Supplementary Material

1. Information of the models used in our experiments

Model	VGG19	ResNet50	ResNet50-C1	ResNet50-C2	ResNet50-D
Params	144M	25M	25M	25M	25M
Top-1 acc	74.5	76.1	79.8	80.0	79.8
Model	**ConvNeXt-T**	**Swin-T**	**Nest-T**	**DeiT-S**	**PiT-S**
Params	25M	28M	28M	22M	23M
Top-1 acc	82.1	81.2	81.5	79.9	80.9
Model	**DeiT-S-distilled**	**PiT-S-distilled**	**LeViT-256**		
Params	22M	23M	19M		
Top-1 acc	81.2	81.9	81.6		

Table 1. Information of the models used in our experiments along with the number of learnable parameters and the Top-1 accuracy on ImageNet-1K.

Model	ResNet50-A2	Swin-T	DeiT-S	DeiT-S-distilled	LeViT-256
Params	25M	28M	22M	22M	19M
Top-1 acc	79.73 ± 0.15	81.09 ± 0.05	79.72 ± 0.07	80.94 ± 0.16	78.76 ± 0.04

Table 2. The number of learnable parameters and the Top-1 accuracy on ImageNet-1K of the models we trained. The left side of the symbol ± is mean value and the right side is standard deviation

Model	ConvNeXt-T	ConvNeXt-T-3	ConvNeXt-T-3-GN	ConvNeXt-T-3-BN
Top-1 acc	82.1	81.3	82.0	80.8
Model	**Swin-T**	**Swin-T-4**	**Swin-T-4-GN**	**Swin-T-4-BN**
Top-1 acc	81.2	81.1	81.1	80.6

Table 3. The number of the Top-1 accuracy on ImageNet-1K of the models we trained and the original models.

In Table 1 we give some information of the models used in our experiments. In Table 2 and 3 we gave some information about the models we trained ourselves. Note that most of the models have similar sizes to remove a potential confounding factor of the analysis. As the Windows of Swin Transformer must not overlap, during the training of the Swin-T with a small receptive field, we opted to modify the window size to 4 only for the first two stages instead of applying a uniform change to 3 for all stages.

2. More Results on Minimal Sufficient Explanations and Their Sub-Explanations

2.1. Complexity of the algorithms for computing subexplanations.

The runtimes (in seconds) for identifying all subexplanations for a single image with a single NVIDIA Tesla V100 GPU are 30 (ResNet50) and 765 (Swin-T), averaged over the first 5,000 images from the ImageNet validation set. Swin-T is much slower due to the huge amount of subexplanations. Note that this runtime pertains only to this analysis to uncover insights, and is not critical for any realistic inference task. There is no additional spatial complexity beyond running inference on the networks.

2.2. More Information on activation map values

Fig. 1 shows the activation map values on Swin Transformers (the main paper has it on ConvNeXt) and indeed the trend is clear that the top feature channels when using BN are much more dominant than with GN and LN as well.

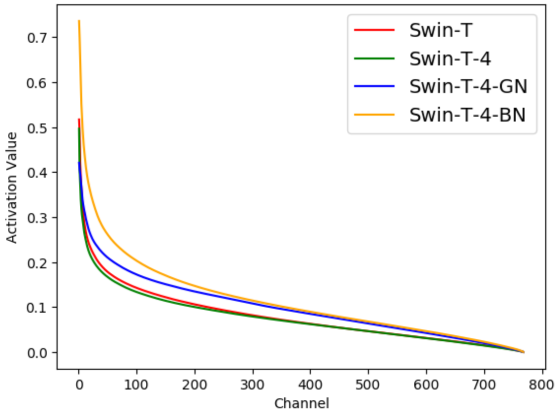


Figure 1. Sorted average values of the maximal activation in each image for each channel in the last block for Swin-T variants

2.3. Random Seeds Experiment for Establishing Statistical Significance

In Sec. 4.1, we show the count of MSEs and subexplanations among different networks. To check the statistical significance of our results, we used several representative models trained by ourselves with different random seeds to count the number of MSEs and subexplanations. The results are provided in the Table 4. These results confirm that the main results are statistically significant: 1) newer CNN model (ResNet50-A2) tends to have smaller MSE sizes. 2) Transformer model Swin-T and DeiT-S without distillation have significantly more subexplanations than CNNs and transformers with distillation; 3) Transformers after distillation have more MSEs with less subexplanations: DeiT-S-dis has significantly more and smaller MSEs than DeiT-S.

2.4. Effect of Perturbation Style

In Sec. 4.1, we set the perturbed pixels to a highly blurred version of the original image (hereafter referred to as the *Blur* perturbation style) for the Minimal Sufficient Explanations and Their Sub-Explanations experiments. This is a

Model		**MSEs**		**Number of Subexplanations**			
Type	**Name**	**Count**	**Size**	$\geq$ **80%**	$\geq$ **70%**	$\geq$ **60%**	$\geq$ **50%**
newer CNNs	ResNet50-A2	9.10 ± 0.33	5.99 ± 0.13	53.11 ± 26.93	84.60 ± 37.19	138.91 ± 44.95	213.82 ± 68.34
Transformers	Swin-T	8.51 ± 0.24	**8.20** ± 0.07	**224.57** ± 59.89	**840.20** ± 117.28	**2615.55** ± 214.73	**6349.22** ± 488.30
	DeiT-S	7.72 ± 1.25	8.01 ± 0.74	127.85 ± 36.73	486.02 ± 89.53	1542.36 ± 350.51	3730.73 ± 1256.73
Distillations	DeiT-S-dis	10.55 ± 0.57	5.72 ± 0.21	59.96 ± 26.74	129.87 ± 36.46	242.67 ± 41.90	431.75 ± 64.78
	LeViT-256	**12.47** ± 0.56	5.49 ± 0.16	50.60 ± 29.12	103.10 ± 37.72	164.53 ± 46.62	231.20 ± 41.67

Table 4. Beam search results to locate MSEs of seed experiments. Confidence represents average amount of nodes with a classification confidence higher than that threshold w.r.t. the confidence of the whole image. The results are shown in the form of mean ± standard deviation obtained with 3 different seeds. Statistical significance were affirmed with T-tests

Model		**Perturbation**	**MSEs**		**Number of Subexplanations**			
Type	**Name**		**Count**	**Size**	$\geq$ **80%**	$\geq$ **70%**	$\geq$ **60%**	$\geq$ **50%**
older CNNs	ResNet50	Blur	6.76	7.28	53.68	108.55	180.44	296.92
		Grey	6.62	**7.65**	21.47	59.76	126.07	233.92
ConvNeXt	ConvNeXt-T	Blur	10.28	6.14	**980.16**	**2001.67**	**3610.37**	5360.43
		Grey	9.38	6.98	81.37	235.08	463.05	769.40
Transformers	Swin-T	Blur	8.90	**8.01**	221.58	882.72	2933.03	**7268.20**
		Grey	10.03	6.99	**171.15**	**433.26**	**851.81**	**1373.90**
Distillations	LeViT-256	Blur	**12.59**	5.50	54.96	103.24	177.33	253.66
		Grey	**11.01**	6.33	42.42	96.45	168.22	234.57

Table 5. Results of beam search to locate MSEs and sub-explanations. Confidence represents average amount of nodes with a classification confidence higher than the respective threshold w.r.t. the classification confidence on the whole image

common approach in model explanation literature to alleviate the adversarialness of perturbations [2]. Here we use another method, setting the perturbed pixels to zeros (hereafter referred to as the *Grey* perturbation style), to obtain image perturbations. Noting that these could include additional edges to the image hence distort the predictions. Due to the limitations of the GPU resources, we select one model from each model type, plus one ConvNeXt-T.

Table 5 shows the count of MSEs and their sub-explanations for two different perturbation styles: *Grey* and *Blur*. It can be seen that some of the main conclusions of the paper still stand when using different types of perturbation: 1) LeViT- 256 (a distilled transformer model) still have a higher mean number of MSEs; 2) Transformers without distillations still have significantly more sub-explanations than other models.

