5 APPENDIX

5.1 QUANTITATIVE ANALYSIS

We performed 783 experiments each using 20 samples uniformly drawn from two classes (not only the high-confidence samples) and achieved 783 new sub-networks by NeuroChains on VGG-19. We evaluated these newly generated sub-networks using the quotient metric "A quotient of "diff to highest scoring other class (extracted)" / "diff to highest scoring other class (original)" Eq. (9). We visualized the result in Figure 8: The left plot is the histogram of the quotient computed over all the 783×20 samples. The histogram shows that most samples keep the original predicted label after pruning, i.e., NeuroChains can preserve the original DNN's outputs in most cases. Moreover, the number of filters preserved in these sub-networks is $157(mean) \pm 43(std)$, which is small enough to explain. The right plot reports the Faithfulness of NeuroChains in terms of the quotient's sign. We remove each filter from each sub-network and report how many samples' predicted labels are changed after the removal, i.e., the quotient is negative. Each point in the scatter plot corresponds to a sub-network, the x-axis is the score of the removed filter given by NeuroChains, and the y-axis is the proportion of samples with negative quotients. The plot shows a strong linear correlation between the score of the removed filter and the degradation of faithfulness. Since removing filters with high scores results in more samples with predicted class changing after pruning, the score given by NeuroChains measures the importance of filters in DNN inference.

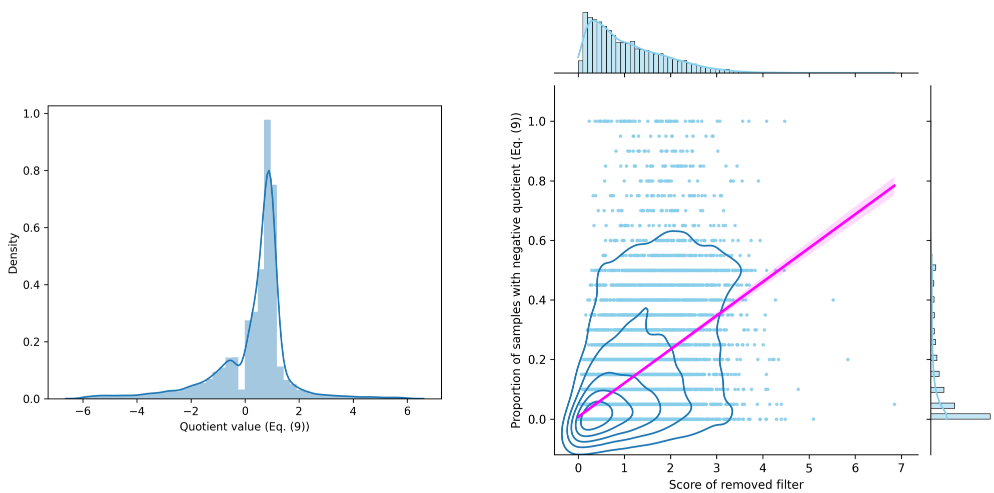

Figure 8: Histogram of the quotient metric in Eq. (9) computed over all the 783×20 samples (LEFT). Faithfulness of NeuroChains in terms of the quotient's sign (RIGHT).

We evaluate the stability of NeuroChains using the nearest neighbours from the penultimate-layer representation space. Because ReLU pattern does not provide an ideal metric to measure the distance of samples, even in the raw input space: (1) the number of ReLU linearity zones grows exponentially with the number of hidden nodes. Most ReLU linearity zones are empty and do not contain any real sample; (2) For the few ReLU linearity zones that do contain samples, each only contains one sample and by large chance its neighboring linearity zones are empty, and this is true for most practical cases as empirical studies suggested. So it is almost impossible to find two samples sharing the same ReLU linearity zone or even close in their ReLU patterns of the first layer; (3) For two ReLU linearity zones that are only different in one facet of their polyhedra (i.e., only one digit of their ReLU patterns flips), their corresponding linear models can still be very different (the linear model is an extreme case of sub-network). Therefore, we speculate that samples close to each other in terms of their ReLU patterns do not share a sufficiently small sub-network preserving their original predictions.

That being said, we evaluated each NeuroChains extracted sub-network of VGG-19 using 20 samples randomly drawn from two classes on the K-th nearest neighbour (NN) of each sample by sorting the Hamming distance on their ReLU patterns. The K-NN samples' prediction cannot be well pre-

