Supplementary: A Comprehensive Study of Image Classification Model Sensitivity to Foregrounds, Backgrounds, and Visual Attributes

1. Additional Details on RIVAL10

We present a full breakdown of the RIVAL10 dataset in this section. RIVAL10 consists of ImageNet-1k samples organized into the classes of CIFAR10. Each RIVAL10 class is comprised of the training and validation samples drawn from two ImageNet-1k classes. In table 1, we present the ten classes of RIVAL10, along with the two corresponding ImageNet-1k classes per class.

In Figures 15 and 16, we present representative examples drawn at random from the dataset, along with localized attribution. Every sample has a class label and complete binary labels for 18 attributes. That is, all positive instances of attributes are marked. This differs from the partial-label setting which is common in attribute learning. Further, for every positive instance of an attribute, a segmentation mask is provided, as well as a segmentation mask for the entire object for every sample. The figures show the object mask and two positive attribute masks per image via applying the mask to the image; that is, taking the elementwise product of the segmentation mask and the image, so to black out any pixels outside of the segmentation mask.

We note that for the attributes *metallic, hairy, wet, tall, long, rectangular*, and *patterned*, we use the entire-object mask as the attribute segmentation, as these attributes pertain to the entire object. Segmentation masks can be leveraged to create many variants of RIVAL10. In Figure 1, we display examples of challenging inputs yielded via attribute removal and insertion.

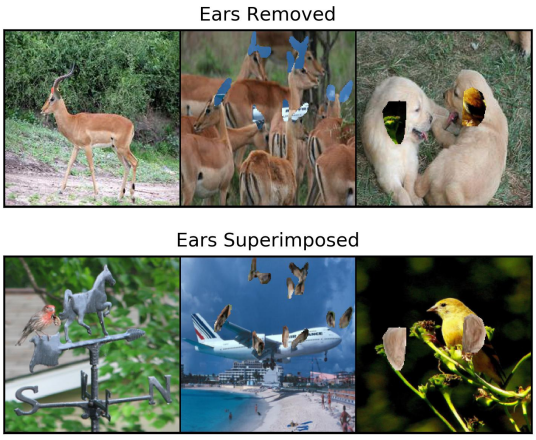

Figure 1. Examples of attribute-swapped inputs.

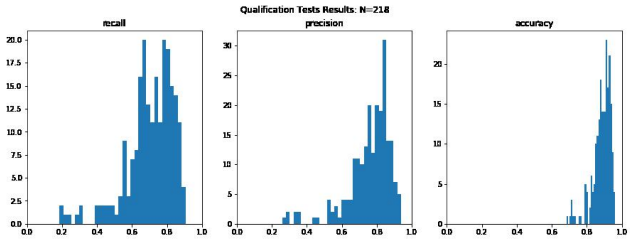

Figure 2. Histograms of Worker recall, precision, and accuracy scores on the qualification exam.

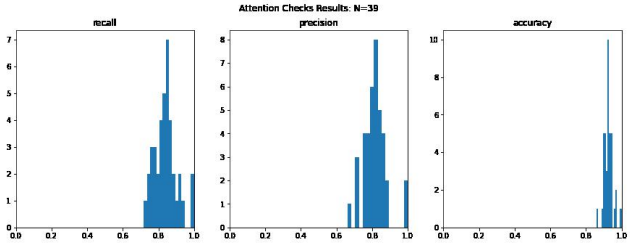

Figure 3. Histograms of per-Worker recall and precision on randomly placed attention checks during the main phase of data collection.

2. Additional Details on Data Collection

Worker Pool: We selected workers from the US to promote English fluency, which is necessary for reading the instructions. We also selected workers who have completed $> 95\%$ of their tasks to further promote successful task completion.

Worker Payment: Each task of 20 images was estimated to take 10 minutes. We set a rate of \$1.50 per task, which amounts to \$9.00 / hour, which is 25% above the US Federal Minimum Wage (\$7.25) at time of this writing. In the second phase of collection, workers were compensated at a rate of \$0.1 per segmentation. We estimate one segmen-

RIVAL10 Class	Number of Instances	ImageNet-1k Classes comprising RIVAL10 Class				Positive Attributions
		Class Name #1	WordNet ID #1	Class Name #2	WordNet ID #2	
Truck	2523	Moving Van	n03796401	Semi	n04467665	13577
Car	2665	Waggon	n02814533	Convertible	n03100240	9415
Plane	2655	Airliner	n02690373	Military plane	n04552348	15277
Ship	2660	Ocean liner	n03673027	Container vessel	n03095699	14122
Cat	2667	Persian cat	n02123394	Egyptian cat	n02124075	9309
Dog	2660	Labrador retriever	n02099712	Golden retriever	n02099601	11251
Equine	2663	Sorrel	n02389026	Zebra	n02391049	13343
Deer	2657	Gazelle	n02423022	Impala	n02422699	12274
Frog	2667	Tailed Frog	n01644900	Tree-frog	n01644373	5317
Bird	2667	Goldfinch	n01531178	Housefinch	n01532829	8822
Total: $\mathbf{26,484}$ instances ($21,178$ train, $5,308$ validation) with $\mathbf{112,707}$ positive attributions (~ 4.26 per image)						

Table 1. Breakdown of RIVAL10 dataset. Corresponding ImageNet-1k classes listed.

tation to take 30-45 seconds on average, which amounts to a wage of $9-12 an hour.

Qualification Exam: As discussed in the main text we required workers to pass a qualification exam for access to the main phase of data collection. The qualification exam consisted of 20 images with ground-truth annotations which we defined. Workers were asked to read the instructions carefully and complete the exam. We then computed precision, recall, and accuracy metrics on these questions. A total of 218 workers took the exam. All workers were paid a $1.50 for the exam, regardless if they passed or not. We report the distribution of worker scores in Figure 2.

We use these distributions to inform a chose of threshold for passing the exam, where the two relevant decision factors are (1) high bar for metrics to promote annotation quality (2) a large pool of workers for higher rate of data collection. We note that since attributes are sparse, accuracy is not a good metric for distinguishing worker performance. This can be seen in the concentration of values in Figure 2 (right). We found that 90 workers scored greater than or equal to 0.75 in precision and recall *jointly*, and decided to use this as our threshold. Of the workers who completed the qualifying exam, $N = 39$ contributed to the main phase. The number of annotations completed by each worker varied (min: 20, max: 1000).

Attention Checks: We additionally measure worker performance during the main phase of data collection through attention checks. Overall, 4% of samples to annotate had ground truth annotations completed by the authors. This allows us to estimate worker quality during the main phase, and ensure that worker attention is maintained. The ground truths for these attention checks were collected from a pool of trusted CS graduate students.

Overall metrics on these attention checks were similar to threshold set for the qualification exam: the average precision and recall *across workers* were 0.81 and 0.84 respectively. We report these per-worker metrics in Figure 3.

Collection of Segmentations: In a second pass, workers submitted segmentation masks for any attribute positively annotated previously. Workers had access to many tools to complete segmentations, including zooming, a polygon tool, and a brush. Detailed example segmentations were provided per attribute. Figure 27 shows a screen shot of the segmentation platform. A similar qualification check was administered before the second phase of data collection, with a minimum average IOU of 0.7 required on at least five segmentations. Also, an average IOU of 0.745 was achieved on attention checks.

Screenshots of Instructions Given to Workers: We show screenshots of the instructions, consent form, examples, and annotation form in Figures 23, 24, 25, and 26 respectively. We have redacted identifying information of the authors appropriately.

3. Model Details

Our experiments included a diverse set of model architectures and training paradigms. A primary challenge of our work was facilitating fair comparisons across models that operate very differently from one another at train and test time. In this section, we provide greater discussion on the differences among the models and their affect on our analysis.

3.1. Architectures and Training Procedures

Architecturally, we focus on ResNets [5] and Transformers [2]. Both architectures are deep, consisting of many layers, though the nature of layers are markedly different. ResNets rely on convolutions, which introduce the spatial inductive biases such as translational invariance. Transformers, on the other hand, view an image as a collection of

Model	Pretraining Set	Parameter Count	RIVAL10 Accuracy	Source of Weights	Original Paper	Notes
ResNet18	IN-1k	11.4M	95.48	[7]	[5]	
ResNet50		23.9M	99.10			
ResNet101		42.8M	99.21			
ResNet152		58.5M	99.43			
Robust ResNet18	IN-1k	11.4M	91.80	[3]	[6]	ℓ_2-PGD, $\epsilon = 3.0$
Robust ResNet50		23.9M	93.82			ℓ_2-PGD, $\epsilon = 3.0$
Robust ResNet18†		11.4M	93.69			ℓ_2-PGD, $\epsilon = 1.0$
Robust ResNet50†		23.9M	97.29			ℓ_2-PGD, $\epsilon = 1.0$
SimCLR	IN-1k	23.9M	93.87	[4]	[1]	RN50 backbone
CLIP ResNet50	YFCC100M	23.9M	96.34	[8]	[8]	
CLIP ResNet101		42.8M	96.27			
CLIP ViT-B/16		86M	99.17			Patch=16×16
CLIP ViT-B/32		87M	98.44			Patch=32×32
ViT (Tiny)	IN-21k + IN-1k	5M	94.82	[12]	[2]	Patch=16×16
ViT (Small)		22M	98.96			Patch=16×16
ViT (Base)		86M	99.64			Patch=16×16
ViT (Small)†		23M	97.86			Patch=32×32
ViT (Base)†		87M	99.26			Patch=32×32
DeiT (Tiny)	IN-1k	5M	96.42	[12]	[11]	Patch=16×16
DeiT (Small)		22M	99.30			Patch=16×16
DeiT (Base)		86M	99.74			Patch=16×16

Table 2. Details on all models analyzed. † denotes models that were only considered in specific ablations (i.e. not present in main figures). IN refers to ImageNet.

patches, an apply attention layers to allow distant patches to effect one another. Thus, images are processed significantly differently across the two architectures. However, seeing as both architectures are used in image classification, comparisons are warranted and necessary. Other works also compare transformers and ResNets, as mentioned in the main text.

Among training procedures, most models seek to minimize cross entropy loss, using single class-label supervision on clean training samples. Robust ResNets instead undergo adversarial training [6], which replaces clean training samples with adversarially attacked ones. These models are then robust in the sense that they admit far fewer adversarial examples, where imperceptible perturbations cause models with high clean accuracy to badly misclassify attacked inputs.

