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

Mazda Moayeri[1] mmoayeri@umd.edu
Phillip Pope[1] pepope@umd.edu
Yogesh Balaji[2] ybalaji@nvidia.com
Soheil Feizi[1] sfeizi@cs.umd.edu

[1] Department of Computer Science, University of Maryland
[2] NVIDIA

Abstract

While datasets with single-label supervision have propelled rapid advances in image classification, additional annotations are necessary in order to quantitatively assess how models make predictions. To this end, for a subset of ImageNet samples, we collect segmentation masks for the entire object and 18 *informative attributes. We call this dataset RIVAL10 (***RI***ch* ***V****isual* ***A****ttributes with* ***L****ocalization), consisting of roughly* $26k$ *instances over* 10 *classes. Using RIVAL10, we evaluate the sensitivity of a broad set of models to noise corruptions in foregrounds, backgrounds and attributes. In our analysis, we consider diverse state-of-the-art architectures (ResNets, Transformers) and training procedures (CLIP, SimCLR, DeiT, Adversarial Training). We find that, somewhat surprisingly, in ResNets, adversarial training makes models more sensitive to the background compared to foreground than standard training. Similarly, contrastively-trained models also have lower relative foreground sensitivity in both transformers and ResNets. Lastly, we observe intriguing adaptive abilities of transformers to increase relative foreground sensitivity as corruption level increases. Using saliency methods, we automatically discover spurious features that drive the background sensitivity of models and assess alignment of saliency maps with foregrounds. Finally, we quantitatively study the attribution problem for neural features by comparing feature saliency with ground-truth localization of semantic attributes.*

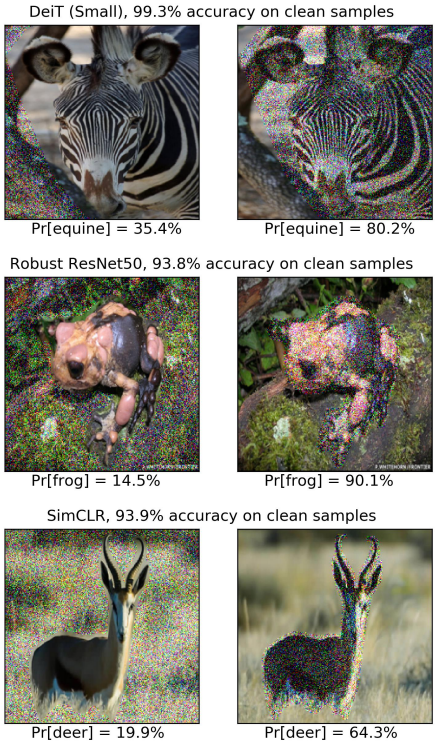


Figure 1. Examples where background noise degrades performance of highly accurate models more than foreground noise. Gaussian ℓ_∞ noise with standard deviation $\sigma = 0.24$ shown. Probabilities are averaged over ten trials. While these examples are cherry picked, we observe that they are surprisingly prevalent, and model design can affect the degree to which such cases arise.

1. Introduction

Large scale benchmark datasets like ImageNet [9] that were constructed with single class label annotation have propelled rapid advances in the image classification task [20, 23, 53, 61]. Over the last decade, several network architectures and training procedures were proposed to yield very high classification accuracies [10, 20, 47, 53]. However, methods to interpret these model predictions and to

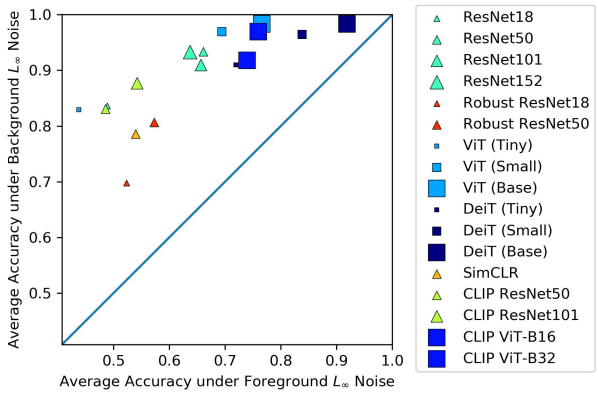


Figure 2. Accuracy under noise averaged over multiple noise levels. Marker size is proportional to parameter count. Models with higher relative foreground sensitivity lie further from the diagonal.