However, in all models, compared to using the *Blur* perturbation style, the number of sub-explanations decreases significantly when the *Grey* perturbation style is used, which showed that the *Grey* style perturbation is more adversarial than the *Blur* perturbation. Besides, the size of Swin-T MSEs decreased significantly once switched to a *Grey* perturbation, which can be explained by that the *Grey* perturbation may have had a negative effect on the features of all categories (including other categories that might be confusing), hence decreasing the amount of patches that are needed for transformers to make a confident prediction.

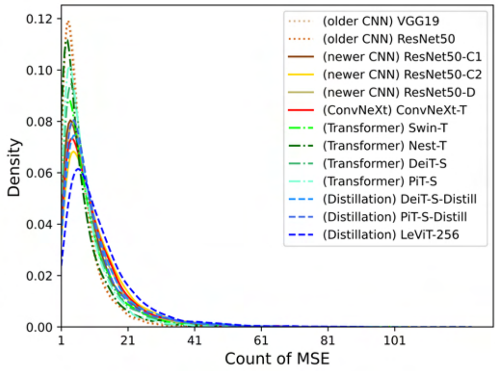


Figure 2. Distribution of MSE counts on ImageNet images

2.5. More Information on MSEs

In Fig. 2 and Fig. 3 we show the distribution of MSE counts and sizes in individual images. As one can see in Fig. 2, in many images, most methods have a small amount of MSEs and similar MSE sizes. However, the peak density is different among different approaches and the tail size is different among different appraoches. older CNNs (VGG19 and ResNet50) and transformers (Swin-T, Nest-T, Deit-S and PiT-S) have the highest density of images with low number of MSEs. In Fig. 3, we see that newer CNNs and distilled transformers can classify up to 20% images confidently with as little as 2 **patches**, whereas for (non-distilled) transformer models only about 10% of images are explained with only 2 patches. On the other hand, there are a significant number of images where non-distilled transformers have more patches in their MSEs, as compared to newer

CNN models and distilled transformers.

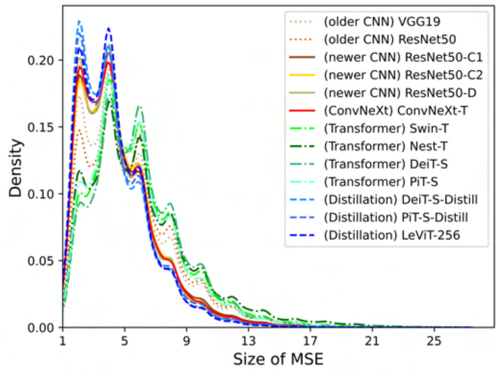


Figure 3. Distribution of MSE Size on ImageNet images

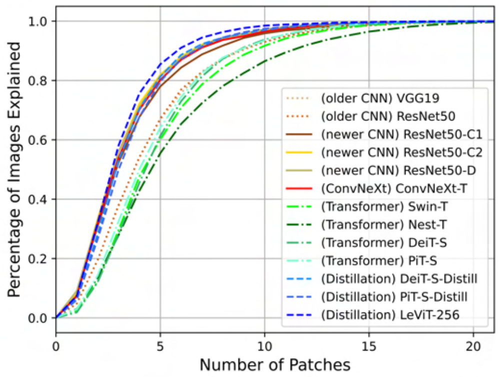


Figure 4. Percentage of images explained by different number of patches.

Fig.5 shows the distribution of MSE counts (with $\leq 50\%$ overlap) and sizes in a few random images. As one can see, in most images, distilled transformer models often have a significantly higher amount of MSEs. For example, DeiT-S-distilled has 50 MSEs in the `Water Tower` and `School bus` image, much higher than other models that have $3-19$ MSEs. Besides, Transformers (Swin-T and DeiT-S) have significantly higher MSE sizes in these two images than all other methods. These plots show a more complete picture of what can happen in each individual image.

We follow [5] to plot the percentage of images that can be explained with a small amount of patches. For each number of patches n, we plot the total proportion of images that contain at least one MSE with size $\leq n$. For Fig. 4, we use the 5,000 images from ImageNet validation dataset, we found that the transformers without distillation needed the most amount of patches to explain an image, also include older CNN models. And distilled transformers and newer CNN models in general can explain more images with a small amount of patches.

Finally, we show more visual examples of Structural Attention Graph (SAG) trees on several images with all the

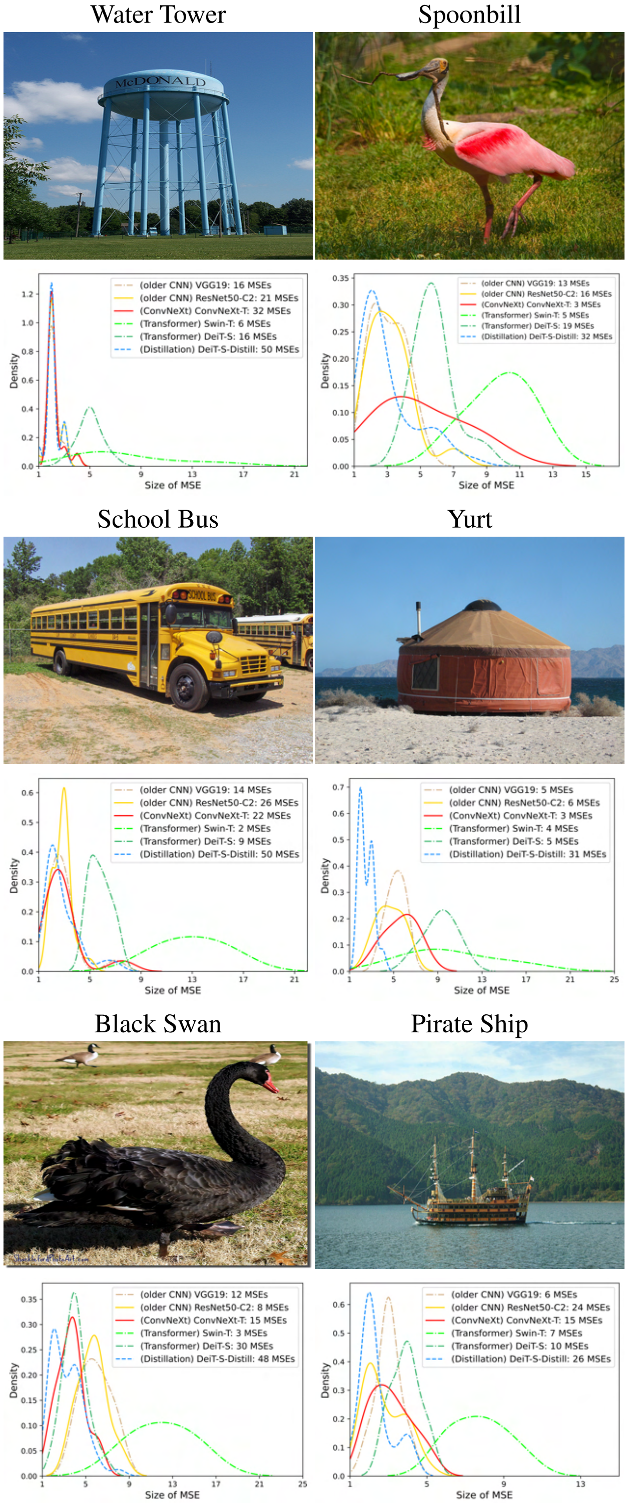


Figure 5. A few example distributions of MSE sizes for different networks on random images. Transformers often have larger MSEs than other networks, and in some images distilled Transformers have significantly more MSEs than other networks