We also consider contrastively trained models, which differ dramatically in that they do no use class-labels during training. The contrastive loss refers to training encoders to draw representations of similar inputs close to one another, while simultaneously pushing representations of different inputs apart. In SimCLR [1], two views of a single input are created via data augmentation. In CLIP [8], the representation of an image is contrastively drawn to the representation of a corresponding *text* caption, obtained using two separate encoders (image and text) that share a latent space, remarkably extending contrastive learning to multiple encoders operating on different mediums. Notice that neither SimCLR nor CLIP has the exclusive objective of image classification, like the other supervised models we study. Instead, they seek to learn informative representations, which can then be used for a variety of downstream tasks. However, object recognition is one of the main downstream task considered, and it is by no means abnormal to finetune SimCLR or CLIP encoders to perform image classification. We note that CLIP models have also been shown to have impressive zero-shot classification abilities. We leave investigation of CLIP's zero-shot classification to future work.

3.2. A Single Test Environment

Given that models differ in their training algorithms and settings, we seek to create a single testing environment that preserves feature spaces learned in pretraining. Simply, we isolate feature extractors, usually by removing the final classifying layer (if present). We then fit a linear layer atop the fixed features via supervised training on RIVAL10. Specifically, we use an Adam optimizer with learning rate of $1e^{-4}$, betas of $0.9, 0.999$, and weight decay of $1e^{-5}$, for ten epochs. When finetuning on background ablated images, we allow for an additional ten epochs. As seen in

table 2, all models achieve over 90% test accuracy using our simple finetuning process. We do not wish to compare model accuracies, though we argue that high accuracies across the board show that no model is significantly disadvantaged with respect to its classification ability.

3.3. Other Factors of Variation

Differences in network size and pretraining set, listed in table 2, are two other significant factors of variation across the models we compare. Most models only use ImageNet-1k as the pretraining set. ViTs and CLIP models use larger datasets. While this is not ideal, differences are unavoidable in any comparison, and we argue that the pretraining sets fundamentally inform the models themselves, similar to how architecture and training procedure do. In the case of ViTs, we also consider DeiTs, which are only trained on Imagenet-1k, allowing for direct inspection of the effect of the larger pretraining set on transformer behavior.

As for varying network sizes, we take multiple measures to paint a full picture. First, we take models of varying size within each category of interest. We find that across model types, larger networks achieve higher accuracies for clean and noisy samples. Our primary metric (RFS), however, normalizes for general noise robustness. Secondly, for all model types aside form CLIP ViTs, we include an instance with roughly 23M parameters. When only comparing these models, the same trends emerge.

4. RFS and other Normalizations

We propose relative foreground sensitivity (RFS) as a normalized measure to directly compare the sensitivities of models with varying general noise robustness. In this section, we expand on the derivation of RFS, and present results using L_2 normalized noise.

4.1. Geometric Derivation of RFS

Recall that the founding logic of our sensitivity analysis is that a model's sensitivity to a region can be measured by the degradation in performance due to noise corruption of that region. However, models with greater general robustness to noise will see lesser degradation due to noise in either region. Similarly, models with low noise robustness may see severe degradation due to noise in both regions. RFS is designed to normalize against variance in general noise robustness, yielding a single measure to compare various models across.

In figure 4 (left), we consider a point with accuracies a_{fg}, a_{bg} under foreground and background noise respectively. Further, we assume $\overline{a} = 1/2(a_{fg} + a_{bg}) \leq 0.5$ and $a_{fg} < a_{bg}$. Now, the distance from (a_{fg}, a_{bg}) to the diagonal (dashed green) is equal to the distance to $(\overline{a}, \overline{a})$, which amounts to

$$\text{Distance to Diagonal} = \sqrt{2}(\overline{a} - a_{fg}) = \frac{\sqrt{2}(a_{bg} - a_{fg})}{2}$$

The maximum distance from the diagonal for a point with general noise robustness $\overline{a}$ then corresponds to the length of the green segment (solid and dashed). Here, the limiting factor is that $a_{fg} \geq 0$. This distance is

$$\text{Max Distance to Diagonal} = \sqrt{2}(a_{fg} + a_{bg} - \overline{a}) = \sqrt{2}\overline{a}$$

Thus, $RFS = \frac{\sqrt{2}/2(a_{bg} - a_{fg})}{\sqrt{2}\overline{a}} = \frac{a_{bg} - a_{fg}}{2\overline{a}}$ when $\overline{a} \leq 0.5$.

Now, we consider a point (a'_{fg}, a'_{bg}) with $\overline{a'} = 1/2(a'_{fg} + a'_{bg}) > 0.5$. The distance to the diagonal (dashed blue) is identical to the first case. Here, the maximum distance from the diagonal (full blue segment) is limited by the fact that $a'_{bg} \leq 1$. This yields

$$\text{Max Distance to Diagonal} = \sqrt{2}(1 - \overline{a'})$$

leading to a final RFS of $\frac{a'_{bg} - a'_{fg}}{2(1 - \overline{a'})}$ when $\overline{a'} > 0.5$. Combining these cases gives the general formula for RFS.

$$RFS = \frac{a_{bg} - a_{fg}}{2\min(\overline{a}, 1 - \overline{a})}$$

Intuitively, RFS measures the gap in accuracy under background and foreground noise under a normalization. The normalization is designed to account for the fact that models with very high or very low noise robustness will be limited in the maximum gap attainable. In Figure 4, we visualize both general noise robustness and RFS for all accuracies under foreground and background noise to add further context.

4.2. Results under L_2 Normalization of Noise

We now reproduce the major figures from our noise analysis under L_2 normalized noise. We consider eight equally spaced noise levels, with L_2 norms ranging from 25 to 200. We find that the trends are near identical. The one small difference is that the class distinctions in Figure 7 are slightly less severe. In particular, for DeiTs, the distribution of $iRFS$ scores on birds is roughly the same as that on ships. Recall that applying equal L_∞ noise to two regions will incur a greater perturbation to the larger region when measuring under the L_2 norm. Thus, L_∞ noise could introduce a bias where larger regions are corrupted more. The direction of this bias is unclear though, as relative sizes of foregrounds and backgrounds vary. Our corroborated results under L_2 normalized noise suggest that the aforementioned bias has little effect on our conclusions.

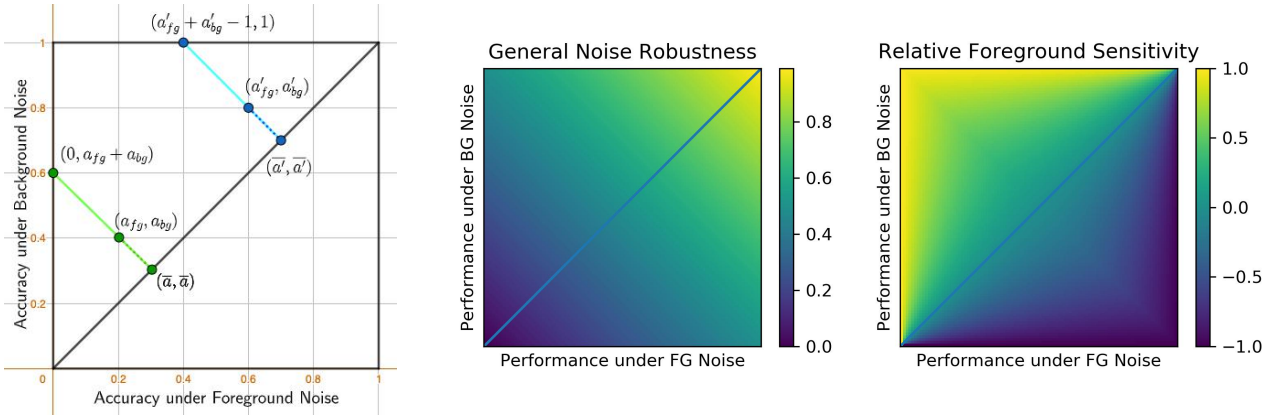

Figure 4. (**Left**) We demonstrate how RFS is derived as a ratio of the distance of (a_{fg}, a_{bg}) from the diagonal over the maximum distance to the diagonal for a point with fixed noise robustness $\overline{a} = 1/2(a_{fg} + a_{bg})$. (**Right**) Visualization of general noise robustness and relative foreground sensitivity for all points in the unit square. Moving along the main diagonal increases general noise robustness, and moving away (above) increases relative foreground sensitivity.

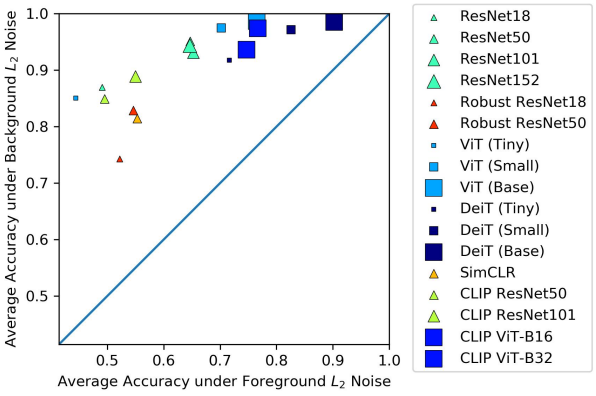

Figure 5. Accuracy under $\mathbf{L_2}$ normalized noise averaged over multiple noise levels. Marker size is proportional to parameter count. Models with higher relative foreground sensitivity lie further from the diagonal.