diagnose undesirable behaviors are fairly limited. One of the most popular class of approaches are saliency methods [45, 50, 51, 62] that use model gradients to produce a saliency map corresponding to the most influential input regions that yielded the resulting prediction. However, these methods are qualitative, require human supervision, and can be noisy, thus making their judgements potentially unreliable when made in isolation of other supporting analysis.

In this paper, we argue that to obtain a proper understanding of how specific input regions impact the prediction, we need additional ground truth annotations beyond a single class label. To this end, we introduce a novel dataset, RIVAL10, whose samples include **RI**ch **V**isual **A**ttributions with **L**ocalization. RIVAL10 consists of images from 20 categories of ImageNet-1k [9], with a total of $26k$ high resolution images organized into 10 classes, matching those of CIFAR10 [28]. The main contribution of our dataset is instance wise labels for 18 informative visual attributes, as well as segmentation masks for each attribute and the entire object. We present our dataset as a general resource for understanding models trained on ImageNet. We then provide a study of the sensitivity of a diverse set of models to foregrounds, backgrounds, and attributes.

Our study of background and foreground model sensitivity is motivated by some counter-intuitive model behaviors on images whose background and foreground regions were corrupted with Gaussian noise: Figure 1 shows instances where highly accurate models have performance degraded much more due to the background noise than the foreground noise. While this is not the norm (i.e. models are more sensitive to foregrounds on average), the existence of these examples warrants greater investigation, as they expose a stark difference in how deep models and humans perform object recognition. Quantifying the degree to which different architectures and training procedures admit these examples can shed new insight on how models incorporate foreground and background information.

To this end, we conduct a *noise analysis* that leverages object segmentation masks to quantitatively assess model sensitivity to foregrounds relative to backgrounds. We proxy sensitivity to a region by observing model performance under corruption of that region. We propose a normalized metric, *relative foreground sensitivity* (RFS), to compare models with various general noise robustness. A high RFS value indicates that the model uses foreground features in its inferences more than background ones since corrupting them result in higher performance degradation.

In Figure 2, we see different architectures and training procedures lead to variations in both general noise robustness (projection onto the main diagonal) and relative foreground sensitivity (normalized distance orthogonal to the diagonal). Notably, we find that adversarially training ResNets significantly reduces RFS, surprisingly suggesting that robust ResNet models make greater use of background information. We also observe contrastive training to reduce RFS, and transformers to uniquely be able to adjust RFS across noise levels, reducing their sensitivity to backgrounds as corruption level increases. Lastly, we find object classes strongly affect RFS across models.

We couple our noise analysis with saliency methods to add a second perspective of model sensitivity to different input regions. Using RIVAL10 segmentations, we can quantitatively assess the alignment of saliency maps to foregrounds. We also show how we can discover spurious background features by sorting images based on the saliency alignment scores. We observe that performance trends that our noise analysis reveals are not captured using qualitative saliency methods alone, suggesting our noise analysis can provide new insights on model sensitivity to foregrounds and backgrounds.

Lastly, we utilize RIVAL10 attribute segmentations to systematically investigate the generalizability of neural feature attribution: for a neural feature (i.e., a neuron in the penultimate layer of the network) that achieves the highest intersection-over-union (IOU) score with a specific attribute mask on top-k images within a class, how the IOU scores of that neural feature behave on other samples in that class. For some class-attribute pairs (e.g. dog, floppy-ears), we indeed observe generalizability of neural feature attributions, in the sense that test set IOUs are also high.