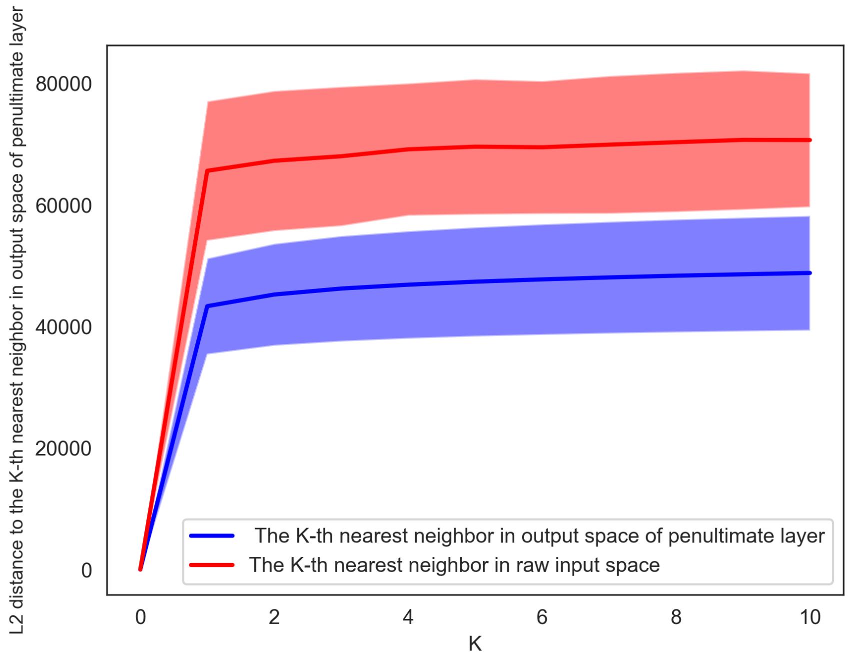

Figure 9: Mean$\pm$std of L2 distance in the the penultimate-layer representation space between each sample and its K-nearest neighbours from the penultimate-layer representation space (blue) and ReLU pattern space (red).

served on the sub-networks, because the nearest neighbors in terms of ReLU patterns have very different semantic concepts or classes from the samples that the sub-networks are extracted for. Hence, the local region of ReLU patterns is not a local region on the smooth data manifold. To see this, in Figure 9, for each sample, we computed its L2 distance to the ReLU pattern K-NN sample's penultimate-layer representation for $K = 1, 2, \cdots, 10$ (the red curve reports mean$\pm$std), and we compared them with the L2 distance to the K-NN in the penultimate-layer representation space (the blue curve reports mean$\pm$std). It shows that the ReLU pattern K-NN has a much larger L2 distance in the semantic space (i.e., penultimate-layer representation), so it is very different in concepts to the original sample. Moreover, we show some examples of the ReLU pattern K-NN images and the penultimate-layer K-NN images for the sample in Figure 10, which show that ReLU pattern K-NN images are much less related to the original sample.

In Figure 11, we show two case studies of comparing SMOE generated heatmaps for the original network and the NeuroChains extracted sub-network. We can see that the patterns extracted by the two networks are consistent and are all critical patterns for the class, e.g., the eyes and fists of kangaroos and the feet and face of the horse. However, compared with the original network, these patterns are strengthened in much shallower layers of the sub-network, producing better interpretations. This observation is also consistent with the result of analysis on adversarial attacks in Figure 5.

In Figure 12, we compare the capability of preserving the original neural network's outputs between NeuroChains and magnitude-based pruning (removing the filters whose output featuremaps' average magnitude (L2 norm) over all considered samples is small). In particular, under the same setting of each experiment in the paper, we prune the original VGG-19 and retain the filters with the largest featuremap magnitude in each layer, 180 in total (more than $157(mean) \pm 43(std)$ filters for sub-networks extracted by NeuroChains), and we then fine-tune the filters' scores/weights as we did for NeuroChains. Figure 12 shows the histogram of the KL divergence between the original output class distribution and the one produced by the sub-networks. For sub-networks generated by NeuroChain, the KL-divergence in most cases stays close to 0, while the output preserving capability of simple pruning is much worse.