5. Saliency Alignment

To complement the noise analysis, we inspected saliency maps obtained via GradCAM [9], which assigns a saliency score of 0 to 1 to each pixel. RIVAL10's segmentation masks allow for quantative assessment of the alignment of saliency to foregrounds. We inspected five metrics, defined as follows for a true binary object segmentation mask $\mathbf{m} \in \{0,1\}^d$ and a saliency map of equal shape $\mathbf{s} \in [0,1]^d$. Let $\mathbf{s}_\tau \in \{0,1\}^d$ be a binarized version of the saliency map, where a pixel of $\mathbf{s}_\tau$ is 1 only when its corresponding value in $\mathbf{s}$ is at least τ. A standard metric in comparing segmentations is intersection over union (IOU), defined below.

$$\text{IOU} = \frac{\sum(\mathbf{m} \odot \mathbf{s}_{\tau=0.5})}{\sum(\mathbf{m}) + \sum(\mathbf{s}_{\tau=0.5})}$$

Here, we assess the quality of the binarized saliency map as a segmentation mask of the foreground. We also found that inspecting the difference in average saliency for foreground and background pixels were useful, particularly in automatically discovering spurious background features. We define this metric, called Δ Densities, below.

$$\Delta \text{ Densities} = \frac{(\sum \mathbf{m} \odot \mathbf{s})/\sum(\mathbf{m})}{(\sum(\mathbf{1} - \mathbf{m}) \odot \mathbf{s})/\sum(\mathbf{1} - \mathbf{m})}$$

A third measure views saliency alignment as a binary classification task. Specifically, we compute average precision of a detector that uses pixel saliency as the discriminant score for classifying each pixel as foreground or background. Average precision combines recall and precision at all thresholds to give a general sense of discriminatory ability of some criteria. Formally,

$$\text{Average Precision} = \sum_n (R_n - R_{n-1}) P_n$$

where R_n, P_n refer to the precision and recall obtained at the n^{th} threshold. Finally, we consider two additional metrics are analogs to precision and recall. Precision and recall typically hold meaning in binary classification tasks, though in our case, we wish to assess the alignment of saliency maps with continuous values (i.e. not true or false predictions). To this end, we define Saliency Precision and Saliency Recall as follows.

$$\text{Saliency Precision} = \frac{\sum \mathbf{s} \odot \mathbf{m}}{\sum \mathbf{s}}$$

Essentially, this amounts to a weighted precision, placing more importance on having highly salient pixels fall in the

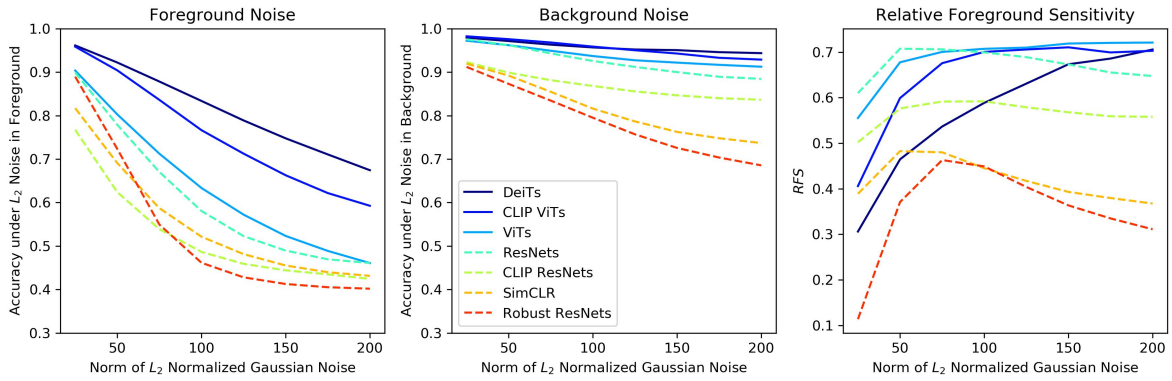

Figure 6. Accuracy under $\mathbf{L_2}$ normalized noise in foreground (**left**) and background (**middle**) at various noise levels. Models are grouped by architecture and training procedure, with a curve corresponding to the average over all models in a group. (**Right**): RFS by group.

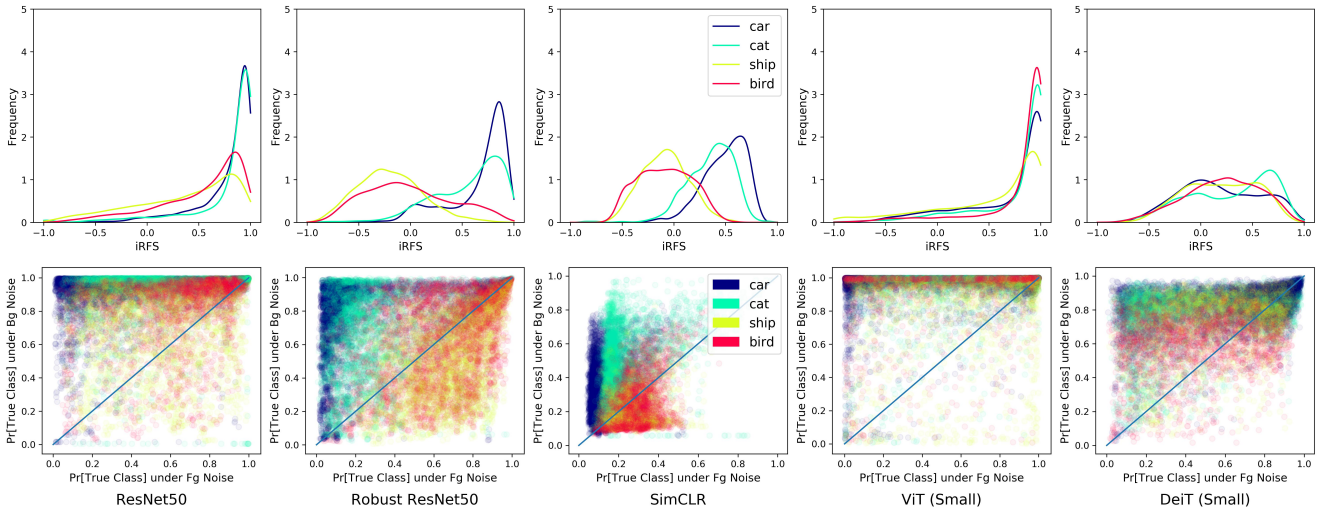

Figure 7. Relative foreground sensitivity per instance for four classes and five models of roughly equal size, computed using $\mathbf{L_2}$ normalized noise corruption. (**Top**): Histogram of $iRFS$; positive denotes greater foreground sensitivity. (**Bottom**): Scatter; top left indicates high relative foreground sensitivity. Class distinction is slightly less pronounced than with L_∞ noise, but still substantial.

foreground. Another interpretation of this metric is the fraction of total saliency in the foreground, similar to [10].

$$\text{Saliency Recall} = \frac{\sum \mathbf{s}_{\tau=\tau^*} \odot \mathbf{m}}{\sum \mathbf{m}}$$

For Saliency Recall, we compute recall as normal on a binarized saliency map. However, the binarization threshold τ^* is chosen dynamically so to only retain the pixels that account for 75% of total saliency. That is, $\frac{\sum \mathbf{s}_{\tau=\tau^*}}{\sum \mathbf{s}} = 0.75$. Intuitively, saliency recall captures the fraction of the segmentation mask that are among the more salient pixels.

5.1. Empirical Observations

We present complete quantitative saliency alignment results in Figure 8. Generally, there is not a strong separation among models observed across all metrics. CLIP ViTs consistently score lower, with an average Δ Densities near zero. ViTs also generally have lower saliency alignment. Recall that at low noise levels, the transformer models had low relative foreground sensitivity. One may be inclined to argue that the saliency alignment analysis corroborates those results. However, we hesitate to make such assertions, as the results are not consistent across metrics, and key exceptions (such as the high alignment of DeiTs and Robust ResNets) exist. Our overall impression from the saliency analysis is that alignment GradCAMs to foregrounds may not always imply high relative sensitivity to foreground noise, suggesting that saliency maps alone may not capture the full story of a model's sensitivities.

Qualitatively, the GradCAMs for all transformers are much more patchy than ResNets, which usually yield Grad-

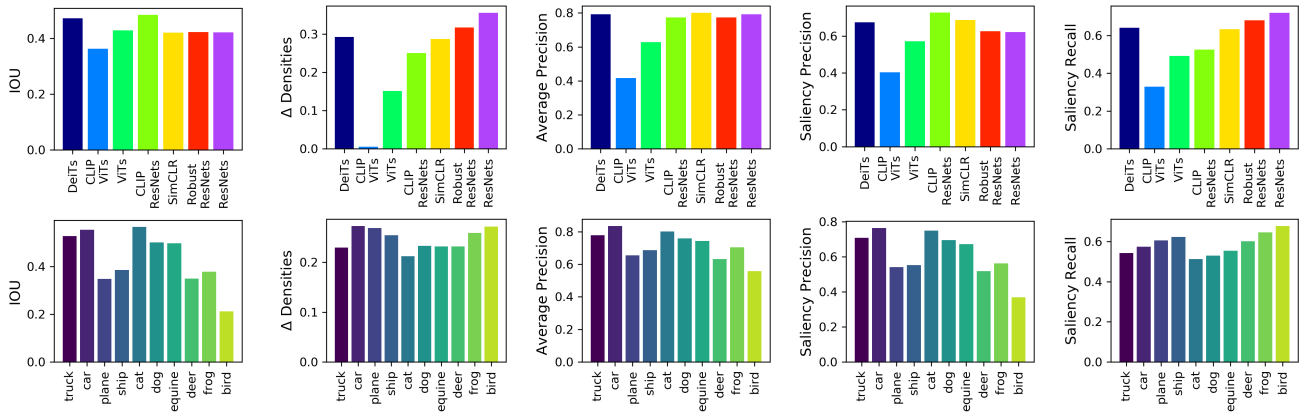

Figure 8. Saliency alignment averaged over model categories (top) and object classes (bottom) for five alignment metrics.

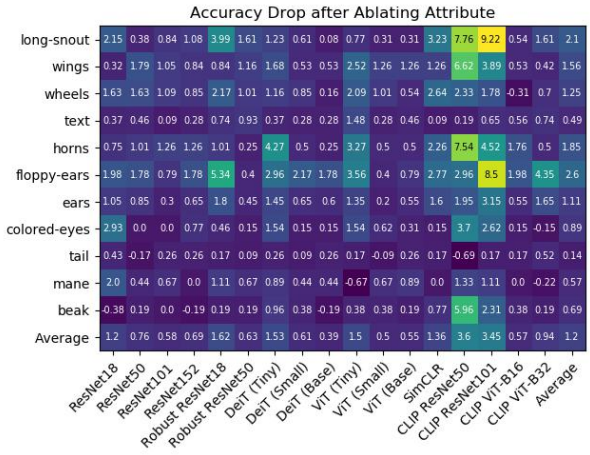

Figure 9. Degradation to model performance due to attribute ablation (via graying), as measured by accuracy.

CAMs with saliency organized in one or two clusters. We attribute this to the fundamental difference in how images are processed by ResNets, who employ significant spatial inductive biases, and transformers, who view images a set of patches that can attend to one another.

Looking to object classes, we see that the variance in alignment due to class observed for IOU is corroborated by average precision and saliency precision. When inspecting saliency recall, however, we see higher alignments for birds and ships. We believe this is an effect of the bias of Saliency Recall in favor of images with smaller foreground masks. Furthermore, high recall can still be consistent with poor foreground sensitivity, as the saliency map may cover much of both the foreground and background.

6. Attribute Ablation

To assess sensitivity to attributes, we inspect the extreme of ablating the attributes entirely via graying. We do not consider attributes that cover the entire object, as ablating the attribute would remove the entire foreground. Overall, the removal of any individual attribute only slightly reduces model performance. The largest reduction occurs in CLIP ResNets, with an average drop in accuracy of roughly 3.5%. For most attribute-model pairs, accuracy drop is less than 1%. This suggests that attributes human deem informative in performing RIVAL10 classification are not very important to deep classifiers.