In summary, we present a novel dataset with rich annotations of object and attribute segmentation masks that can be used for a myriad of applications including model interpretability. We then present a study involving three quantitative methods to analyze the sensitivity of models to different regions in inputs. We hope that our richly annotated RIVAL10 dataset helps studying failure modes of current deep classifiers and paves the way for building more reli-

able models in the future.

2. Review of Literature

2.1. Related Datasets

Prior to the rise of deep learning, a number of works studied attribute classification, leading to the construction of datasets such as "Animals with Attributes" [29] and aPASCAL VOC 2008 [16] (adding annotations to [14]). [57] published the Caltech-UCSD Birds 200 (CUB), a fine-grained classification datasets of bird species with object segmentations and part *localizations* in the form of single coordinates as opposed to segmentation masks like in RIVAL10. Finally, [43] collected object attributes on a small-scale subset of ImageNet. More recently, [37] publish a large-scale object attribute dataset on a subset of ImageNet. The Celeb-A dataset from [31] contains attribution and has applications to generative modeling, but limited utility for general representation learning since it only contains face images. A broader dataset is Visual Attributes in the Wild (VAW) [40], which provides large-scale in the wild attribute annotations for 250k object instances.

Many datasets aim to stress test models to reveal limitations. [21] introduces ImageNet variants under diverse corruption types, including Gaussian noise. [22] adds two more ImageNet variants that include challenging natural samples and out of distribution samples, on which top models see massive accuracy drops. Models evaluated on [2] similarly see large drops, though this dataset differs in that it is strictly a test set. Other works introduce datasets related to the task of assessing background reliance of classifiers, such as [60] and [44], which perform some variation of swapping or altering foregrounds and backgrounds. Though similar, these works differ in objective and technical contribution to ours. [44] focuses on developing a novel distributionally robust optimization procedure. [60] emphasizes designing a multitude of test datasets through creative editing of foreground and background regions to serve as a general benchmark to evaluate models. In contrast, our work presents a novel method to analyze relative foreground sensitivity, and demonstrates its utility by applying it to a breadth of diverse, cutting edge models, engaging different architectures and training paradigms in a comprehensive fashion, leading to *model-specific* observations. Further, our RIVAL10 dataset is significantly larger and richer in annotation.

Recently, [48] uses saliency maps and feature visualization in a semi-automated process to identify deep neural nodes corresponding to core or spurious features for an object of a given class, resulting in a large-scale dataset with segmentations corresponding to salient features. However, annotation of the segmented regions are limited to just labeling them as 'core' or 'spurious'.

2.2. Interpretability Methods

A number of methods have been proposed to interpret model predictions, such as saliency or class activation maps [45], influence functions [27], and surrogate white box models [42, 59]. However, saliency maps have been found to be noisy and influence functions are fragile [3, 18]. Certain methods seek to interpret the functions of neural nodes via synthesizing inputs that maximize the activation of the node [34, 36, 49], though these methods are limited when non-adversarially robust models are used [35], and largely offer qualitative insights.

A motivation behind the development of interpretability methods is to work towards addressing the 'shortcut learning' issue, where models rely on easy-to-learn features that lead to high performance on training sets, but poor generalization in other settings. [17] discusses this at length, recommending the development and usage of challenging datasets whose inputs are out-of-distribution with respect to standard benchmarks. RIVAL10's rich annotations open the door to the construction of many challenge datasets, in which shortcuts are broken via swapping backgrounds, foregrounds, and *attributes*. We show examples of such crafted images in the appendix.

Other constructive works aimed to reduce the reliance of deep models on spurious features appeal to counterfactual data generation [1, 6, 19], often appealing to disentangled representations or explicit annotations to break correlations of texture, shapes, colors, and backgrounds. Further, [25] found that removing spurious features can in fact hurt accuracy and disproportionately affect groups. Thus, the notion that spurious features are always harmful is incomplete, and a closer look is required to ground discussions regarding the shortcut learning issue. Lastly, [55] provides theoretical context for stress testing models to discern causal factors.