Generation →	**VGG19**		**ResNet50**		**ConvNeXt-T**		**Swin-T**		**Nest-T**		**DeiT-S**		**DeiT-S-distill**		**LeViT-256**	
Evaluation ↓	Del.	Ins.	Del.	Ins.	Del.	Ins.	Del.	Ins.	Del.	Ins.	Del.	Ins.	Del.	Ins.	Del.	Ins.
VGG19	0.139	0.742	0.196	0.732	0.336	0.517	0.195	0.673	0.178	0.568	0.200	0.608	0.187	0.634	0.280	0.626
ResNet50	0.172	0.767	0.179	0.752	0.356	0.542	0.205	0.691	0.186	0.600	0.216	0.631	0.198	0.660	0.296	0.648
ConvNeXt-T	0.298	0.752	0.317	0.748	0.412	0.632	0.298	0.723	0.275	0.675	0.298	0.684	0.277	0.693	0.401	0.698
Swin-T	0.310	0.706	0.324	0.704	0.428	0.607	0.297	0.679	0.302	0.680	0.304	0.670	0.288	0.679	0.392	0.647
Nest-T	0.300	0.721	0.316	0.721	0.430	0.619	0.309	0.690	0.255	0.656	0.292	0.662	0.272	0.676	0.396	0.674
DeiT-S	0.261	0.758	0.284	0.749	0.404	0.602	0.277	0.713	0.258	0.657	0.244	0.664	0.246	0.689	0.365	0.665
DeiT-S-distill	0.272	0.815	0.298	0.813	0.458	0.658	0.288	0.763	0.268	0.699	0.281	0.708	0.235	0.726	0.406	0.743
LeViT-256	0.320	0.844	0.346	0.844	0.516	0.703	0.336	0.800	0.309	0.741	0.334	0.744	0.305	0.767	0.433	0.781

Table 6. Score-CAM Cross-Testing. Deletion/Insertion metrics when generating heatmaps using the model on the first row and evaluating the heatmaps by using the model on first column.

tested approaches (Fig. 9 - Fig. 29). Even if we only limited to showing 3 children per parent node, the size of the trees in Swin-T, DeiT-S and ConvNeXt-T are huge, showing strong evidences that the predictions of them are built by simultaneously taking into account the contributions of many different parts. On the other hand, DeiT-S-distill, VGG and ResNet have very small trees and can obtain very high confidences with very small amount of parts, which makes them to have very few sub-explanations. This shows the difference between compositional and non-compositional disjunctive models more intuitively.

3. More Results on Cross-Testing

3.1. More Results with Score-CAM as the Attribution Map

In addition to iGOS++, we also performed Cross-Testing using Score-CAM [6]. This is a visual explanation method based on class activation mapping. We chose this method because it obtains the weight of each activation map through its forward passing score on target class, not depending on gradients. Hence it is a different type of attribution map than the perturbation-based iGOS++. We also normalize the scores for each model [4], and use Kernel PCA to project them to 2 dimensions to better visualize their similarities [3].

Fig. 6 shows the projection results. We find similar trends as shown in the main paper: that older CNNs (VGG19 and ResNet50) are closer to each other, transformers (Nest-T, Swin-T and DeiT-s) are closer to each other, ConvNeXt-T is closer to transformers, and distillation (from a CNN) brings DeiT-S closer to older CNNs. These four points are consistent with the results we obtained in the main paper with iGOS++. The only difference is that LeViT-256 becomes an outlier. It becomes not very similar with the other distilled model: DeiT-S-distilled. We do want to note that from Table 6, we can see that the deletion score related to LeViT-256 is very high. For example, the deletion score of using LeViT-256 to evaluate attribution maps derived from the same algorithm is as high as 0.433, significantly higher than other methods, which indicates that this method may be less reliable on LeViT-256, in that it highlights some areas that are irrelevant to the prediction. In comparison, the deletion score for iGOS++ on LeViT-256 is only 0.117. In terms of insertion score, iGOS++ is $0.9-0.97$ for all networks if tested on itself, whereas Score-CAM usually averages only around $0.63-0.78$, showing that it significantly underperformed iGOS++ by $20\%-30\%$. Note that most attribution maps generate significantly worse deletion scores than I-GOS [1], the predecessor of iGOS++, and Score-CAM is already an excellent one of its kind, outperforming GradCAM and others based on a third-party benchmark [1].

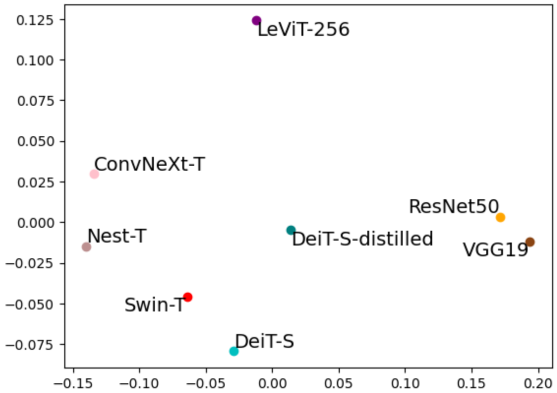


Figure 6. Kernel PCA projections of different models using the insertion metrics obtained with Score-CAM

3.2. More Results on iGOS++

In Sec. 4.2, we set the perturbed pixels to a highly blurred version of the original image for the cross-testing experiments using iGOS++. Here, we provide cross-testing results for the main models with a zero-image baseline using iGOS++ in Figure 7, which showed similar trends as results in the main paper.

In Section 4.2, we provide the the visualized results of

Generation → **Evaluation ↓**	**VGG19** Del.	 Ins.	**ResNet50** Del.	 Ins.	**ConvNeXt-T** Del.	 Ins.	**Swin-T** Del.	 Ins.	**DeiT-S** Del.	 Ins.	**DeiT-S-distill** Del.	 Ins.
VGG19	0.082	0.617	0.113	0.529	0.131	0.317	0.117	0.394	0.131	0.369	0.098	0.491
ResNet50	0.124	0.605	0.097	0.649	0.145	0.400	0.131	0.467	0.144	0.451	0.105	0.574
ConvNeXt-T	0.203	0.709	0.204	0.709	0.155	0.683	0.182	0.621	0.200	0.603	0.150	0.703
Swin-T	0.224	0.672	0.226	0.665	0.224	0.553	0.155	0.745	0.223	0.598	0.166	0.682
DeiT-S	0.219	0.694	0.217	0.687	0.220	0.563	0.195	0.651	0.138	0.762	0.133	0.774
DeiT-S-distill	0.219	0.716	0.221	0.715	0.224	0.584	0.199	0.657	0.193	0.677	0.094	0.831

Table 7. iGOS++ Cross-Testing with a zero-image baseline. Deletion/Insertion metrics when generating heatmaps using the model on the first row and evaluating the heatmaps by using the model on first column.