7. Additional Qualitative Examples of Background Sensitivity

We provide additional examples qualitatively demonstrating instances where models have high background sensitivity. Figure 10 shows GradCAMs where saliencies have worst alignment with foreground, as measured by Δ Densities, for a Robust ResNet50. In Figure 11, we display instances where noise corruption reveals greater background sensitivity for Robust ResNet50 and DeiT (Small).

8. Additional Visualizations for Neural Node Attribution

We show GradCAMs and IOU histograms for another top feature attribute pair, cars and wheels, in Figure 12. We observe qualitatively the same results as in the main text: IOU scores are high for this attribute on samples in this class. We also show scatterplots of IOUs vs. feature activations for this top pair as well as dogs and floppy-ears, the pair discussed in the main text, in Figures 13 and 14 respectively. Interestingly, feature activation value and saliency alignment (as measured by IOU) do not seem to be strongly

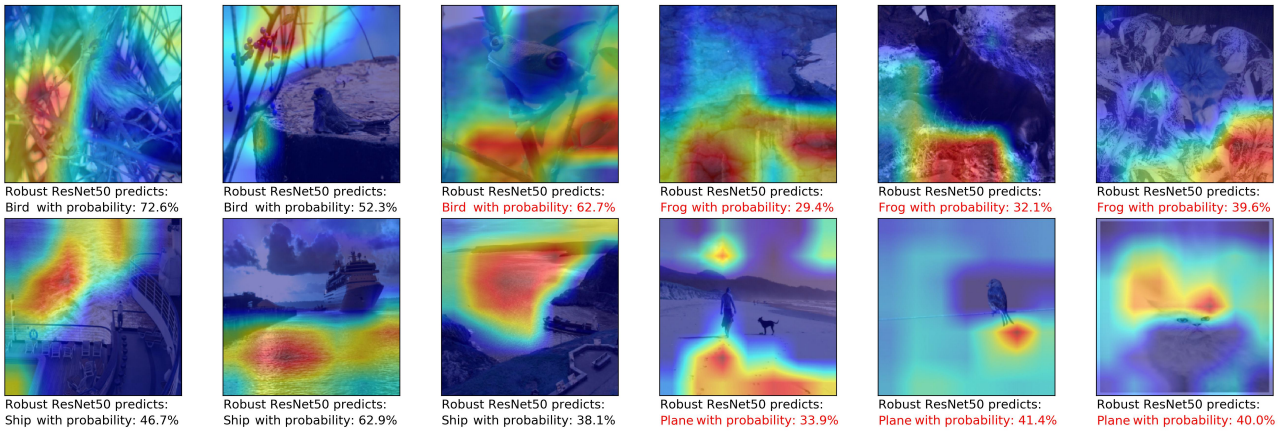

Figure 10. Additional examples of spurious features used by a Robust ResNet50 observed via sorting images by saliency alignment (Δ Densities). Misclassifications are in red text. Spurious features include branches, dry leaves, water, and sky.

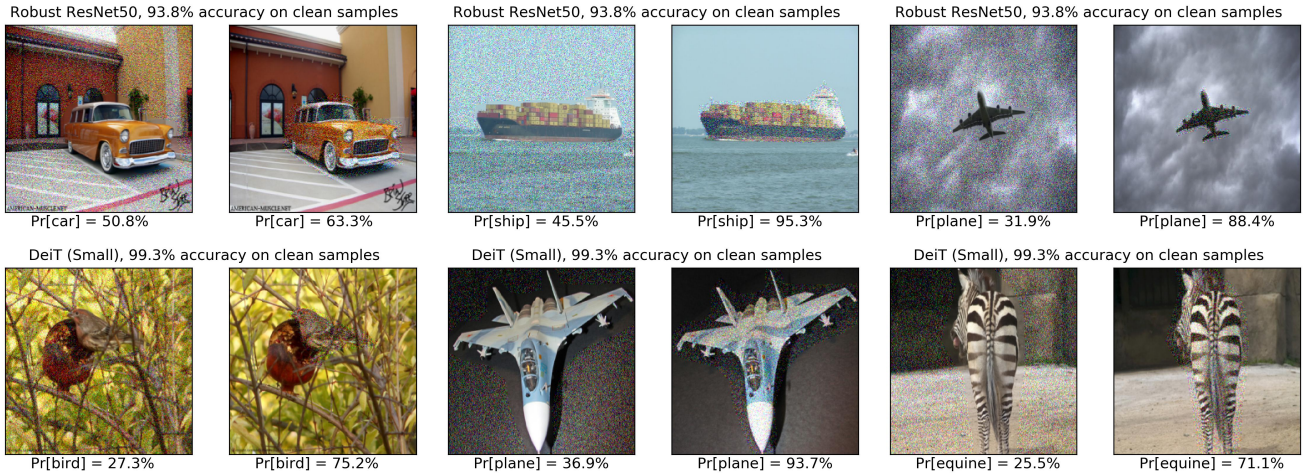

Figure 11. Additional examples where background noise degrades performance of highly accurate models more than foreground noise. (**top**): Robust ResNet50, (**bottom**): DeiT (Small). Gaussian ℓ_∞ noise with standard deviation $\sigma = 0.12$ shown. Probabilities are averaged over ten trials.

related.

9. Attribute-specific Neural Node Attribution

We report a variant of the neural node attribution section in the main text, where we do not filter by class. Instead, we focus the analysis on attributes. We use the same procedure to identify top feature attribute pairs as in the main text, *except* for filtering by class. We show the complete saliency results for top feature attribute pairs for all attributes in Figures 17, 18, 19, 20. In addition, we show activation histograms for top feature attribute pairs identified by the method, colored them by the presence or not of the attribute in Figures 21 and 22. We observe that the feature distributions do not separate for test samples with and without that attribute, *despite* the reasonable quality of the GradCAMs. Note that we present GradCAMs on the top activating test images. The GradCAMs for top activating training images are even better, though this by design, as we choose feature-attribute pairs to maximize saliency alignment in training images.

This implies that filtering by class is necessary for the node attribution methods here discussed. When the same analysis is carried out irrespective of class, nodes cannot clearly be attributed. This result casts doubt on performing node attribution in *class-free* fashion via saliency methods, though some authors argue that filtering by class reflects the actual practice of node attribution via saliency methods.

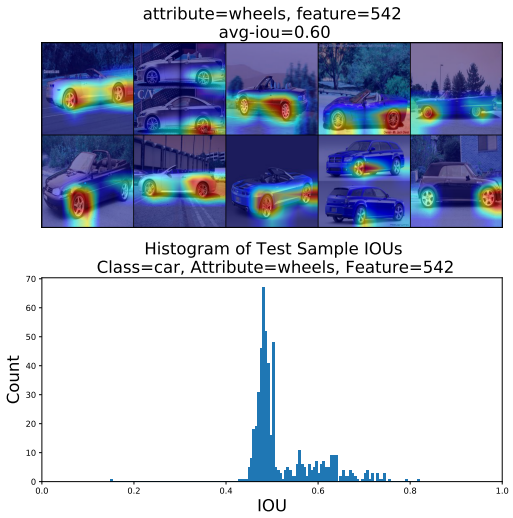

Figure 12. (**Top**): Example GradCAMs on test images with respect to the top feature identified by IOU in training set. (**Bottom**): Histograms of IOUs corresponding to this feature, attribute pair.

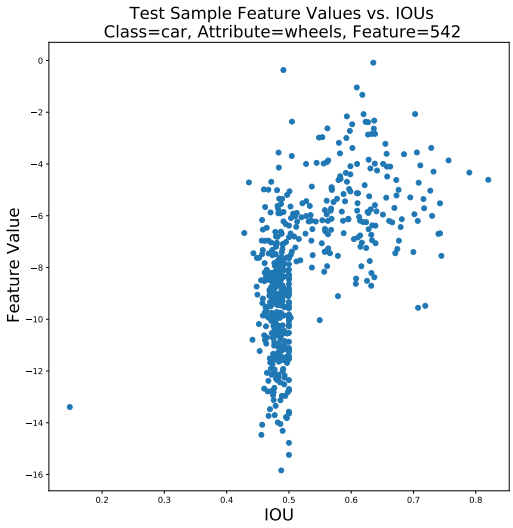

Figure 13. Feature values vs IOU scores for class-attribute pair car and wheels.

10. Limitations

The central challenge of our work is performing comparisons across diverse model types. In particular, the variance in general noise robustness poses as a major obstacle in employing our noise analysis. We believe that we have devised a normalization scheme to account for this, though there are likely other differences across models that could not be completely controlled against.

Moreover, our study only considers classification on ten

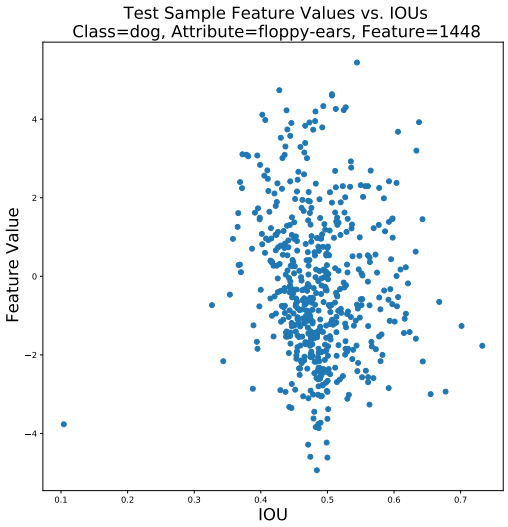

Figure 14. Feature values vs IOU scores for class-attribute pair dog and floppy-ears.

relatively disparate classes. It is possible that as the classification task becomes more challenging, models rely less on short cuts out of necessity. However, it is also plausible that they make greater use of spurious features, as they seek any information that will help. Frankly, our study can not directly anticipate the outcome of repeating our analysis for a more difficult classification task. In future work, we may build on RIVAL10 to craft more finegrained classification tasks, perhaps leveraging attribute insertion and removal.

Lastly, we focus on only one saliency method throughout our analysis. It is possible that other saliency methods may produce maps that were more informative, or more in line with the results of our noise analysis. We chose GradCAM because of its popularity and did not include others because the saliency analysis was not the central focus of our work.

11. Statement of Potential Harms

All AI technology has the potential to cause harm to others and this work is no exception. Our work targets improved robustness and interpretability of deep models, which authors believe may help reduce harm by permitting transparent explanation of model decisions.

12. Code and Dataset License

We plan to release our code and data under the MIT license to facilitate open and collaborative research. We have attached a zip file with the code to this submission.