3. RIVAL10

3.1. Overview

RIVAL10 differs from previous attributed datasets in that it provides *attribute-specific* localizations. That is, for every positive instance of an attribute, a polygonal segmentation mask (possibly multiple parts) is provided to identify the image region in which the attribute occurs.

Perhaps, the most similar dataset in this regard is the recent Fashionpedia [24], a dataset providing attributes and localizations of 27 apparel categories. However, the dataset is proposed for the fashion domain which limits its utility for general purpose object recognition task. To the best of our knowledge, RIVAL10 is the first *general domain* dataset to provide both rich semantic attributes and localization, the combination of which we envision to aid in analyzing the robustness and interpretability of deep networks. While other datasets used for semantic

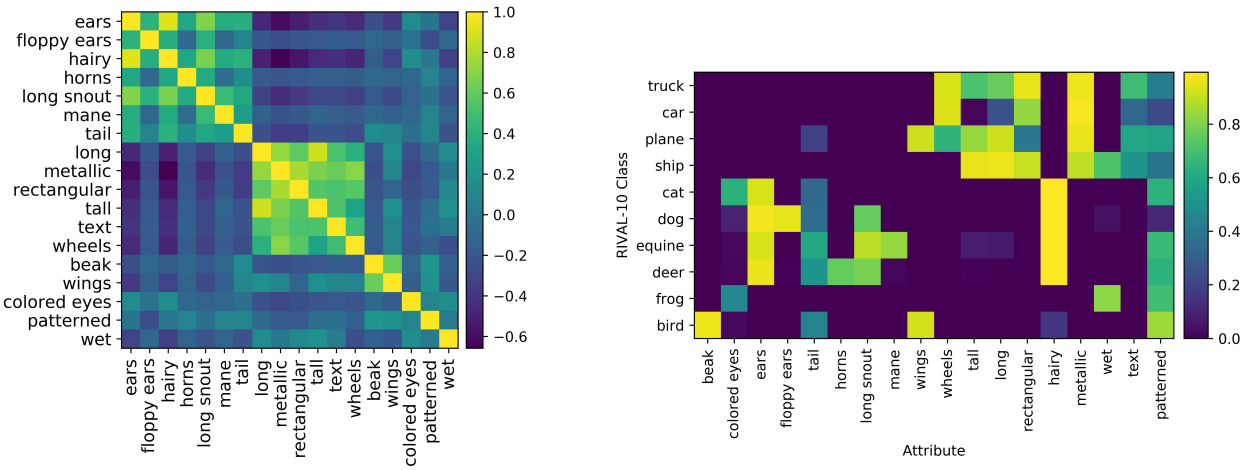


Figure 3. (**Left**): Correlations between attributes in the training split. (**Right**): Class-wise means of attribute vectors in the training split.

segmentation and object detection go beyond single label annotations [8, 13, 30], they are not designed with classifiers specifically in mind, like RIVAL10.

Classes were chosen to be aligned with CIFAR-10 to enable analyzing the existing architectures and training techniques developed for the object recognition task. In particular, the classes we provide are: *bird, car, cat, deer, dog, equine, frog, plane, ship, truck*. We collected the following attributes for these object categories: *beak, colored-eyes, ears, floppy-ears, hairy, horns, long, long-snout, mane, metallic, patterned, rectangular, tail, tall, text, wet, wheels, wings*. Some attributes were inspired from [43].

We chose attributes to be intuitively informative, capturing semantic concepts that humans may allude to in classifying RIVAL10 objects. While the attributes contain some redundant information, they are nonetheless discriminative in the sense that a linear classifier on attributes achieves 93.3% test accuracy. We visualize attribute correlations and class-wise averages in Figure 3.