Generate → **Evaluate ↓**	**VGG19** Del	**ResNet50** Del	**ResNet50-C1** Del	**ResNet50-C2** Del	**ResNet50-D** Del	**ConvNeXt-T** Del	**ConvNeXt-T-3** Del	**ConvNeXt-T-3-GN** Del	**ConvNeXt-T-3-BN** Del	**Swin-T** Del	**Swin-T-4** Del	**Swin-T-4-GN** Del	**Swin-T-4-BN** Del	**Nest-T** Del	**DeiT-S** Del	**PiT-S** Del	**DeiT-S-distill** Del	**PiT-S-distill** Del	**LeViT-256** Del
VGG19	0.944	0.863	0.786	0.778	0.807	0.651	0.651	0.653	0.66	0.715	0.717	0.732	0.765	0.647	0.699	0.707	0.789	0.8	0.796
ResNet50	0.88	0.926	0.824	0.821	0.839	0.719	0.724	0.72	0.739	0.767	0.768	0.781	0.821	0.705	0.758	0.767	0.844	0.85	0.851
ResNet50-C1	0.855	0.851	0.911	0.838	0.852	0.745	0.752	0.741	0.774	0.77	0.771	0.784	0.821	0.717	0.763	0.769	0.833	0.847	0.84
ResNet50-C2	0.863	0.867	0.858	0.923	0.872	0.772	0.78	0.77	0.805	0.799	0.799	0.81	0.843	0.751	0.79	0.801	0.86	0.871	0.867
ResNet50-D	0.857	0.859	0.84	0.84	0.902	0.752	0.757	0.751	0.776	0.783	0.782	0.794	0.83	0.736	0.776	0.783	0.847	0.86	0.854
ConvNeXt-T	0.823	0.821	0.803	0.803	0.814	0.947	0.797	0.797	0.801	0.796	0.792	0.805	0.824	0.758	0.783	0.795	0.824	0.835	0.821
ConvNeXt-T-3	0.815	0.809	0.773	0.778	0.797	0.767	0.95	0.762	0.78	0.773	0.769	0.779	0.824	0.721	0.752	0.766	0.805	0.818	0.804
ConvNeXt-T-3-GN	0.829	0.822	0.809	0.819	0.832	0.808	0.807	0.97	0.81	0.815	0.814	0.823	0.839	0.776	0.811	0.82	0.848	0.861	0.843
ConvNeXt-T-3-BN	0.832	0.832	0.822	0.828	0.839	0.796	0.805	0.789	1.237	0.81	0.804	0.823	0.861	0.749	0.782	0.792	0.84	0.859	0.85
Swin-T	0.824	0.818	0.79	0.79	0.809	0.749	0.745	0.75	0.753	0.943	0.831	0.818	0.861	0.751	0.788	0.792	0.835	0.844	0.834
Swin-t-4	0.811	0.807	0.755	0.759	0.782	0.732	0.729	0.731	0.738	0.819	0.939	0.836	0.855	0.734	0.773	0.78	0.82	0.83	0.816
Swin-t-4-GN	0.804	0.804	0.767	0.765	0.787	0.747	0.742	0.75	0.757	0.82	0.828	0.954	0.857	0.749	0.79	0.794	0.831	0.841	0.825
Swin-t-4-BN	0.833	0.826	0.792	0.795	0.817	0.761	0.754	0.753	0.798	0.839	0.847	0.858	1.106	0.751	0.794	0.797	0.858	0.869	0.856
Nest-T	0.812	0.801	0.77	0.771	0.787	0.728	0.725	0.731	0.727	0.78	0.773	0.783	0.801	0.932	0.767	0.774	0.819	0.829	0.81
DeiT-S	0.853	0.854	0.831	0.828	0.843	0.789	0.784	0.796	0.794	0.848	0.852	0.865	0.887	0.798	1.008	0.867	0.946	0.919	0.896
PiT-S	0.86	0.86	0.829	0.832	0.852	0.797	0.793	0.802	0.798	0.844	0.844	0.858	0.876	0.805	0.867	0.972	0.906	0.935	0.896
DeiT-S-distill	0.852	0.873	0.851	0.85	0.867	0.798	0.795	0.801	0.81	0.85	0.852	0.863	0.897	0.801	0.89	0.862	0.995	0.928	0.911
PiT-S-distill	0.883	0.881	0.859	0.866	0.886	0.815	0.814	0.817	0.829	0.859	0.857	0.874	0.9	0.816	0.871	0.888	0.927	0.978	0.921
LeViT-256	0.879	0.879	0.852	0.856	0.871	0.797	0.795	0.796	0.812	0.843	0.845	0.856	0.886	0.795	0.848	0.852	0.904	0.916	0.974

Table 8. iGOS++ Cross-Testing. Insertion metric when generating heatmaps using the model on the first row and evaluating the heatmaps by using the model on first column.

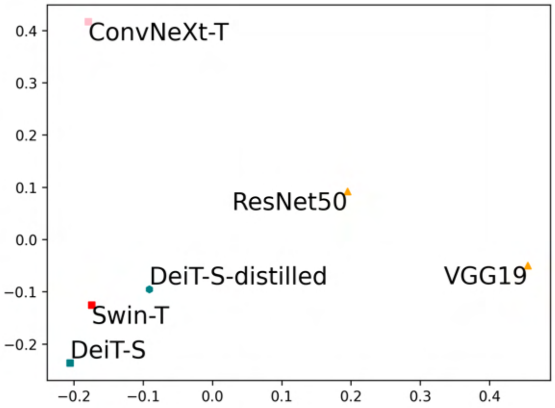


Figure 7. Kernel PCA projections of different models using the insertion metrics with a zero-image baseline.

different classifiers using Kernel PCA, which show the similarities between different models. Here we show the normalized deletion and insertion scores obtained from different classifiers.We can see that most of the algorithms have similar and fairly low deletion scores and fairly high relative insertion scores. This indicates that the attribution maps found by the algorithm explain the decisions consistently and the model is able to obtain similar confidence as the full image by only using a few top-ranked patches from the attribution map, which proves that the attribution map algorithm we use is sound as a basis for the cross-testing experiments. Note that the distilled transformer models (DeiT-S-distill, PiT-S-distill and LeViT-256) have slightly higher insertion scores which indicate they need fewer patches to achieve the same confidence as the full image than the other algorithms. DeiT-S-distill even has a relative insertion score close to 1, indicating that in many cases, partially occluded images have been more confidently predicted than the full image. Another observation is that transformers trained with the batch normalization (ConvNeXt-T-3-BN and Swin-T-4-BN) consistently exhibit significantly higher insertion scores, surpassing 1.1. This finding implies that these transformers require fewer patches to attain higher confidence levels compared to the full image than the other models.

From the results shown in Table 8 and 9 one can see the significant differences between different models. Cross-testing insertion metric is usually around 80%, which indicates that the models agree on less than 90% of the images. The similarity between models of the same category are usually higher, e.g. between VGG19 and ResNet50, and among DeiT-S-distilled, PiT-S-distilled and LeViT-256, also during ResNet50-C1, ResNet50-C2 and ResNet50-D. Still, the similarities between Swin-T and other transformer models are higher than between Swin-T and the CNN models.

In Section 4.2, we stated that distilled transformer models sometimes obtain high confidence while only showing a small number of regions. Here we provide more qualitative results from cross-testing, Fig. 8. The `Cougar` image in the second column of the first row, the heatmap is generated on Swin-T, however, both DeiT-S (84.68%) and DeiT-S-distilled(96.96%) have higher confidence than

Generate →	VGG19	ResNet50	ResNet50-C1	ResNet50-C2	ResNet50-D	ConvNeXt-T	ConvNeXt-T-3	ConvNeXt-T-3-GN	ConvNeXt-T-3-BN	Swin-T	Swin-T-4	Swin-T-4-GN	Swin-T-4-BN	Nest-T	DeiT-S	PiT-S	DeiT-S-distill	PiT-S-distill	LeViT-256
Evaluate ↓	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del	Del
VGG19	0.111	0.16	0.175	0.178	0.172	0.191	0.201	0.194	0.196	0.173	0.172	0.17	0.154	0.198	0.128	0.189	0.145	0.144	0.15
ResNet50	0.175	0.125	0.181	0.18	0.172	0.196	0.207	0.202	0.2	0.177	0.177	0.173	0.157	0.209	0.263	0.194	0.149	0.145	0.151
ResNet50-C1	0.204	0.193	0.169	0.196	0.192	0.214	0.224	0.223	0.219	0.199	0.199	0.194	0.176	0.233	0.224	0.218	0.17	0.162	0.168
ResNet50-C2	0.211	0.198	0.202	0.178	0.199	0.227	0.239	0.235	0.234	0.21	0.211	0.206	0.186	0.248	0.24	0.232	0.179	0.169	0.177
ResNet50-D	0.198	0.184	0.192	0.191	0.159	0.212	0.222	0.216	0.216	0.194	0.195	0.189	0.172	0.232	0.221	0.213	0.164	0.156	0.161
ConvNeXt-T	0.237	0.224	0.232	0.229	0.223	0.151	0.231	0.225	0.237	0.208	0.208	0.204	0.191	0.247	0.239	0.227	0.187	0.176	0.188
ConvNeXt-T-3	0.225	0.214	0.204	0.198	0.188	0.199	0.145	0.205	0.211	0.193	0.195	0.191	0.191	0.231	0.221	0.213	0.176	0.166	0.176
ConvNeXt-T-3-GN	0.238	0.225	0.222	0.217	0.202	0.215	0.229	0.151	0.233	0.206	0.209	0.201	0.189	0.246	0.236	0.227	0.183	0.172	0.184
ConvNeXt-T-3-BN	0.218	0.204	0.206	0.201	0.187	0.213	0.219	0.222	0.142	0.2	0.201	0.201	0.175	0.243	0.236	0.227	0.177	0.169	0.177
Swin-T	0.245	0.232	0.244	0.24	0.235	0.228	0.243	0.236	0.243	0.129	0.181	0.196	0.165	0.232	0.222	0.216	0.175	0.169	0.181
Swin-T-4	0.241	0.231	0.223	0.219	0.207	0.226	0.24	0.233	0.241	0.176	0.128	0.18	0.156	0.23	0.219	0.211	0.172	0.168	0.179
Swin-T-4-GN	0.239	0.228	0.226	0.223	0.208	0.228	0.243	0.233	0.243	0.18	0.176	0.122	0.156	0.231	0.219	0.213	0.173	0.169	0.18
Swin-t-4-BN	0.233	0.225	0.222	0.218	0.204	0.231	0.245	0.239	0.239	0.185	0.18	0.176	0.109	0.238	0.226	0.225	0.173	0.17	0.179
Nest-T	0.241	0.231	0.237	0.238	0.232	0.23	0.244	0.236	0.245	0.196	0.2	0.196	0.181	0.144	0.222	0.213	0.173	0.168	0.182
DeiT-S	0.244	0.231	0.249	0.249	0.239	0.247	0.266	0.255	0.266	0.204	0.202	0.197	0.18	0.241	0.127	0.214	0.144	0.165	0.178
PiT-S	0.249	0.237	0.238	0.236	0.218	0.244	0.261	0.253	0.263	0.206	0.206	0.201	0.183	0.245	0.221	0.14	0.169	0.151	0.177
DeiT-S-distill	0.244	0.232	0.246	0.246	0.236	0.246	0.263	0.257	0.264	0.205	0.205	0.198	0.18	0.245	0.204	0.221	0.095	0.16	0.173
PiT-S-distill	0.25	0.235	0.238	0.232	0.217	0.25	0.27	0.264	0.272	0.216	0.217	0.211	0.19	0.256	0.238	0.216	0.17	0.109	0.179
LeViT-256	0.257	0.24	0.253	0.251	0.245	0.254	0.269	0.263	0.27	0.224	0.221	0.219	0.197	0.263	0.25	0.241	0.186	0.178	0.118