13. Statement of Offensive Content and Personally Identification Information (PII)

We declare that our dataset has minimal risk of offensive content. The classes we choose for this dataset (e.g. airplane, car, truck..) are generally of a benign and non-offensive nature.

The images in our dataset were sourced from ImageNet. Therefore our dataset carries the same risks of PII as those in ImageNet, albeit restricted to the classes considered. For instance, although each selected class is not human-related, some images nevertheless contain images of humans. We could not verify that consent of these individuals to have their picture contained in a computer vision database. In future versions of the data, we plan to remove these images with face detectors.

No PII associated to Workers will be released.

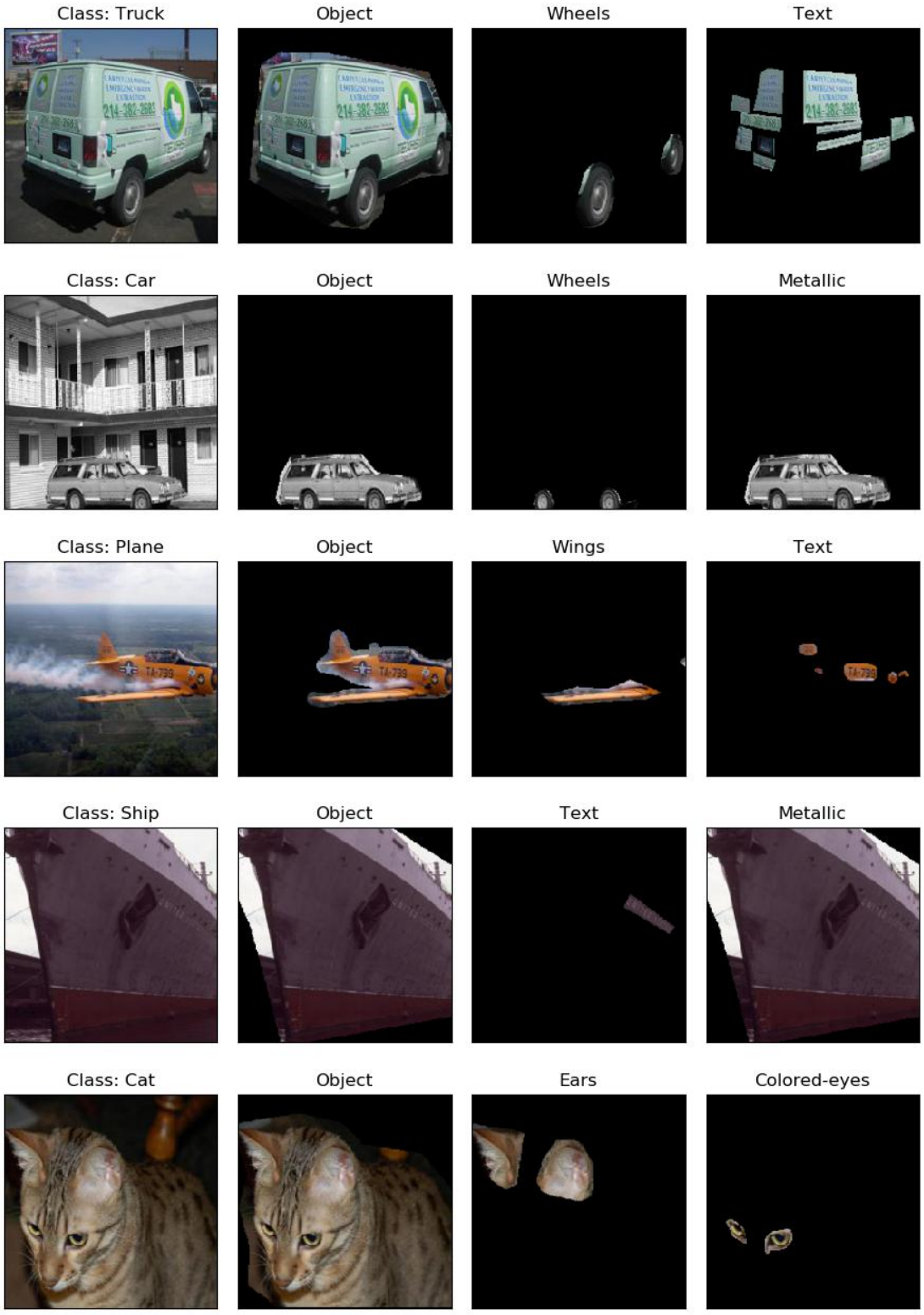

Figure 15. RIVAL10 examples. Left column has original image. Next column shows object mask applied onto the original image. The following two columns show attribute masks applied onto the original image.

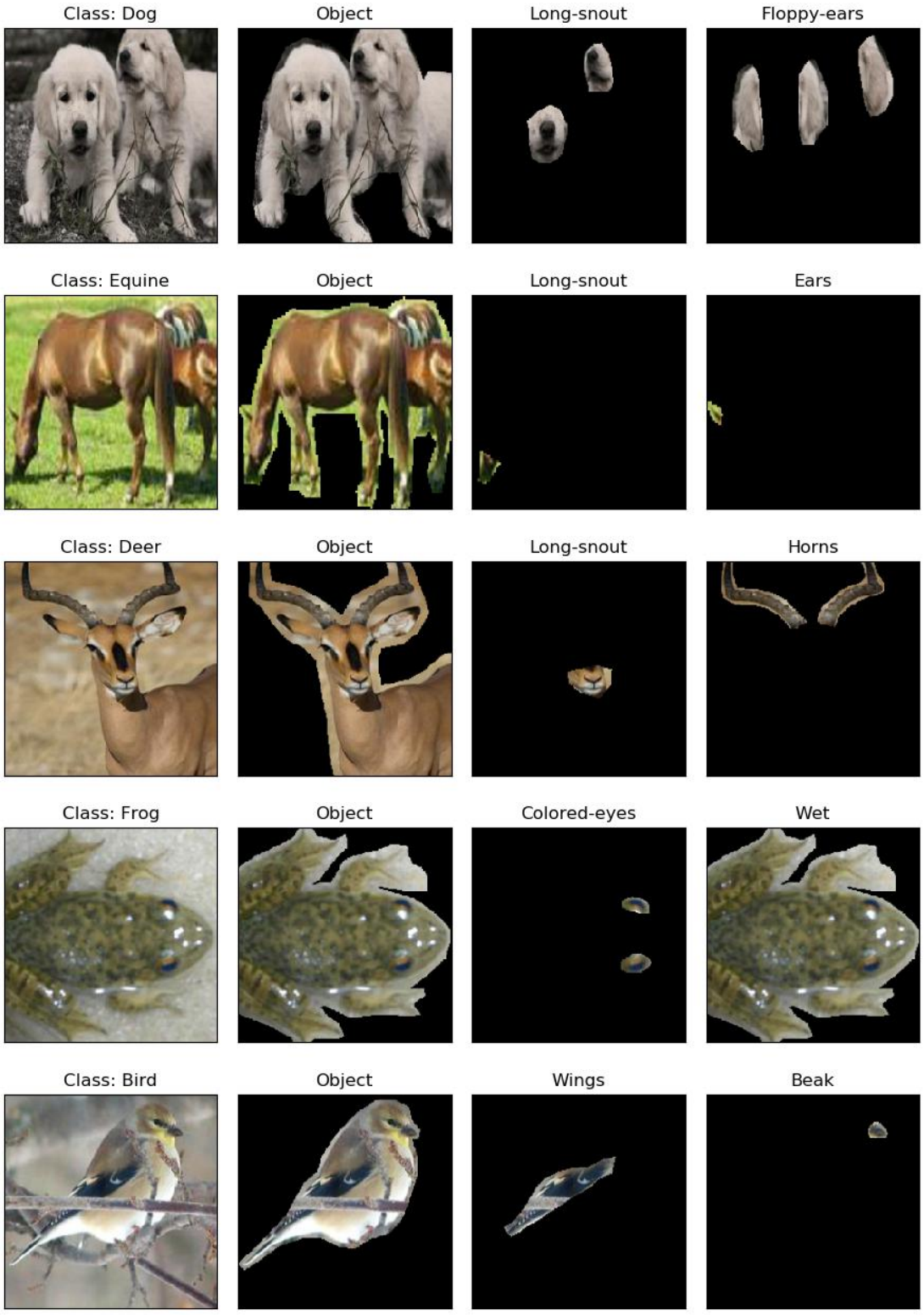

Figure 16. RIVAL10 examples. Left column has original image. Next column shows object mask applied onto the original image. The following two columns show attribute masks applied onto the original image.