3.2. Data Collection

All images were sourced from ImageNet [9]. The images used in each RIVAL10 class were derived from pairs of related ImageNet classes. In other words, 20 classes from Imagenet were used to build the 10 RIVAL10 classes (details in appendix). To collect our attributes and localizations, we hired workers from Amazon Mechanical Turk (AMT). Data collected through AMT without careful control may be of low quality. To encourage quality annotations, we utilize strategies recommended by the HCI community [33]: providing detailed instructions, *screening* workers for aptitude, and monitoring worker performance with attention checks.

Binary attributions were collected first. Workers were required to pass a qualification test of 20 images with known ground truth attributes: only workers who achieved a minimal overall precision and recall of 0.75 were hired for full data collection. Because the task of segmentation is more involved than indicating whether or not an attribute is present, we required a second qualification test, assessing annotation quality by computing intersection-over-union (IOU) of the submitted attribute masks with ground truth masks. Workers were required to complete five segmentations with an average IOU of at least 0.7. This strategy encourages selection of workers who have demonstratively read and understood the instructions.

To ensure that quality is maintained in both the attribution and segmentation phases, roughly 5% of images provided to workers to annotate already had ground truth labels. These so-called attention checks allowed for the monitoring of annotation quality during the collection process. In the first stage of collecting binary attribute labels, the average precision and recall scores were 0.81 and 0.84 respectively. For each positive instance of an attribute marked in the first phase of data collection, an attribute segmentation was collected in the second phase. Completeing attribute segmentations in a second pass allowed for the review of the binary attributions and the removal of any false positives. Average IOU of attention checks completed during the second phase of data collection was 0.745.

Further details about our data collection pipeline, including images of our instructions shown to workers, histograms of scores on the qualifying exam and attention checks, selection process of the AMT workers, payments and other details can be found in the appendix.

4. Models

In our analyses, we focus on ResNets and Vision Transformers [10, 20]. We inspect ResNets trained (i) in a standard supervised fashion, (ii) adversarially via ℓ_2 projected gradient descent [32], and (iii) contrastively (i.e. no direct label supervision), with SimCLR and CLIP [7, 41]. We also

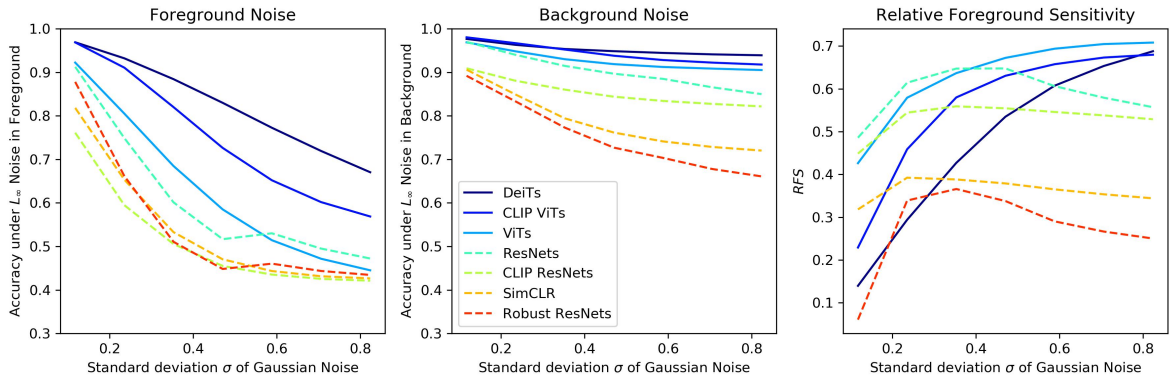


Figure 4. Accuracy under noise in foreground (**left**) and background (**middle**) at various noise levels. Models are grouped by architecture and training procedure, with a curve corresponding to the average over all models in a group. (**Right**): RFS by group.

consider CLIP Vision Transformers, as well as standard Vision Transformers (ViT) and Data efficient Image Transformers (DeiT) [54]. DeiTs differ from ViTs primarily in their training set, solely using ImageNet-1k while ViTs used ImageNet-21k. To make up for not having the inductive biases of ResNets, ViTs increased the amount of training data, while DeiTs instead rely upon extensive augmentation. All other models, with the exception of those trained with CLIP, use ImageNet-1k as the pretraining set. CLIP, on the other hand, uses a much larger dataset of images and associated text. A full discussion on models is offered in the appendix.

