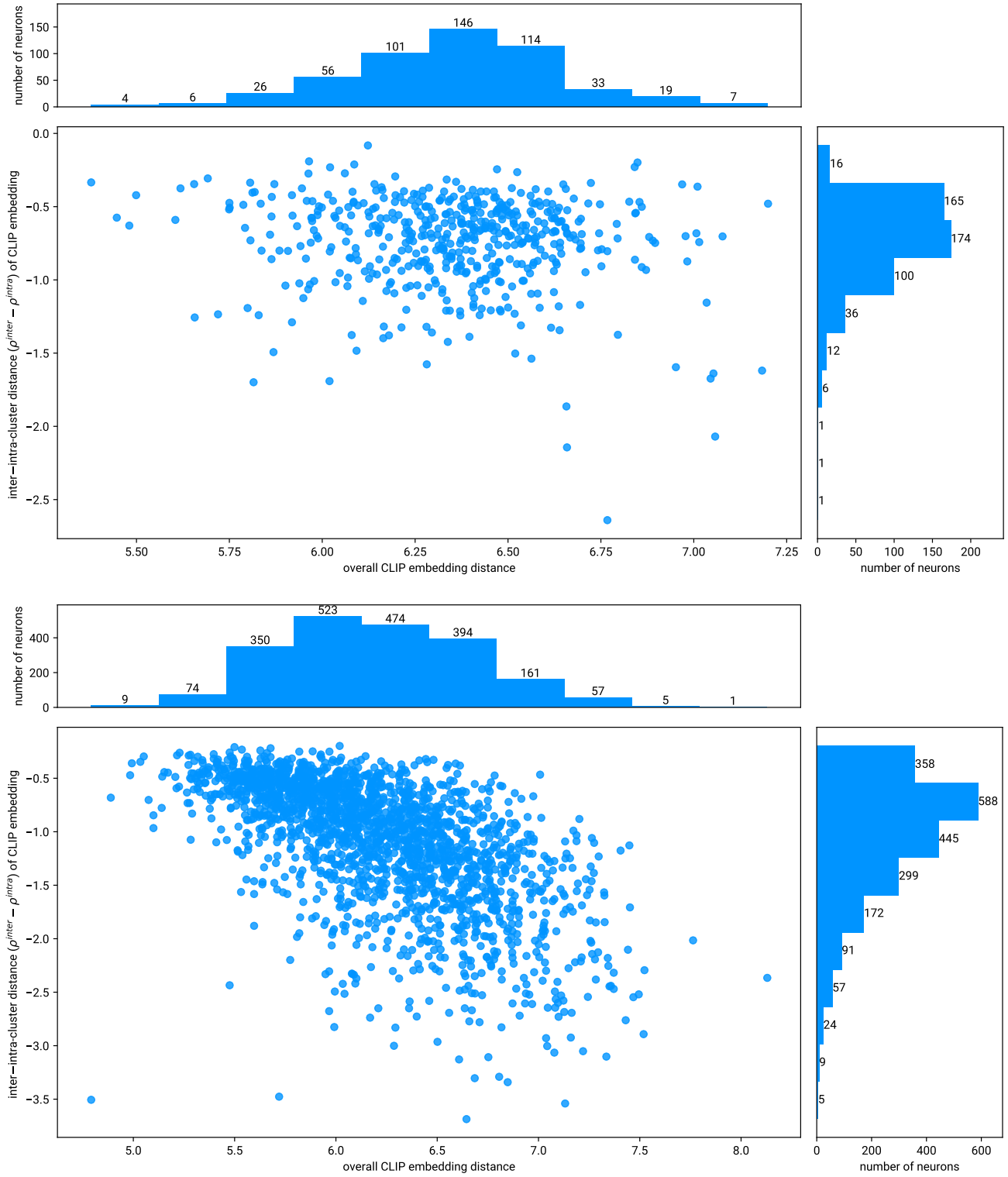


Figure A.3. Distribution of CLIP embeddings distances for all neurons of the ResNet-34 (*top*) and ResNet-101 (*bottom*) model. We show the overall distances between feature visualizations of a neuron on the horizontal axis, and the difference between inter- and intra-cluster distance after clustering visualizations into two clusters on the vertical axis.