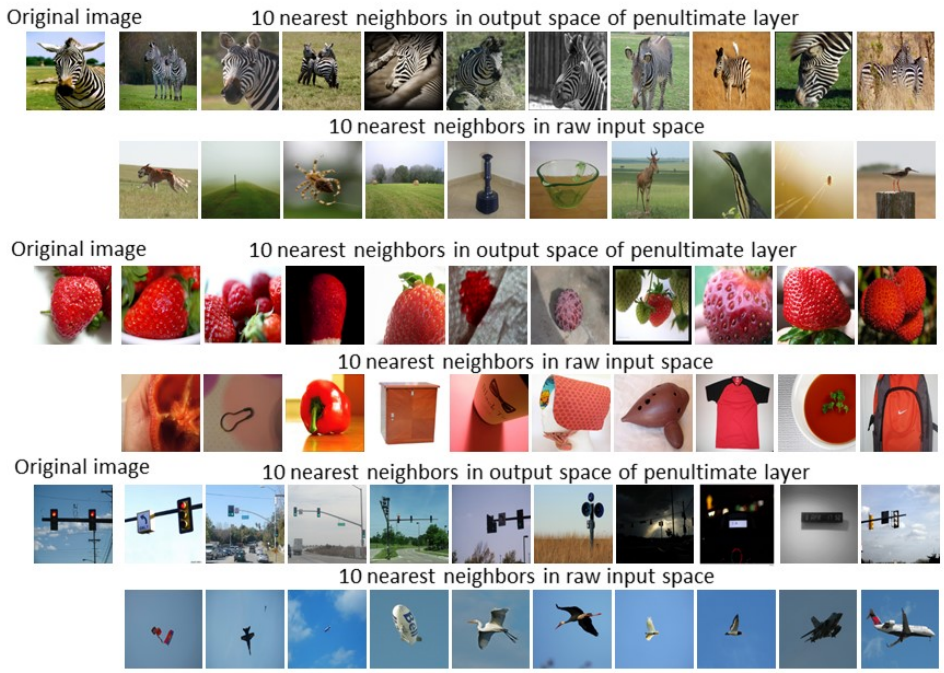

Figure 10: Case studies of an image, its 10-nearest neighbours in the output space of penultimate layer (Top), and its 10-nearest neighbours in the raw input space in terms of Hamming distance between first-layer ReLU patterns (Bottom).

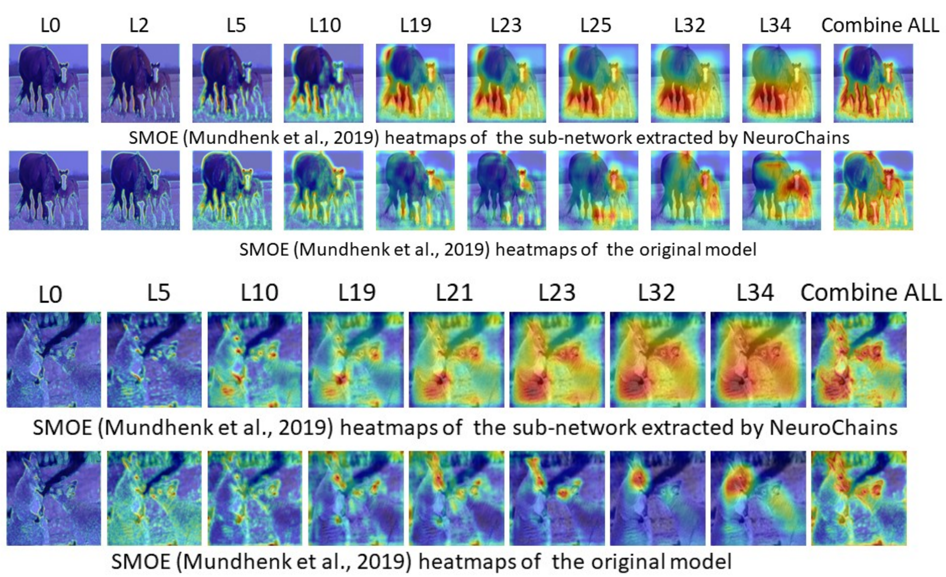

Figure 11: Case studies of SMOE generated heatmaps for the original network and the NeuroChains extracted sub-network.

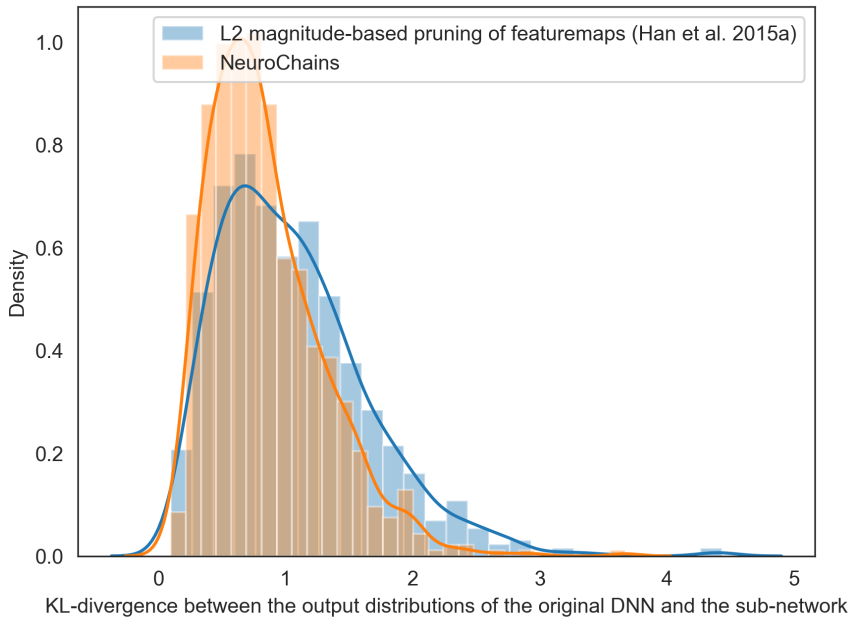

Figure 12: Comparison of NeuroChains and magnitude-based pruning on the capability of preserving the original network's output distribution (smaller KL divergence means better preservation) over 783×20 uniform samples.