To perform classification on RIVAL10 dataset, we attach a linear head to the features extracted from the penultimate layer of the base models. All base models have their weights frozen (from pretraining) and are not updated during fine-tuning. This preserves the feature space learned in the original pretraining. All models achieve upwards of 90% accuracy on the RIVAL10 test set, essentially controlling for classification ability. We note that while there is leakage between ImageNet-1k and the RIVAL10 test set, the purpose of this study is not to improve model's predictive accuracy directly, but instead to better understand the information used in making predictions.

Recently, a number of works compare the robustness of ViTs to ResNets. While there are mixed findings on adversarial robustness [4, 46], there is agreement that ViTs have stronger out-of-distribution generalization, likely due to self attention [5, 39]. In contrast, our work focuses on relative robustness to noise in foreground and background regions.

5. Foreground and Background Sensitivity

5.1. Noise Analysis

We add noise to the foreground and background separately to see how corrupting each region degrades model performance. Consider a sample $\mathbf{x}$ with a binary object mask $\mathbf{m}$ where $\mathbf{m}_{i,j} = 1$ if the pixel $\mathbf{x}_{i,j}$ is a part of the object. We first construct a noise tensor $\mathbf{n}$ that has pixel values drawn i.i.d. from $\mathcal{N}(0, \sigma^2)$, where σ is a parameter controlling the noise level. Then, we obtain noisy-background $\tilde{\mathbf{x}}_{bg}$ and noisy-foreground $\tilde{\mathbf{x}}_{fg}$ samples as:

$$\tilde{\mathbf{x}}_{fg} = \text{clip}(\mathbf{x} + \mathbf{n} \odot \mathbf{m}),\ \tilde{\mathbf{x}}_{bg} = \text{clip}(\mathbf{x} + \mathbf{n} \odot (\mathbf{1} - \mathbf{m}))$$

where $\odot$ is the hadamard product, and 'clip' refers to clipping all pixel values to the $[0, 1]$ range. We add Gaussian noise so to preserve the image content. Note that additive pixel-wise noise leads to the same magnitude of perturbation in the foreground and background under the ℓ_∞ norm. We also repeat our analysis with ℓ_2 normalized noise (presented in the appendix) to avoid a bias against larger regions and obtain similar results.

We seek to quantify the sensitivity of a model to foregrounds relative to its sensitivity to backgrounds. To this end, we introduce *relative foreground sensitivity* (RFS). Let a_{fg} and a_{bg} denote accuracy under noise in the foreground and background, respectively, and $\bar{a} := (a_{fg} + a_{bg})/2$ denote their mean (referred to as the general noise robustness). We then define RFS for a model $\mathbf{F}$ as

$$RFS(\mathbf{F}) = \frac{a_{bg} - a_{fg}}{2\min(\bar{a}, 1 - \bar{a})}.$$

Essentially, RFS normalizes the gap in model performance under foreground and background noise by the total possible gap, given the general noise robustness of the model. In Figure 2, RFS takes on the geometric meaning of the ratio between the distance of (a_{fg}, a_{bg}) to $(\bar{a}, \bar{a})$, to the largest possible distance from the diagonal in the unit square for a point with general noise robustness $\bar{a}$. The scale factor in the denominator gives RFS a range of $[-1, 1]$, with larger values corresponding to greater relative foreground sensitivity.