Table 9. iGOS++ Cross-Testing. Deletion metric when generating heatmaps using the model on the first row and evaluating the heatmaps by using the model on first column.

Swin-T (77.39%) on this partially occluded image.

Also, we can see that in the case even if the figure was generated by VGG19, the distilled and non-distilled transformer models have sometimes much higher confidences on occluded images. Especially, DeiT-S-distilled more often has confidence than VGG. This shows the robustness of transformers over convolutional networks.

We also performed 3 trials of cross-testing to generate standard deviations on selected models. Table 10 shows the quantitative results of cross-testing using the same model with different seeds. We can see that, for the same model, the insertion score of cross-testing results with different seeds has reasonable standard deviations between 0.01 and with Swin-T (0.052), DeiT-S (0.047) and LeViT-256 (0.041) being the highest.

Model	ResNet50-A2	Swin-T	DeiT-S	DeiT-S-distilled	LeViT-256
Del	0.185 ± 0.012	0.161 ± 0.021	0.168 ± 0.028	0.122 ± 0.020	0.134 ± 0.028
Ins	0.885 ± 0.010	0.869 ± 0.052	0.926 ± 0.047	0.960 ± 0.019	0.931 ± 0.041

Table 10. Cross-Testing Results on Models with the same Architecture and Different Seeds. Each column represents the mean and standard deviation of Del/Ins scores obtained by doing cross-testing for models trained using different random seeds with the same architecture.

References

[1] Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. Scouter: Slot attention-based classifier for explainable image recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1046–1055, 2021. 4

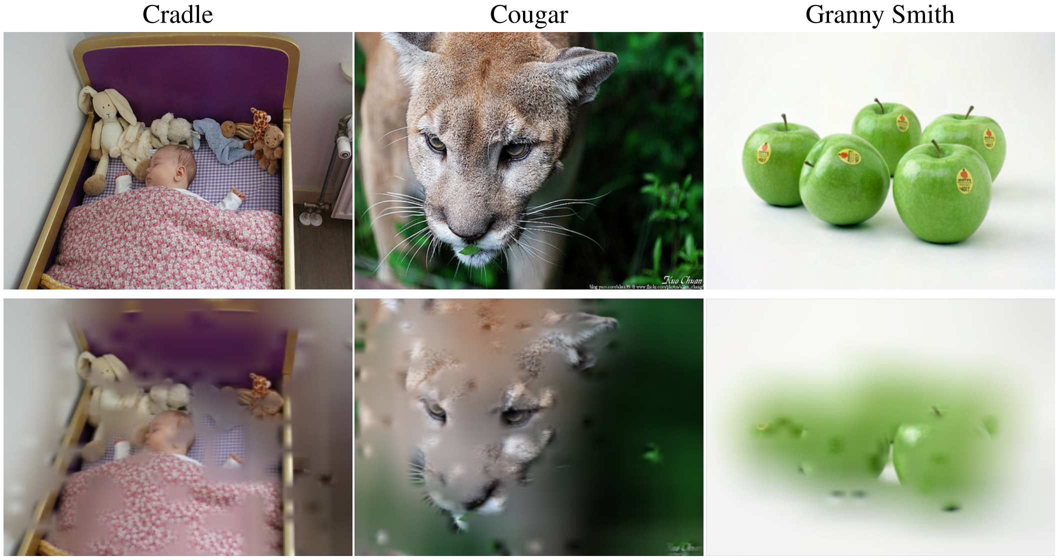
[2] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. 2

[3] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING*, pages 327–352. MIT Press, 1999. 4

[4] Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. Restricting the flow: Information bottlenecks for attribution. *arXiv preprint arXiv:2001.00396*, 2020. 4

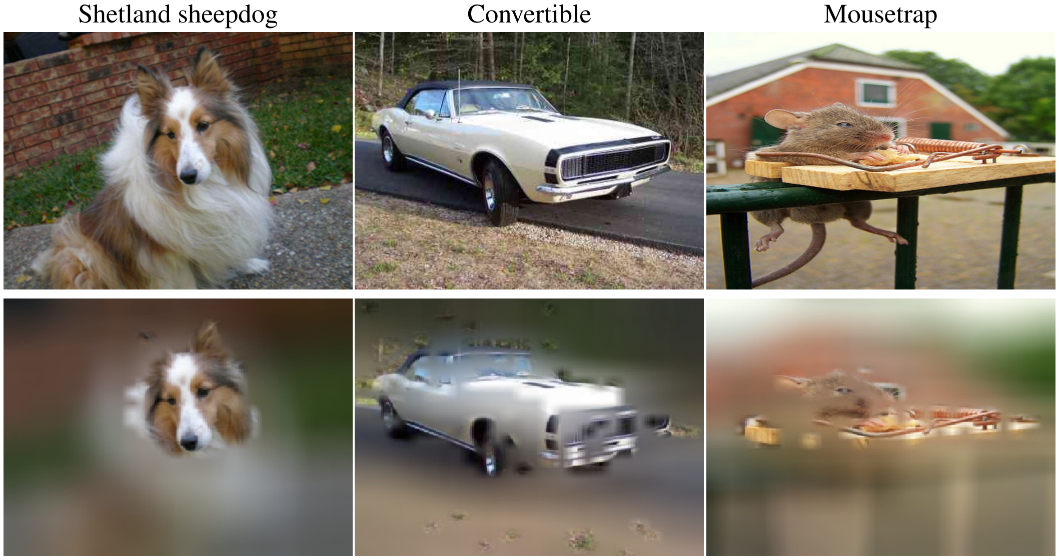
[5] Vivswan Shitole, Fuxin Li, Minsuk Kahng, Prasad Tadepalli, and Alan Fern. One Explanation is Not Enough: Structured Attention Graphs for Image Classification. In *Advances in Neural Information Processing Systems*, 2021. 3