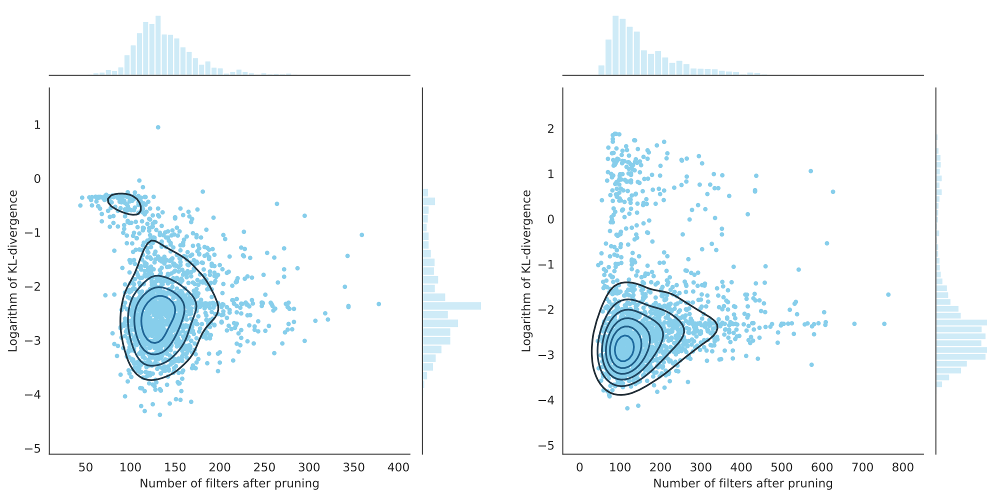

Figure 13: Statistics of the output discrepancy between sub-networks extracted by NeuroChains and the original network: VGG-19 (Left) and ResNet-50 (Right). The x-axis refers to the number of filters in sub-DNNS, the y-axis is the logrithm of KL-divergence between the output distributions produced by the sub-networks and the original network. The KL-divergence for most samples are small, indicating the sub-networks preserve the original network's output distribution for most samples.

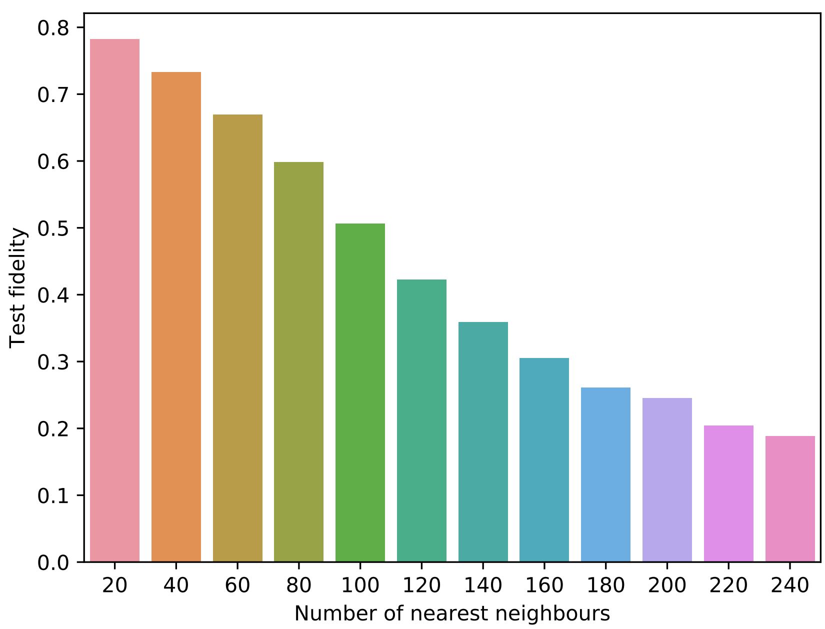

Figure 14: Histogram of stability of sub-networks extracted by NeuroChains for ResNet-50. The x-axis refers to 20K for the K-nearest neighbours of the 20 samples used to extract the sub-network) in the penultimate-layer representation space, while the y-axis is the test fidelity (averaged over all sub-networks), i.e., the accuracy of sub-networks in preserving the predicted class by the original network on the unseem K-nearest neighbours.