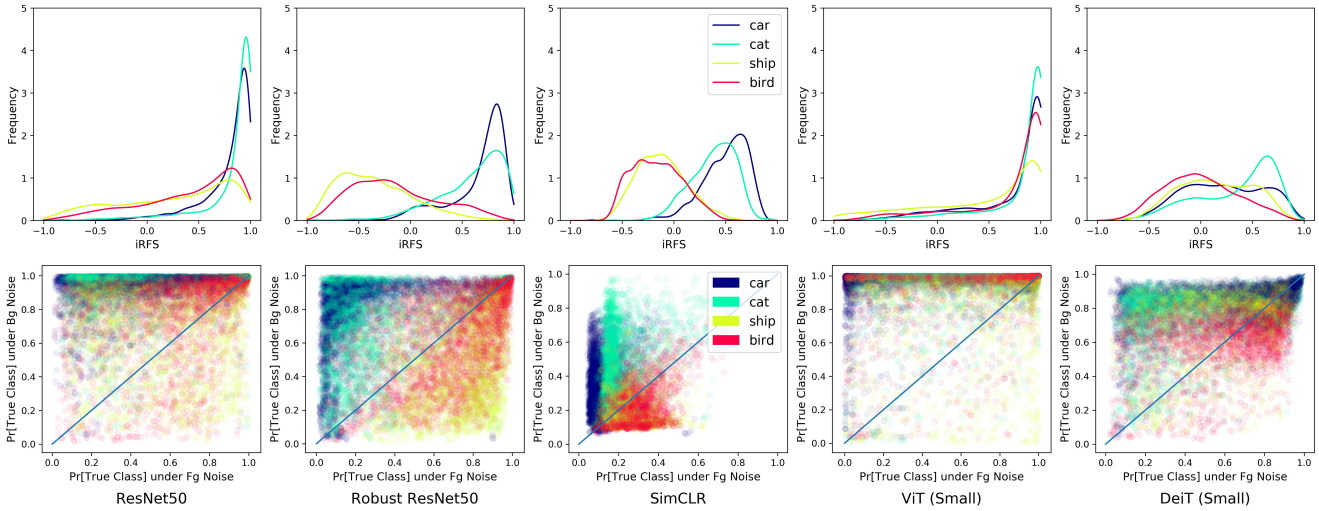


Figure 5. Relative foreground sensitivity per instance for four classes and five models of roughly equal size. (**Top**): Histogram of $iRFS$; positive denotes greater foreground sensitivity. (**Bottom**): Scatter; top left indicates high relative foreground sensitivity. Across models, ships and birds have low foreground sensitivity, often being more sensitive to noise in the background than the foreground.

We also consider an instance-wise version, $iRFS$, defined for a model $\mathbf{F}$ *and* a sample $\mathbf{x}$. Here, we use the probability that model $\mathbf{F}$ predicts sample $\mathbf{x}$ to belong to its true class as the measure of model performance instead of accuracy. Let p_{fg} and p_{bg} denote this probability for $\tilde{\mathbf{x}}_{fg}$ and $\tilde{\mathbf{x}}_{bg}$, respectively. Thus, with $\bar{p} := (p_{fg} + p_{bg})/2$,

$$iRFS(\mathbf{F}, \mathbf{x}) = \frac{p_{bg} - p_{fg}}{2\min(\bar{p}, 1 - \bar{p})}.$$

In our experiments, we consider seven equally spaced noise levels from $\sigma = 30/255$ to $210/255$. For each sample in the test set, we take ten trials of adding noise to the foreground and background separately *per noise level*. RIVAL10's test set consists of roughly $5k$ images, so for each model type, we assess $5k \times 7 \times 10 = 350,000$ trials in total.

5.2. Empirical Observations

Fig. 2 shows different models have vastly different performance in terms of both general noise robustness and relative foreground sensitivity. In Figure 2, transformers generally lie further up the main diagonal than ResNets, corroborating observations that transformers are more robust to common corruptions. Increasing model size improves general robustness, though it does more so for transformers than ResNets. Models lie at different distances orthogonal to the diagonal as well, indicating architecture and training procedure affect relative foreground sensitivity.