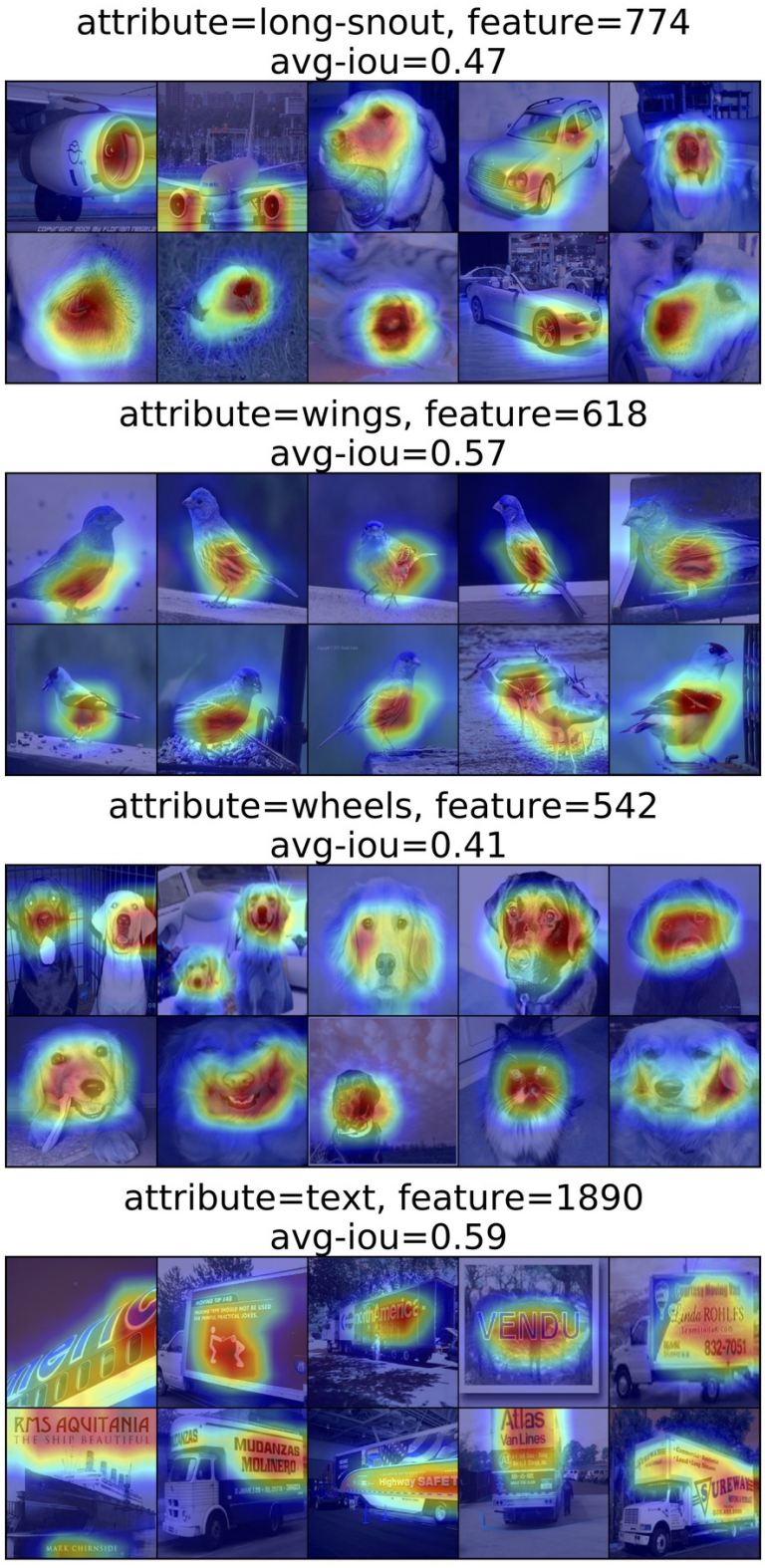

Figure 17. Saliency for top feature attribute pairs by IOU. First quarter of results shown here.

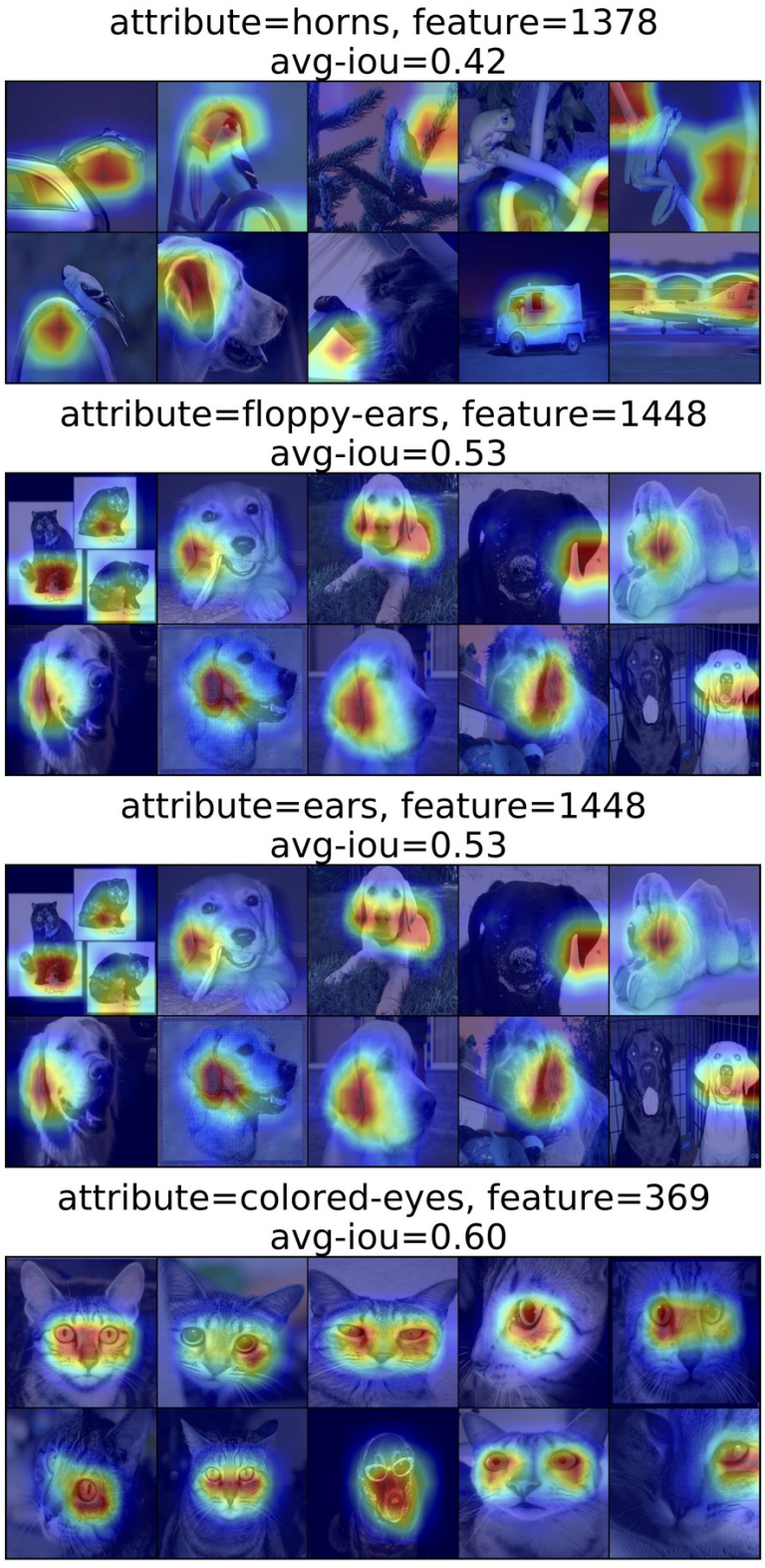

Figure 18. Saliency for top feature attribute pairs by IOU. Second quarter of results shown here.

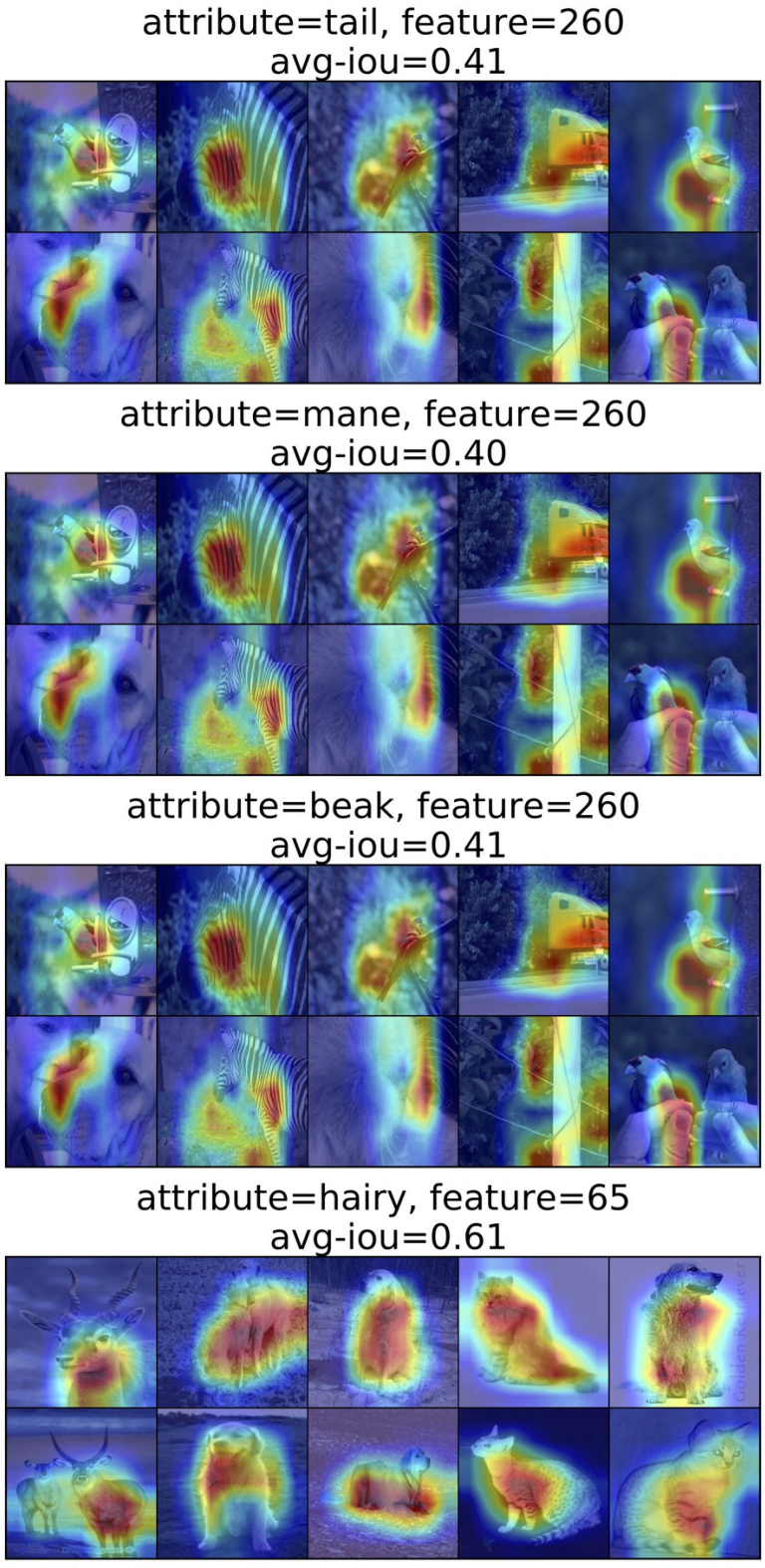

Figure 19. Saliency for top feature attribute pairs by IOU. Third quarter of results shown here.