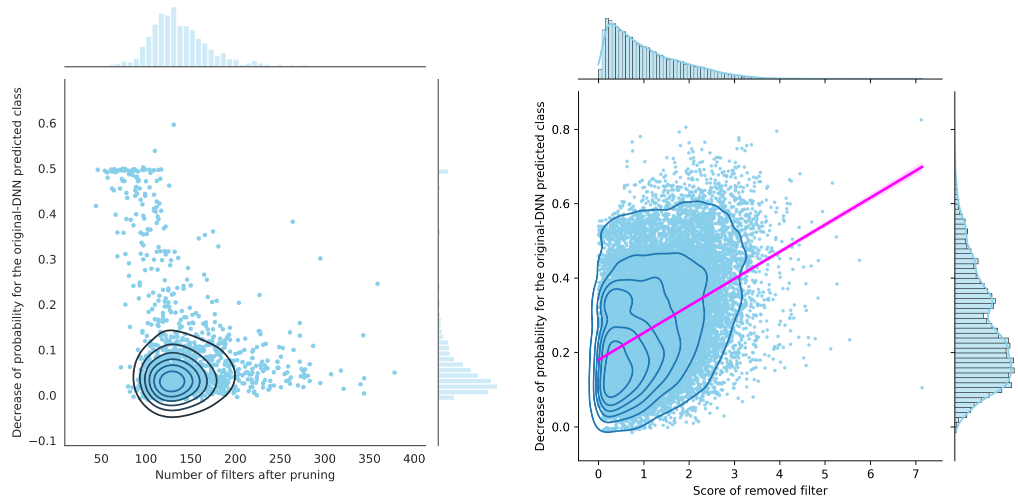

Figure 15: **Left:** Scatterplot with a jointly density estimate of the performance of sub-networks extracted by NeuroChains for VGG-19. Each point corresponds to a sample. The x-axis refers to the number of filters in the sub-network, the y-axis measures the decrease of probability on the original network predicted class. For VGG-19, most sub-networks' output probabilities drop very little regardless of how many filters are retained. **Right:** Scatterplot with a jointly density estimate of faithfulness of sub-DNNs extracted by NeuroChains for VGG-19. The x-axis refers to the scaling score of removed filter, the y-axis is the decrease of average probability for the original-DNN predicted class compared with the complete sub-DNNs. For VGG-19, it seems the higher the score, the more the probability drops. The slope of the magenta line is the linear (Pearson) correlation, while the shaded area around the line represents the confidence interval.

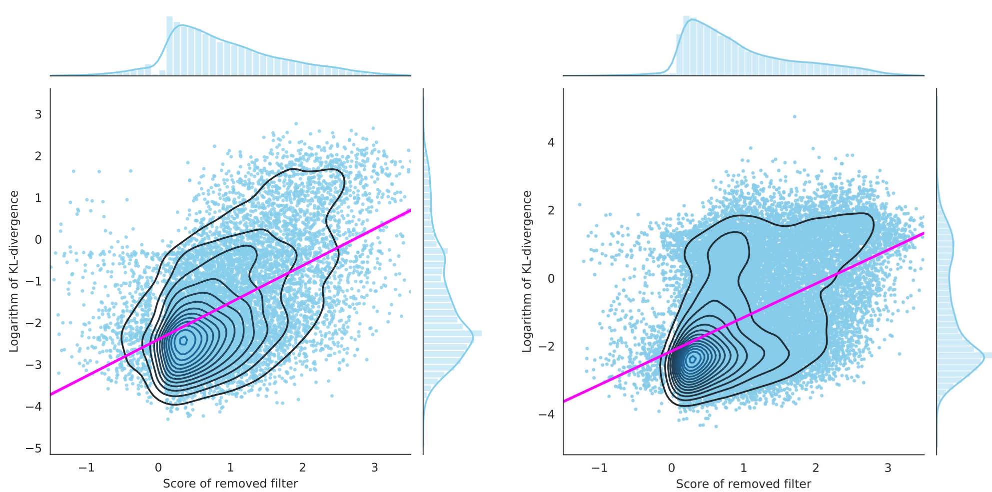

Figure 16: Scatterplot with a jointly density estimate of faithfulness of sub-networks extracted by NeuroChains for VGG-19 (Left) and ResNet50 (Right). The x-axis refers to the scaling score (weight) of removed filter in the sub-networks, the y-axis is the logrithm of KL-divergence between the outputs of the new sub-networks (after removal) and the original sub-networks (before removal). For both VGG-19 and Resnet-50, it shows that the higher the score, the higher the KL-divergence.

5.2 More Details about Case Studies

On the sub-network's architecture, we use "L0" to denote the corresponding convolution layer in VGG-19 and "L0_1" to denote the first filter from this layer. For ResNet-50, we further use "L1B1" to denote the first sub-block in the first bottleneck block, "SC" for the shortcut connection and "C1" for the first convolution layer in the sub-block. The redder the node in the sub-network, the larger the scaling score, conversely, the bluer the node, the lower the score. More case studies can be found in the Appendix. In SMOE, Mundhenk et al. propose to measure information at the end of every feature scale and then combined them into a saliency map. We apply this technique to each layer of the sub-network since each layer may prefer different features. In each of our case study, a featuremap-overlaid input image is shown for each layer and for the whole sub-network which is marked as "Combine All". The visualization of each selected filter is achieved by maximizing its activation w.r.t. the input. Afterward, we shows the patterns that the filter aims to detect which is independent of the input image.