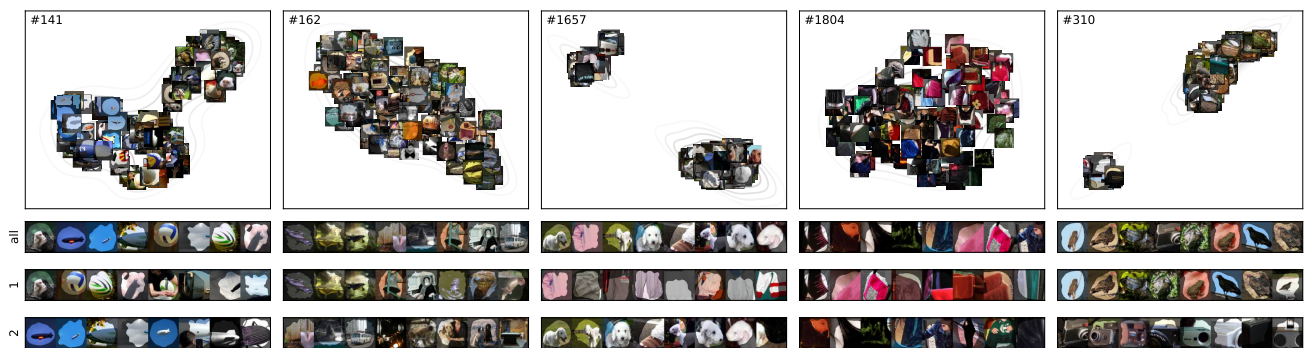


Figure A.4. Examples of applying PURE to *randomly chosen* neurons. Here we see the UMAP embeddings of the maximally activating patches, and the resulting reference sets before and after applying purification when identifying two circuits via k -means. In ResNet-50 neurons with different level of polysemanticity can be found.

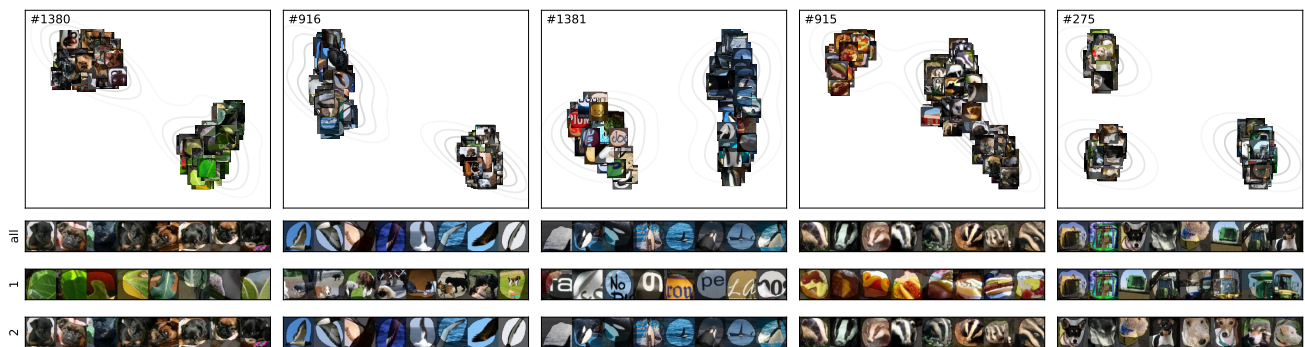


Figure A.5. Examples of applying PURE to neurons with *high degree of polysemanticity*. Here we see the UMAP embeddings of the maximally activating patches, and the resulting reference sets before and after applying purification when identifying two circuits via k -means. We show that neurons with high degree of polysemanticity can be successfully disentangled into two (or more) monosemantic neurons.

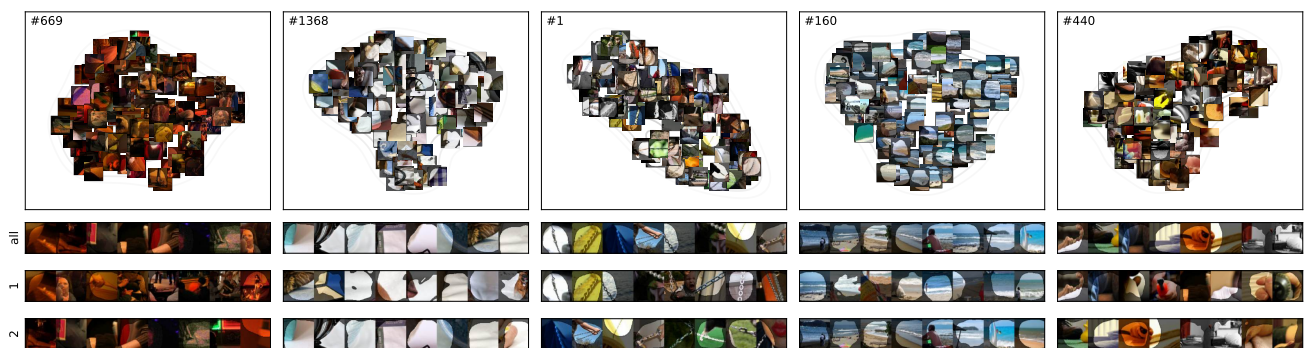


Figure A.6. Examples applying PURE to neurons with *low degree of polysemanticity*. Here we see the UMAP embeddings of the maximally activating patches, and the resulting reference sets before and after applying purification when identifying two circuits via k -means. The neurons encode one or similar features in this case, resulting in one circuit to be identified.