[6] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020. 4



Prediction Confidence on the Partially Occluded Image

VGG19	ResNet50-C2	ConvNeXt-T	VGG19	ResNet50-C2	ConvNeXt-T	VGG19	ResNet50-C2	ConvNeXt-T
0.1646	0.1526	0.3726	0.1609	0.4226	0.2244	0.3474	0.0063	0.8044
DeiT-S	DeiT-S-dis	Swin-T	DeiT-S	DeiT-S-dis	Swin-T	DeiT-S	DeiT-S-dis	Swin-T
0.1831	0.9081	**0.9573**	0.8468	0.9696	**0.7739**	0.1787	0.4536	**0.6497**



Prediction Confidence on the Partially Occluded Image

VGG19	ResNet50-C2	ConvNeXt-T	VGG19	ResNet50-C2	ConvNeXt-T	VGG19	ResNet50-C2	ConvNeXt-T
0.4311	0.6174	0.2859	0.1273	0.1962	**0.8120**	0.0026	0.1846	0.0897
DeiT-S	DeiT-S-dis	Swin-T	DeiT-S	DeiT-S-dis	Swin-T	DeiT-S	DeiT-S-dis	Swin-T
0.7315	0.9677	0.8174	0.6469	0.8251	0.2517	0.5709	0.9832	**0.8725**

Figure 8. Qualitative Cross-Testing Results. The partially occluded images were generated using iGOS++ heatmaps on the algorithm with bolded number (not necessarily the highest). Then the same image is tested on multiple algorithms and we show predicted class-conditional probabilities on the ground truth class (written above).

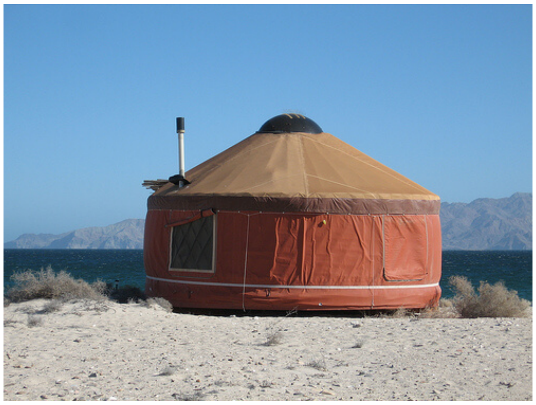


Figure 9. An image of a yurt

Figure 10. An example SAG tree explaining Swin Transformers on Fig. 9. This tree is too big to be visualized efficiently, but the sheer size of it shows the robustness of Swin Transformers to different types of occlusions. It also justifies our approach of looking at statistics rather than the visualization themselves

Figure 11. An example SAG tree explaining Fig. 9 for ConvNeXt-T. The tree size is too large to be visualized properly

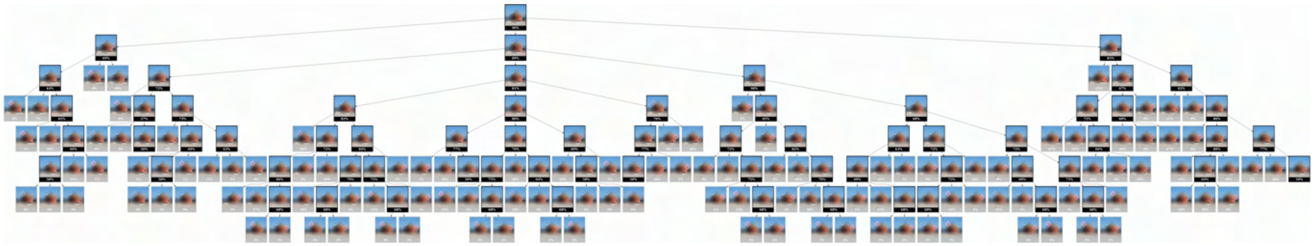


Figure 12. An example SAG tree explaining Fig. 9 for DeiT-S. The tree size is too large to be visualized properly

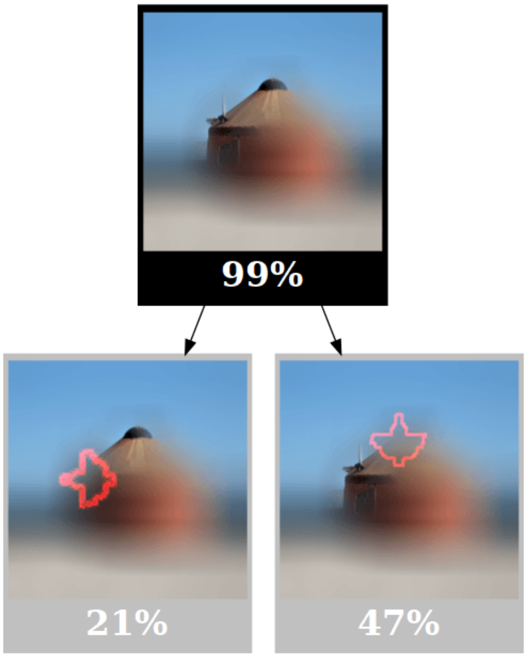


Figure 13. An example SAG tree explaining Fig. 9 for DeiT-S Distilled

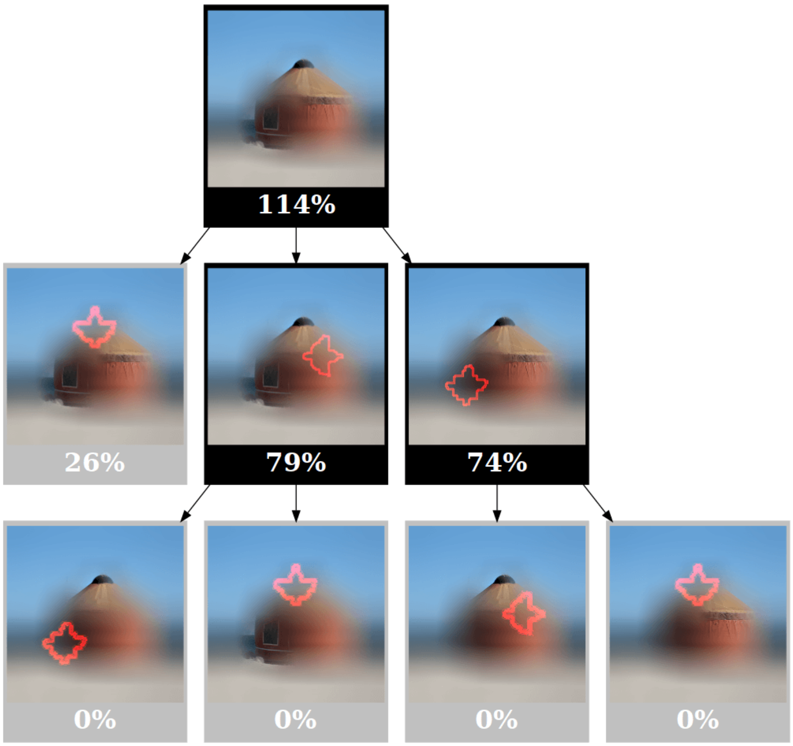


Figure 14. An example SAG tree explaining Fig. 9 for ResNet-50-C2. It can be seen that the SAG is small and focused on a very specific combination of patches of the sausage

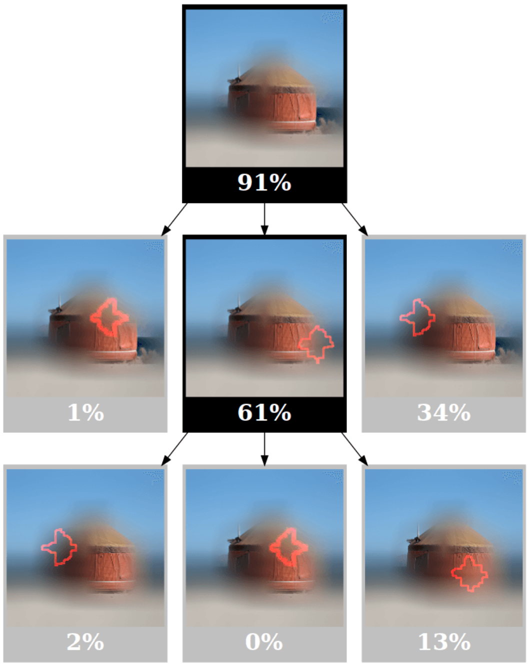


Figure 15. An example SAG tree explaining Fig. 9 for VGG. It can be seen that in many cases removal of a few parts lead to low-confidence predictions