5.3 Case Studies

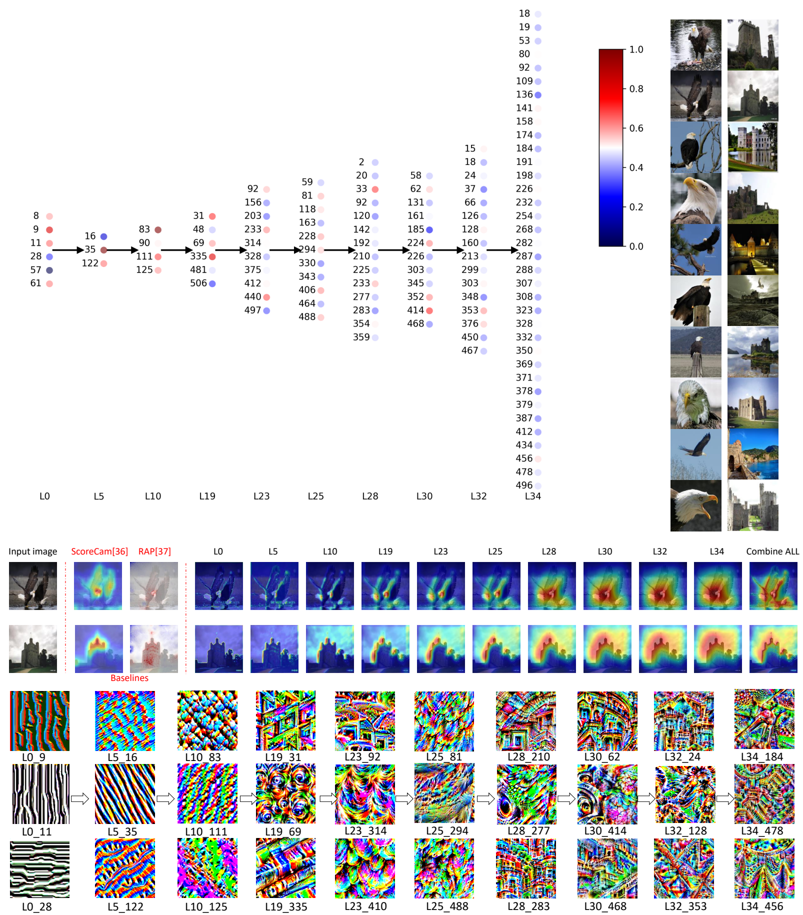

Figure 17: Inference chain by NeuroChains for VGG-19 when applied to images of "bald eagle" and "castle". **Top:** The sub-network retains only 10/16 layers and 118/4480 filters. **Middle:** The per-layer featuremaps generated by SMOE. Since there is nothing similar between bald eagles and castles, it's easy for VGG-19 to tell them apart. Different types of feathers are an important feature of eagle and the contour of the castle is highlighted. **Bottom:** Filters with the largest scores. In shallower layers, L23_314 and L19_69 capture the patterns of feathers and eyes of eagle, which are different from other species. L28_277 in the deeper layer combines the above two patterns. L25_81 identify the half circle of feathers around neck to be key pattern of eagle. L32_24 and L34_184 can be explained as detectors of the whole head and neck of eagle that combines all the patterns detected in previous layers. L23_92 shows the pattern of small room with windows. L28_210, L28_283, L30_62 and L32_24 extract clear patterns of castle. It shows an inference chain for eagle: L10_83 → L19_69, L23_314 and L25_294 → L28_277 → L30_414 → L32_128 → L34_184. It shows an inference chain for castle: L10_125 → L19_31 → L23_92 → L28_210 and L28_283 → L30_62 and L30_468 → L32_24 → L34_478.