A.4. Evaluating Neuron Purification using CLIP

In Sec. 4.2, we evaluate the effectiveness of disentanglement via CLIP. Specifically, we create k new virtual neurons for each neuron by clustering the 100 most activating reference samples of a neuron into k clusters. Then, the reference samples of each cluster are evaluated using CLIP, where ideally, the CLIP embedding distance decreases inside clusters and increases across clusters, indicating well (visually) separated clusters. In Fig. A.8, we present additional results for the main manuscript (where $k = 2$ and ResNet-101 model results are shown) for ResNet-34 and ResNet-50 for $k \in \{2, 3, 4, 5\}$. For all experiments, PURE leads to better cluster separation than activation-based clustering. Notably, the higher k is, the lower intra-cluster distances are in general, indicating visually more monosemantic feature visualizations per virtual neuron. However, inter-cluster distance often decreases when k is increased, which indicates that most neurons are rather monosemantic, as also discussed in Appendix A.2.

A clustering that is aligned with CLIP results from similar distances between feature visualization pairs according to the respective methods. Concretely, CLIP and DINOv2 embeddings $\mathbf{e}_i^{\text{CLIP}}$ (and $\mathbf{e}_i^{\text{DINOv2}}$) are computed using feature visualizations (cropped reference samples) as the input for reference sample i w.r.t. to a neuron p in layer L . For PURE distances are computed on lower-level layer attributions \mathbf{R}^{L-1} as given by Eq. (3) when explaining neuron p . Further, for activation-based distances, activations \mathbf{A}^L in layer L are used, as proposed by [6]. The more aligned to CLIP, the higher the correlation of distances between the methods is. We provide additional results for Sec. 4.2 with results for ResNet-50 and ResNet-34 in Fig. A.7, for which PURE also shows higher correlations than activation. The correlation analysis involves examining the pairwise distances for the top-50 most activating reference samples across all neurons in the penultimate layer. The standard error of mean is computed by partitioning distances in 30 subsets (over which the mean is computed).

A.4.1 When Disentanglement Diverges from CLIP

In the following, we present examples, when activation-based or PURE attribution-based clustering of reference samples diverges from how CLIP embeddings cluster feature visualizations for the ResNet-50.

Activation We observe unfaithful clustering with activations when a significant portion of feature visualizations is of a single class, leading to high activation similarities (due to similar features present in the reference samples). The reference samples of the same class dominate in clustering, leading to all samples from different classes being clustered in another cluster. This is given, e.g., for neurons #143

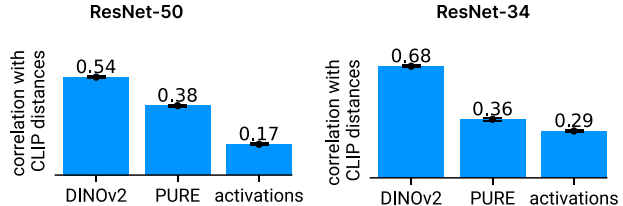


Figure A.7. Correlation between feature visualization distances of CLIP to other methods for ResNet-50 (left) and ResNet-34 (right), which extends the results given in Fig. 4.

(class “custard apple”), #385 (class “shoji”), #614 (class “lacewing”), or #1055 (class “buckeye”). Further, all reference samples can have small activation similarity if, e.g., most are from different classes, which is the case for neuron #1147, leading to noisy clustering.

PURE Similarly, as for activation, attributions can result in different clustering (compared to CLIP), e.g., for neurons #614 or #1055 as shown in Fig. A.10, where very small distances between reference samples from the same class (“lacewing” and “buckeye”, respectively) lead to unaligned clustering. For other neurons, semantics can be more abstract and difficult to understand, such as #1032 (radially outspreading lines) or #1121. In these cases, CLIP seems to result in clustering reference samples corresponding to same depicted object classes (e.g., dogs or monkeys, respectively). This is also the case for neuron #271, where the semantics seems to correspond to triangular shapes that can be found for toy windmills or dog ears. Here, CLIP separates dogs from windmills, even though they correspond to the same semantics, indicating also a disadvantage of using CLIP for evaluation. Especially for abstract concepts, attributions can result in low distances, whereas according to CLIP, bigger distances exist. This also raises the question whether the ultimate goal of disentanglement is to separate clusters according to *visual* difference or *semantic* difference (as seen by the model).