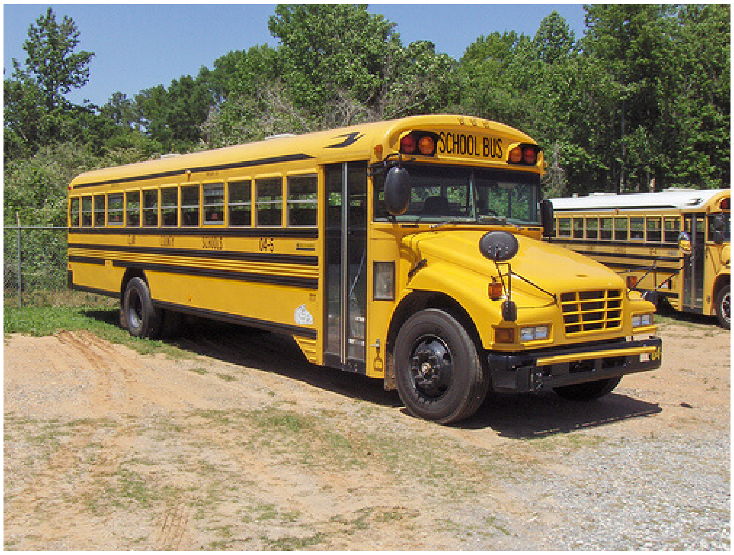


Figure 16. An image of the School Bus

Figure 17. An example SAG tree explaining Swin Transformers on Fig. 16. Again, the tree size is too large to be visualized properly

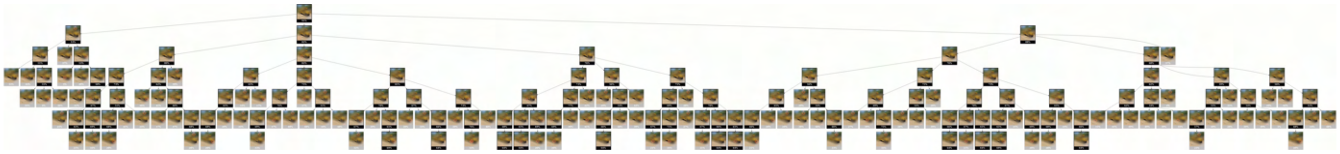


Figure 18. An example SAG tree explaining Fig. 16 for ConvNeXt-T. Again, the tree size is too large to be visualized properly

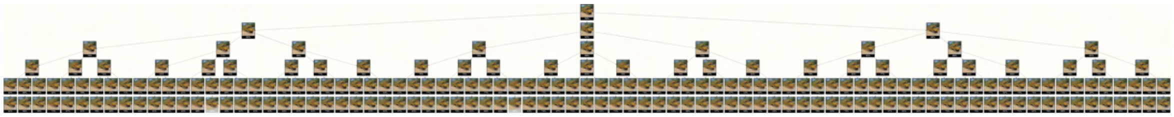


Figure 19. An example SAG tree explaining Fig. 16 for DeiT-S. Again, the tree size is too large to be visualized properly

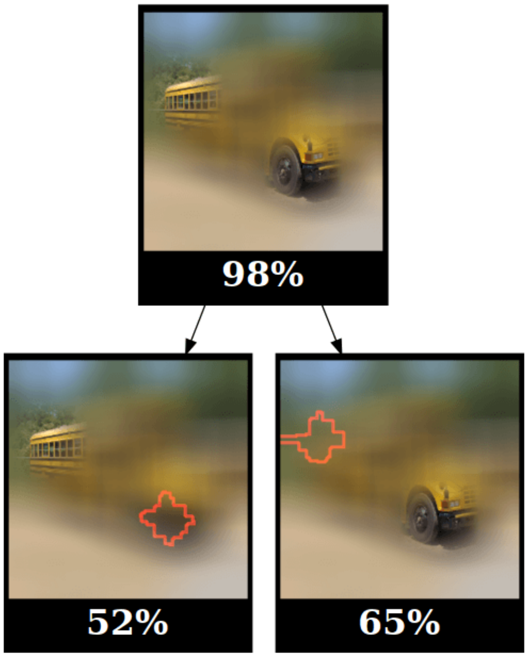


Figure 20. An example SAG tree explaining Fig. 16 for DeiT-S Distilled

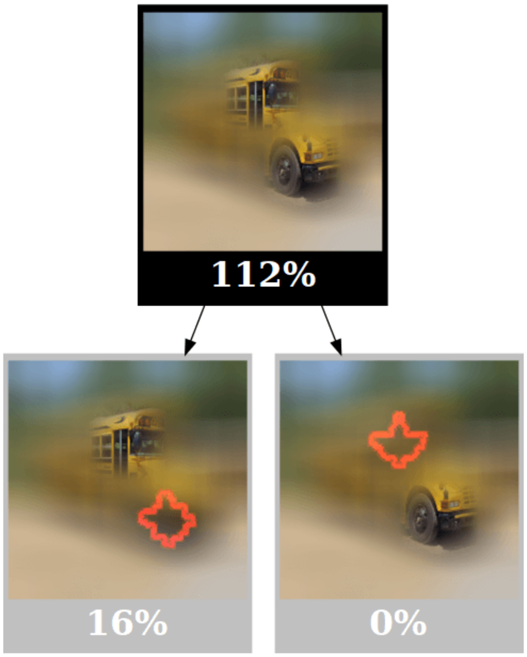


Figure 21. An example SAG tree explaining Fig. 16 for ResNet-50-C2.

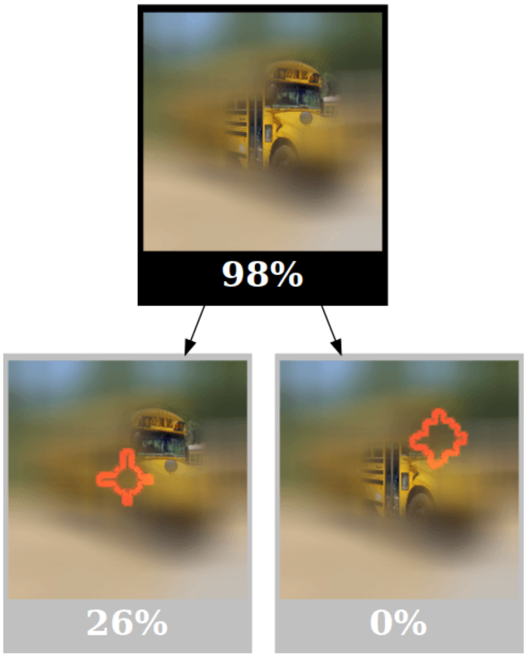


Figure 22. An example SAG tree explaining Fig. 16 for VGG.