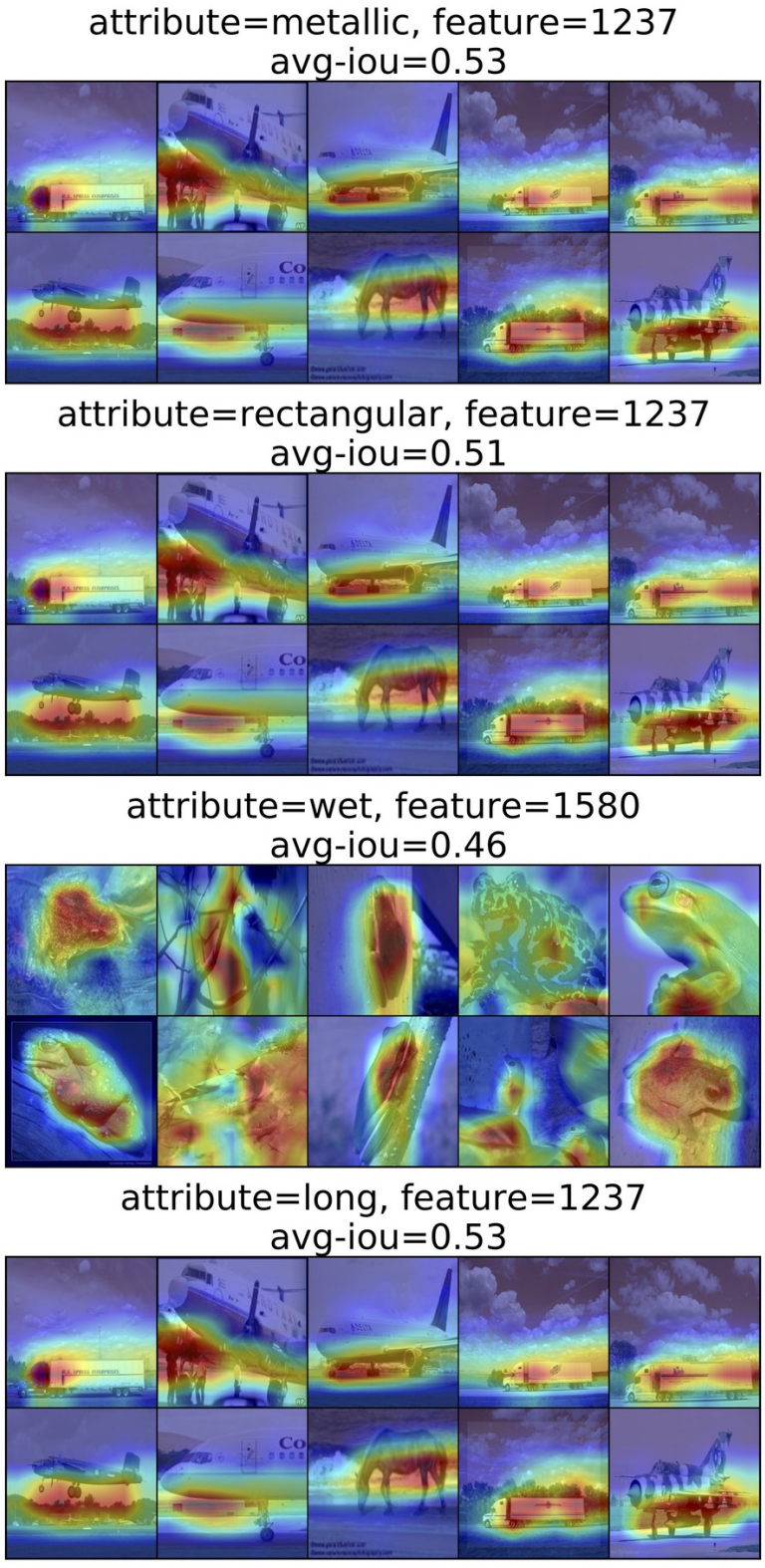

Figure 20. Saliency for top feature attribute pairs by IOU. Fourth quarter of results shown here.

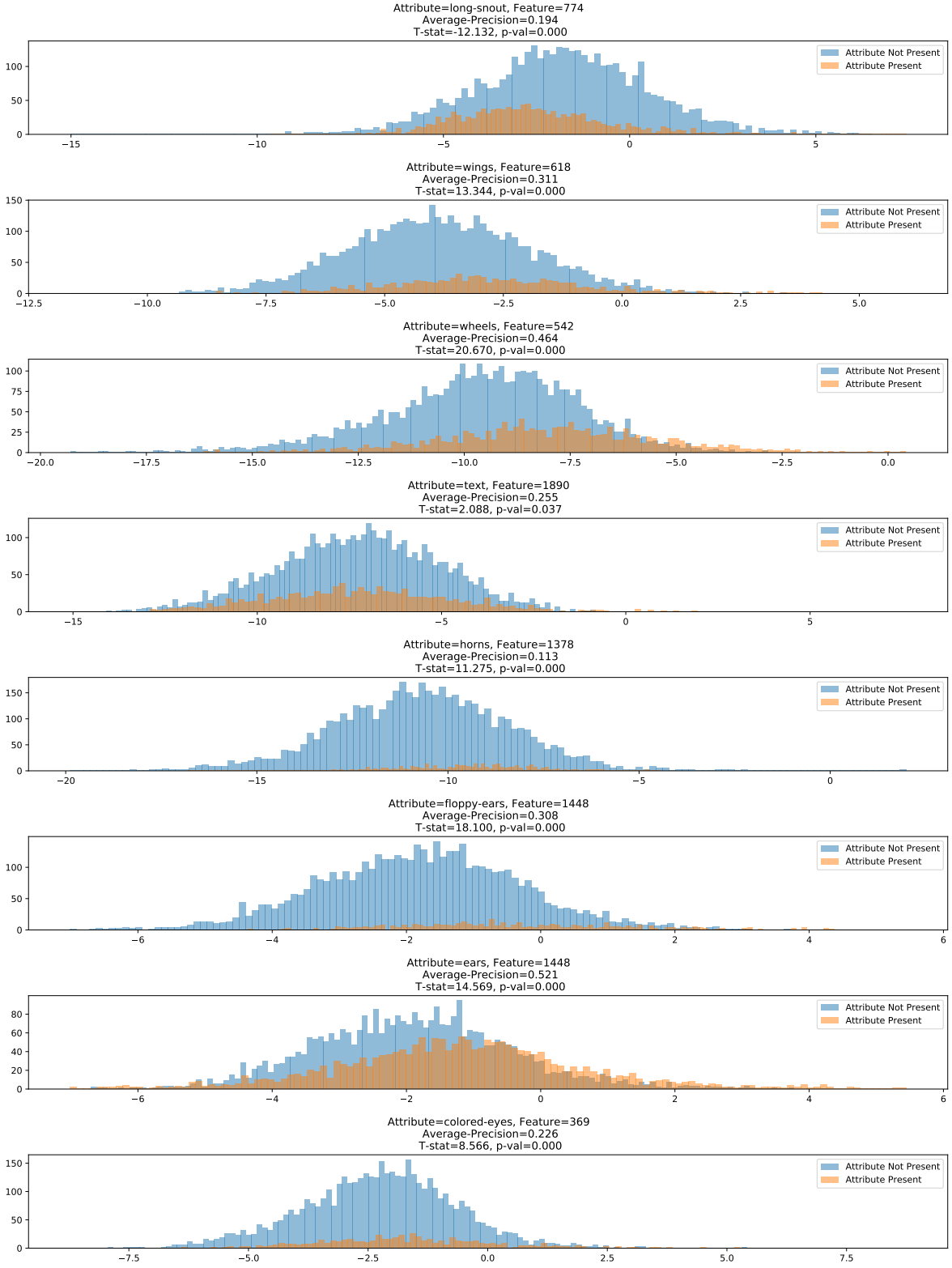

Figure 21. Top feature histograms for the top attribute feature pairs. First half shown here.

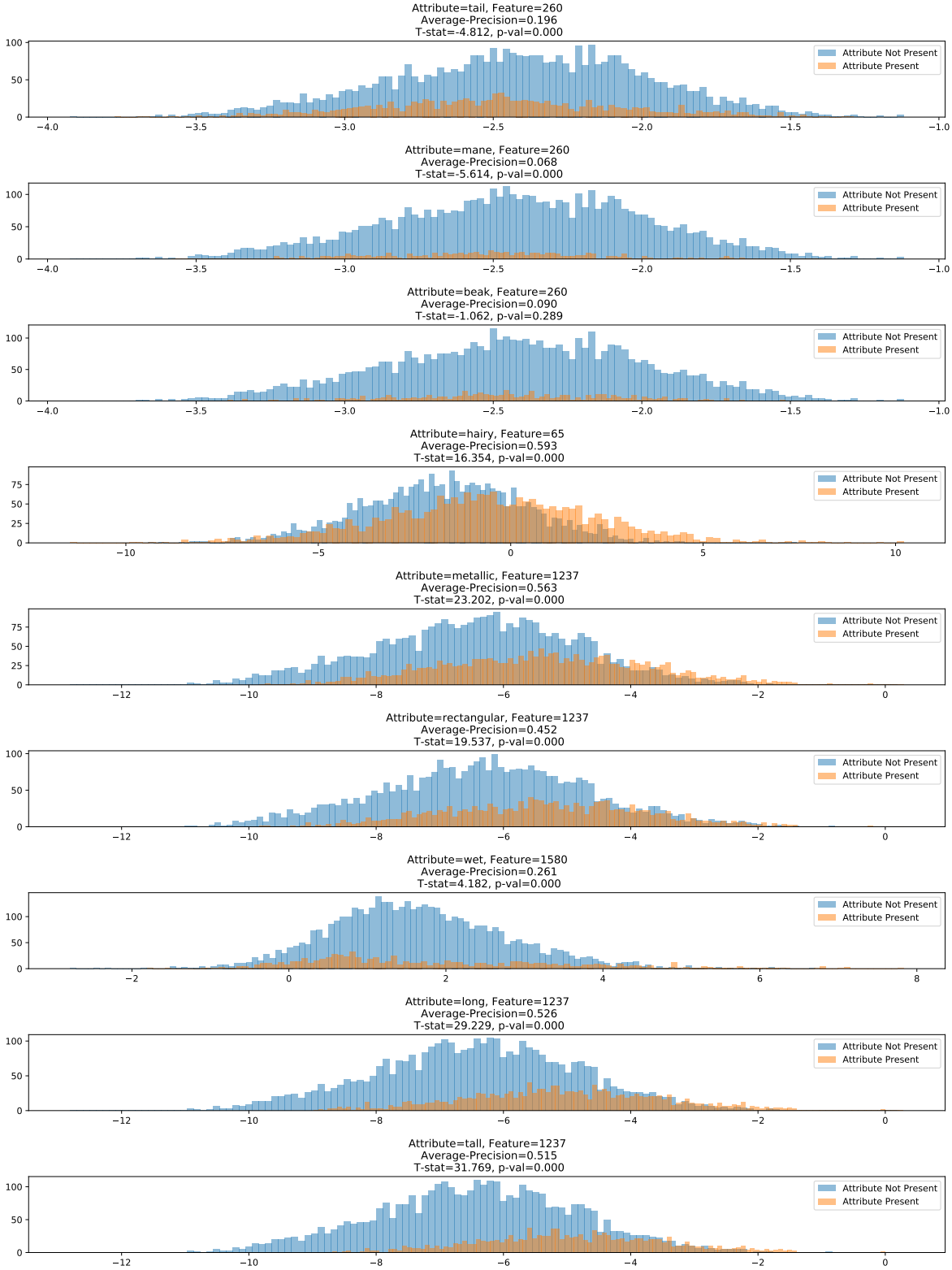

Figure 22. Top feature histograms for the top attribute feature pairs. Second half shown here.