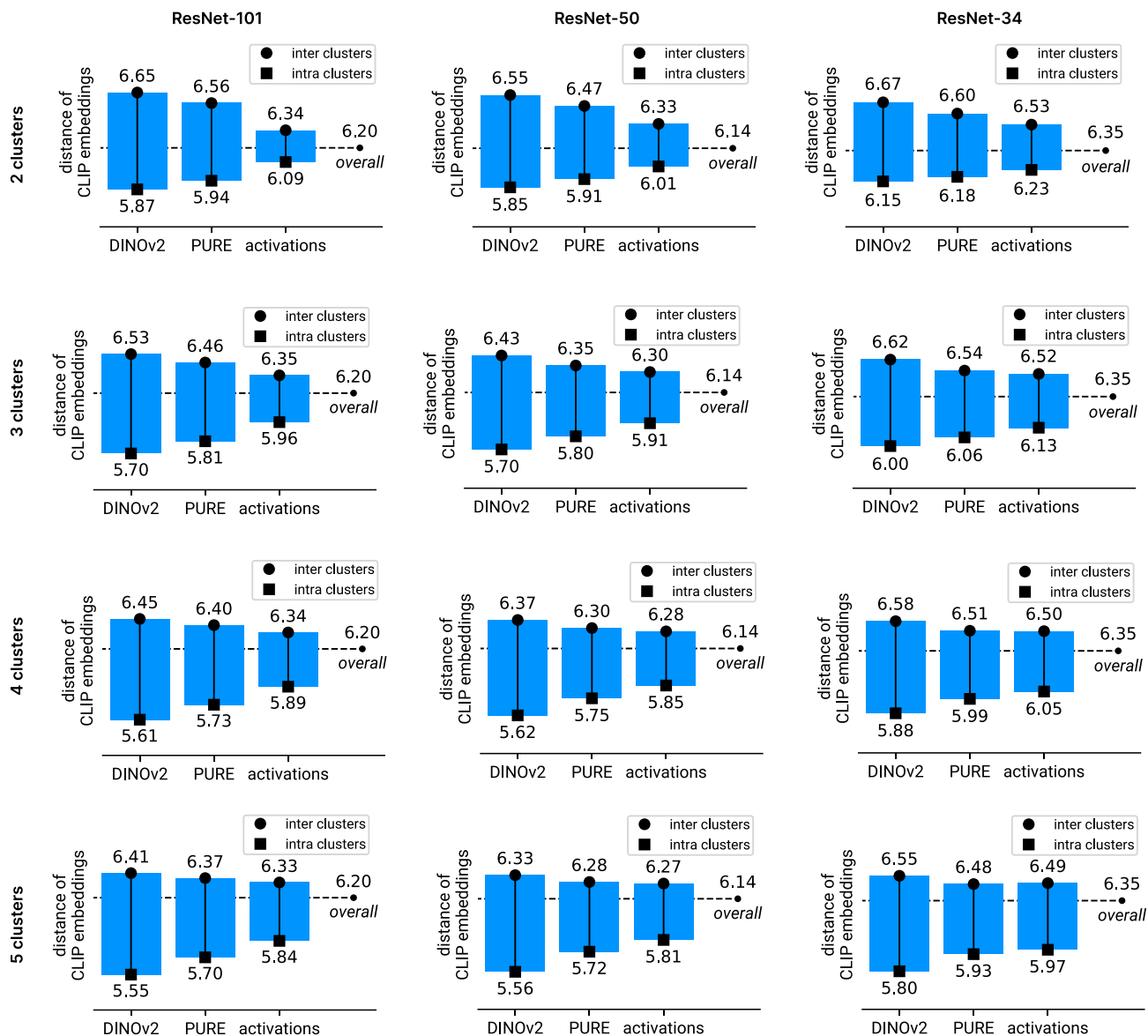


Figure A.8. Results for measuring the degree of monosemanticity of clustered feature visualizations using CLIP embedding distances, as discussed in Sec. 4.2 for all ResNet architectures and different number of clusters (virtual neurons).



Figure A.9. Examples for diverging disentanglement using activations compared to CLIP embeddings for the ResNet-50 model. We show UMAP embeddings and the corresponding feature visualizations before (“all”) and after (cluster “1” and “2”) disentanglement using activations (*top*) and CLIP embeddings (*bottom*)



Figure A.10. Examples for diverging disentanglement using PURE attributions compared to CLIP embeddings for the ResNet-50 model. We show UMAP embeddings and the corresponding feature visualizations before (“all”) and after (cluster “1” and “2”) disentanglement using attributions (*top*) and CLIP embeddings (*bottom*)

References

- [1] Reduan Achtabat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance propagation. *Nature Machine Intelligence*, 5(9):1006–1019, 2023. [1](#)
- [2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140, 2015. [1](#)
- [3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [1](#)
- [4] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018. [2](#)
- [5] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. [1](#)
- [6] Laura O’Mahony, Vincent Andrearczyk, Henning Müller, and Mara Graziani. Disentangling neuron representations with concept vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3770–3775, 2023. [6](#)
- [7] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. [1](#)
- [8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [1](#)