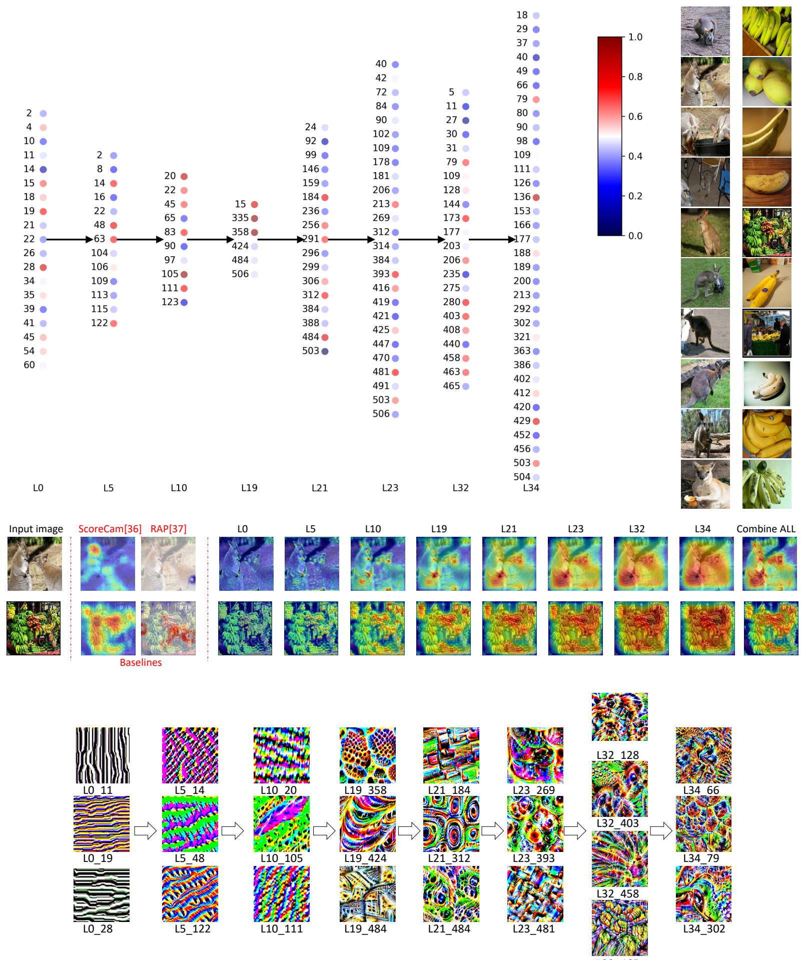

Figure 18: Inference chain by NeuroChains for VGG-19 when applied to images of "kangaroo" and "banana". **Top:** The sub-network retains only 8/16 layers and 148/4480 filters. **Middle:** The per-layer featuremaps generated by SMOE. The black ball-shape pattern exists both in kangaroos and bananas. The hands and eyes of kangaroos are highlighted, and the ends of bananas are lit up. These parts are all black and round in images. **Bottom:** Filters with the largest scores. L21_312, L23_269, L23_393 are all related to the black round pattern. L21_312 shows the basic black round pattern. L23_269 looks like the eyes and noses of animals while in L23_393 these nodes are closely arranged like a hand of bananas. To better distinguish this two class, VGG-19 introduces some key patterns for each class. L32_465 and L34_79 depict the whole image of hands of bananas. L34_66 combine the previous features and show the pattern of animal faces. It shows an inference chain for kangaroo: L10_105 $\rightarrow$ L19_358 $\rightarrow$ L21_312 and L21_484 $\rightarrow$ L23_269 $\rightarrow$ L32_403 $\rightarrow$ L34_66.