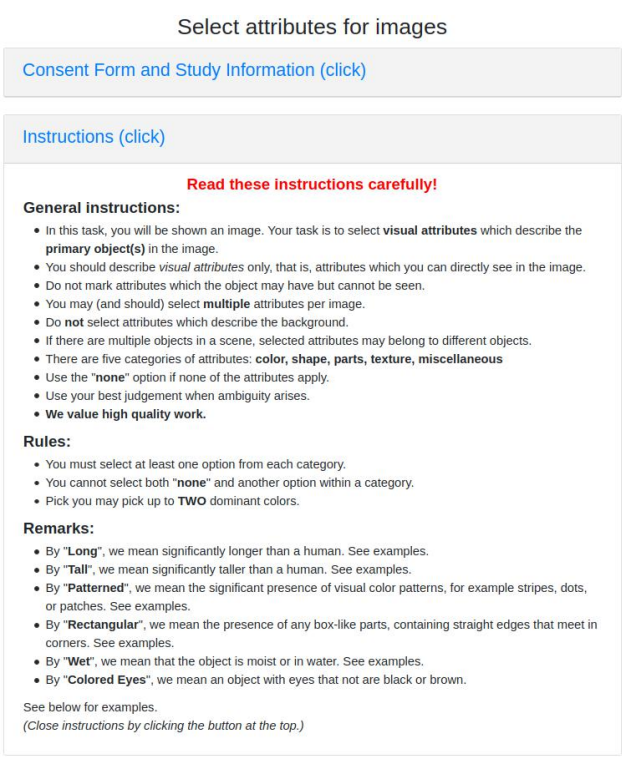

Figure 23. Screenshot of instructions page shown to workers

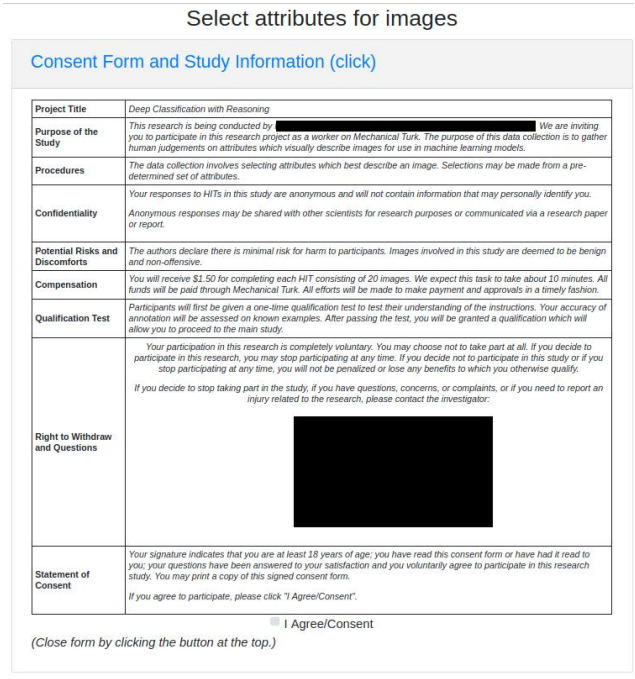

Figure 24. Screenshot of consent page shown to workers

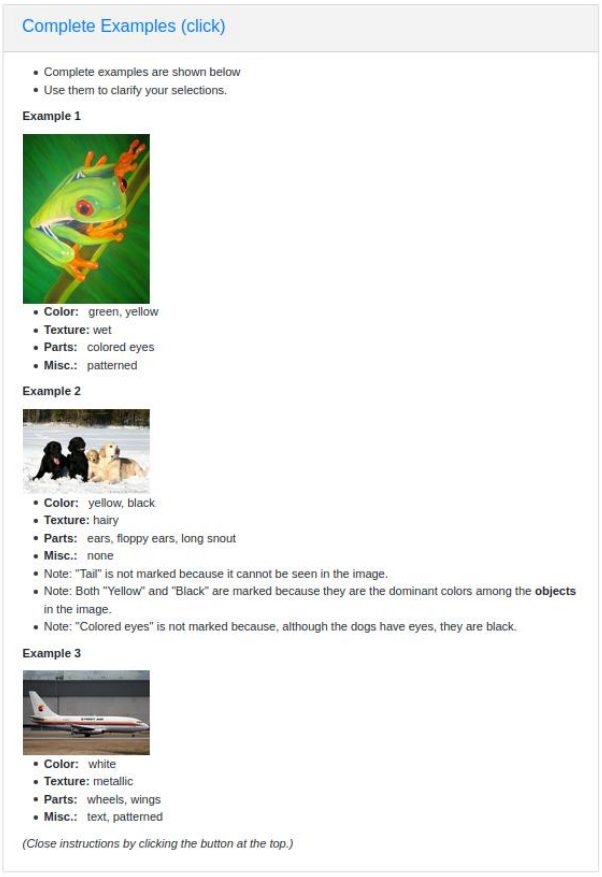

Figure 25. Screenshot of examples page shown to workers

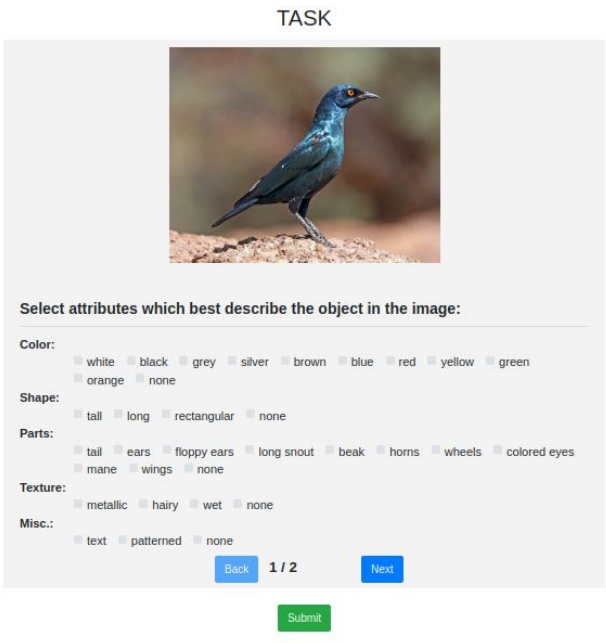

Figure 26. Screenshot of annotation form shown to workers

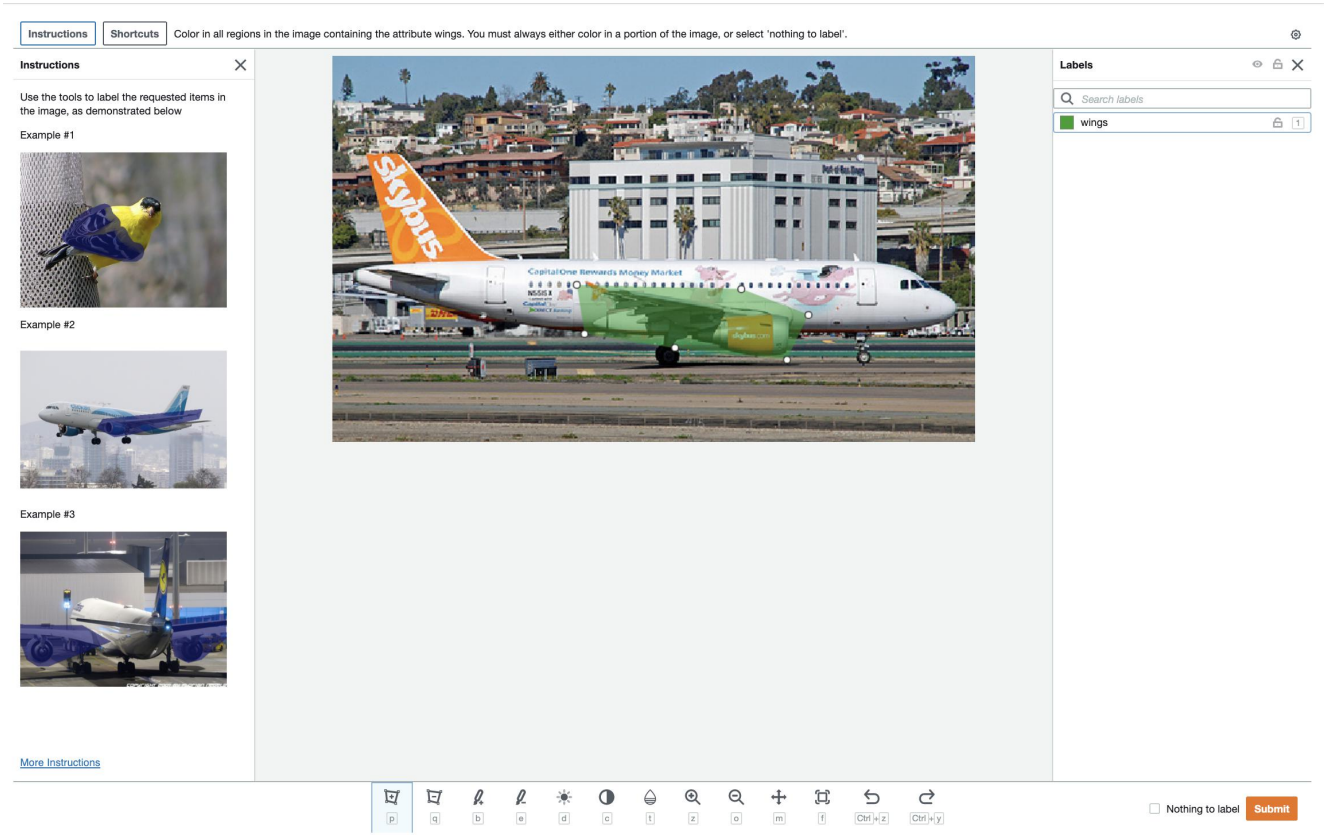

Figure 27. Screenshot of annotation form and tools for completing segmentations.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[3] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[4] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020.

[5] 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 (CVPR)*, June 2016.

[6] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[8] 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. *CoRR*, abs/2103.00020, 2021.

[9] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016.

[10] Kamil Szyc, Tomasz Walkowiak, and Henryk Maciejewski. Checking robustness of representations learned by deep neural networks. In Yuxiao Dong, Nicolas Kourtellis, Barbara Hammer, and Jose A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track*, pages 399–414, Cham, 2021. Springer International Publishing.

[11] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020.

[12] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.