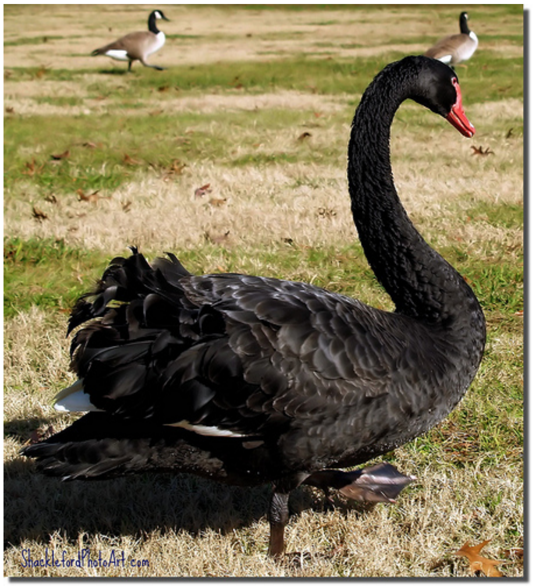


Figure 23. An image of the Black Swan

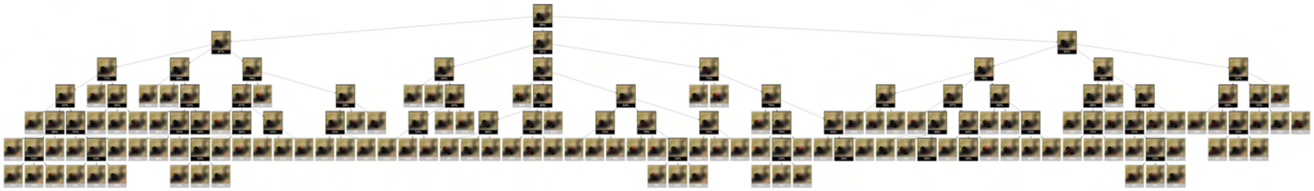


Figure 24. An example SAG tree explaining Swin Transformers on Fig. 23. Again, the tree size is too large to be visualized properly

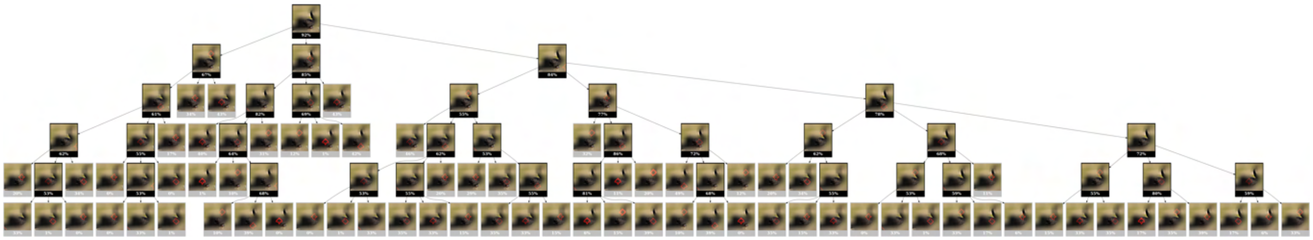


Figure 25. An example SAG tree explaining Fig. 23 for ConvNeXt-T. Again, the tree size is too large to be visualized properly

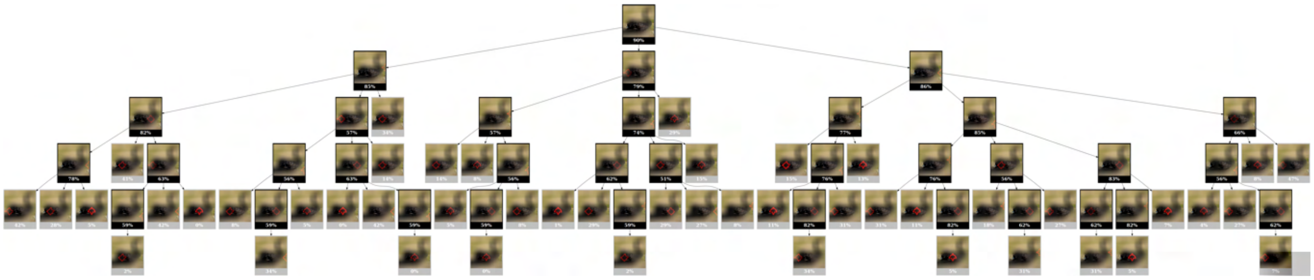


Figure 26. An example SAG tree explaining Fig. 23 for DeiT-S. Again, the tree size is too large to be visualized properly

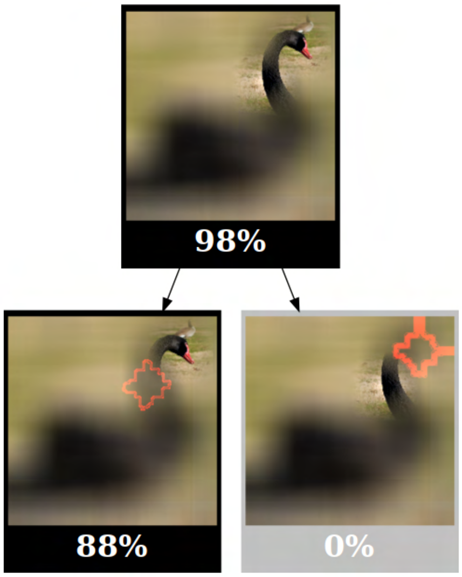


Figure 27. An example SAG tree explaining Fig. 23 for DeiT-S Distilled

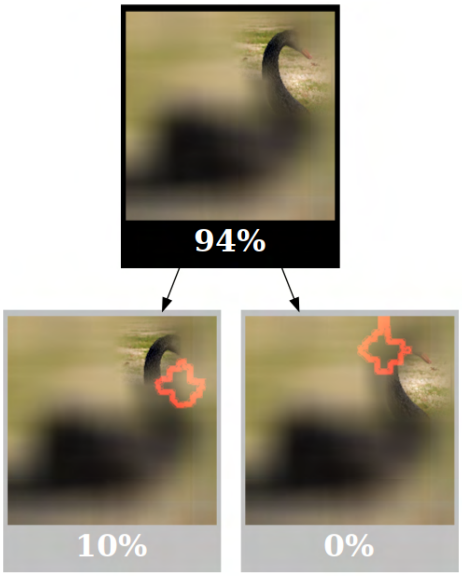


Figure 28. An example SAG tree explaining Fig. 23 for ResNet-50-C2.

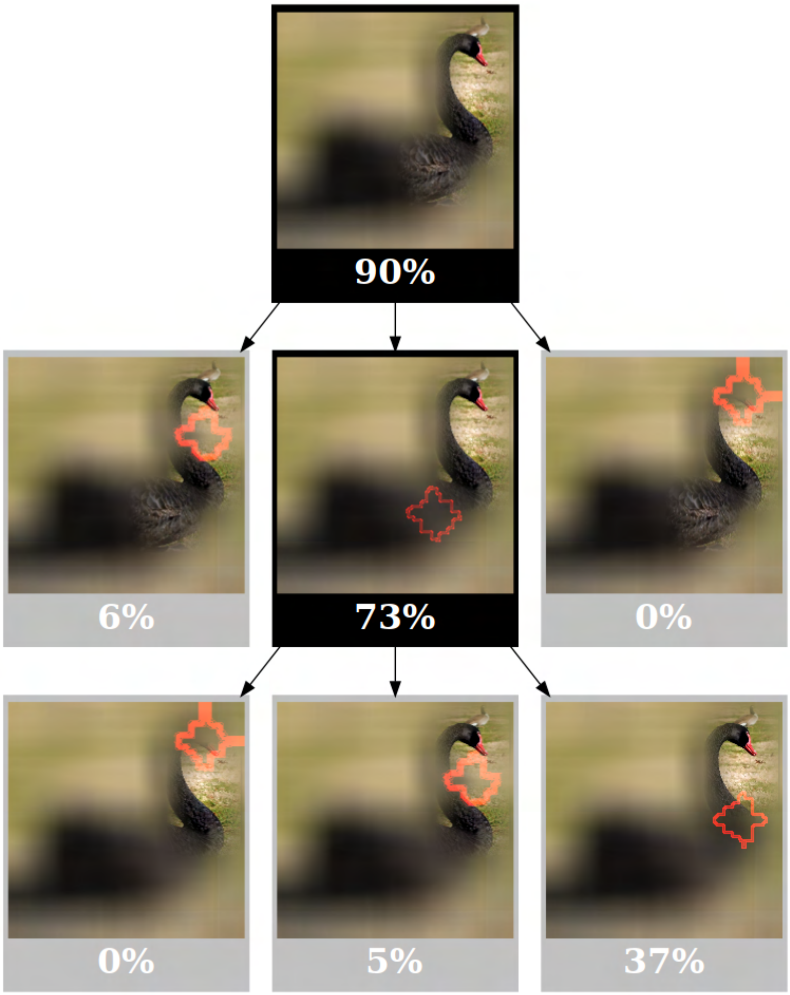


Figure 29. An example SAG tree explaining Fig. 23 for VGG.