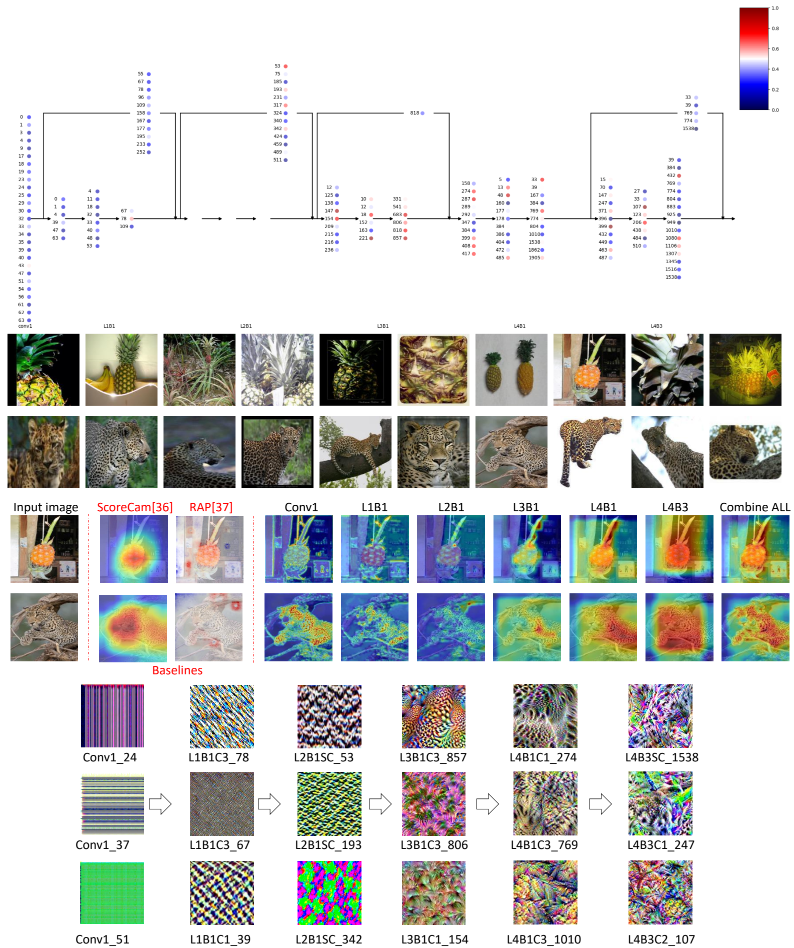

Figure 19: Inference chain by NeuroChains for Resnet-50 when applied to images of "pineapple" and "leopard". **Top:** The sub-network retains only 17/67 layers and 157/26560 filters. **Middle:** The per-layer featuremaps generated by SMOE. Both the body and leaves of the pineapple are highlighted. The special skin texture is enough for ResNet-50 to identify leopard. **Bottom:** Filters with the largest scores. By observing the patterns in the activation maximization result and the highlighted regions in the featuremap, we can find that some filters extract different local patterns appearing at different parts of pineapple. For example, L4B1C3_1010 capture the texture and the color of the main body, L3B1C1_154 capture the patterns of the leaf part. It is interesting to see that L4B3C2_107 is the accurate descriptor of the main body and the leaf parts and thus provide nearly orthogonal features. For leopard, the skin marked with black spots is its most obvious feature. L4B1C1_274 extracts the basic texture and color while L4B3SC_1538 and L4B3C1_247 really show the skin pattern of the leopard. It shows an inference chain for pineapple: L2B1SC_342 → L3B1C3_806 and L3B1C1_154 → L4B1C3_1010 → L4B3C2_107.

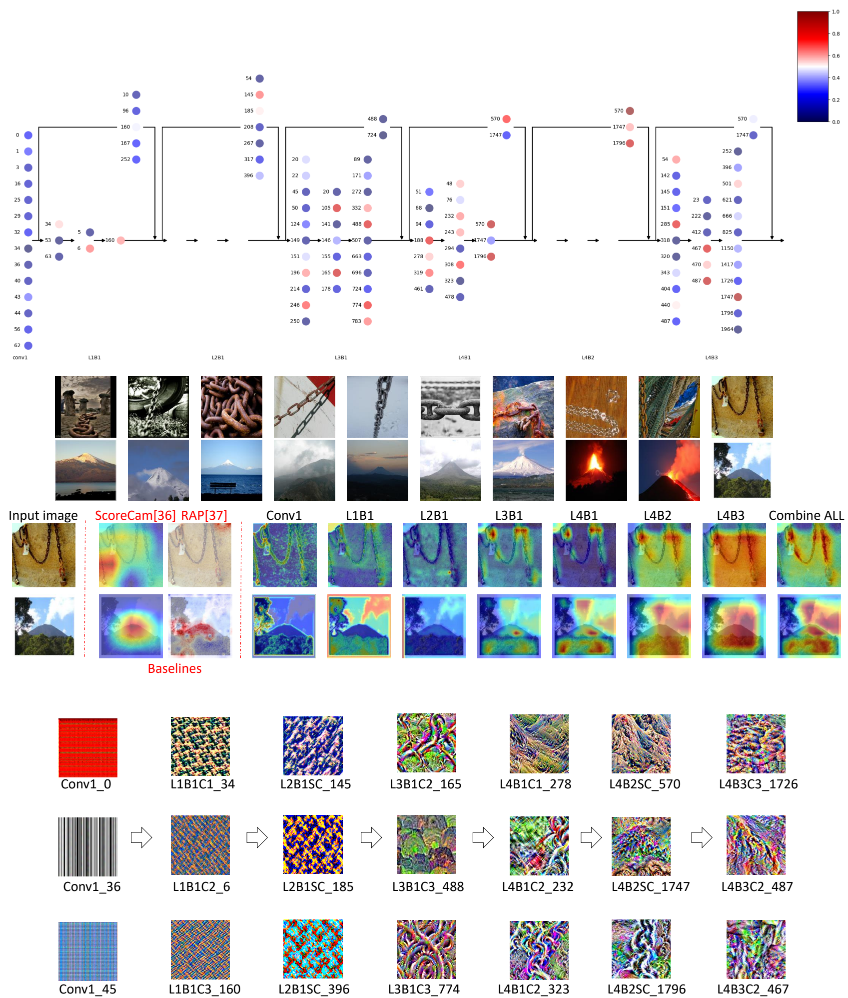

Figure 20: Inference chain by NeuroChains for ResNet-50 when applied to images of "chain" and "volcano". **Top:** The sub-network retains only 18/67 layers and 114/26560 filters. **Middle:** In the SMOE featuremaps, not only the main body of the volcano, but the crater is also highlighted as key features to identify volcanos. **Bottom:** In the first several layers, L3B1C2_165, L3B1C3_774 and L3B1C3_488 extract basic patterns such as curved steel bars and the arc of mountains, whilst deeper layers focus on more global patterns such as different orientations of the folded strata (L4B1C1_278 and L4B2SC_570) and chains (L4B2SC_1796 and L4B3C2_467). L4B3C2_487 captures the features when lava erupts from volcanos as in the penultimate image. It reveals an inference chain for volcano: L2B1SC_145 → L3B1C3_488 → L4B1C1_278 → L4B2SC_570 → L4B3C2_487.