In Figure 4, we categorize model types based on architecture and training procedure, averaging RFS over groups to reveal general trends. Robust ResNets have the lowest RFS, much lower than standard ResNets, a somewhat surprising result given that background reliance has been thought to be linked to increased adversarial vulnerability in the past [56,60]. SimCLR has the next lowest RFS overall, and generally, contrastive training procedures (CLIP, SimCLR) seem to reduce RFS in both ResNets and ViTs.

In comparing transformers to ResNets overall, we see at low noise levels, transformers sometimes have lower RFS than ResNets. Interestingly, as noise level rises, RFS in transformers increases as well, while RFS is mostly stable for ResNets. This suggests that transformers can adaptively alter the attention paid to different image regions based on the level of corruption. Comparing between transformers, we see DeiTs with much lower RFS than ViTs, suggesting that the heavy augmentations DeiTs leveraged to achieve increased data efficiency may have also made the models much more sensitive to backgrounds.

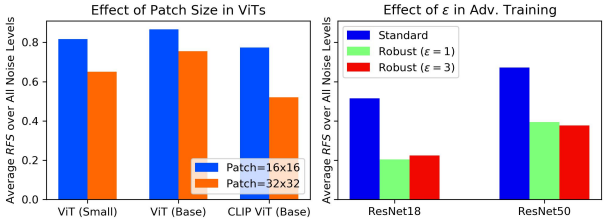


Figure 6. Controlled ablation studies. Average RFS over all noise levels presented for brevity. (**Left**): Increasing patch size in ViTs decreases relative foreground sensitivity. (**Right**) Robust models are much less relatively sensitive to foregrounds, but ϵ used in adversarial training does not affect RFS much.

In Figure 6, we more closely inspect the effect of patch sizes in ViTs and the attack budget ϵ used in adversarial training (which affects accuracy-robustness trade-off). We find that increasing the patch size in ViTs from 16×16 to

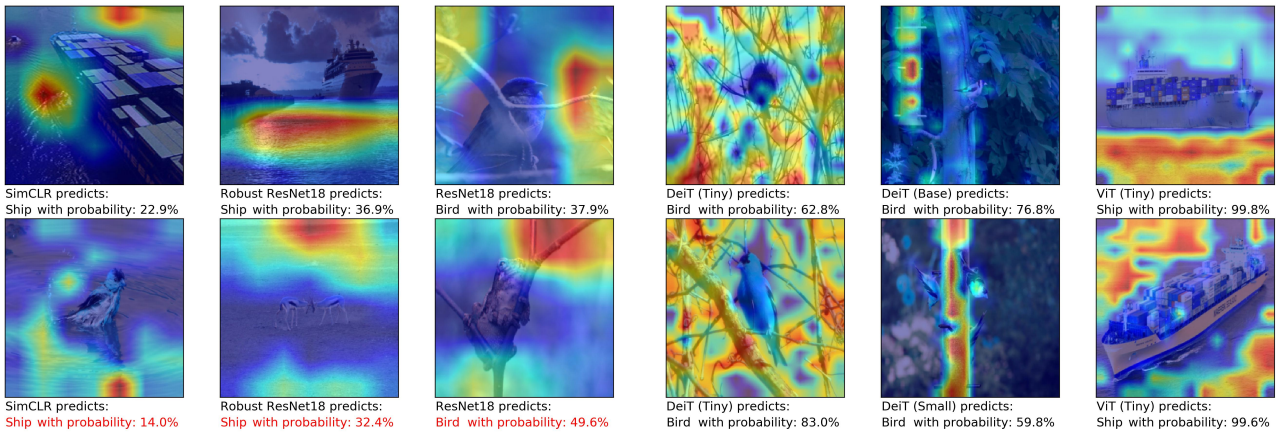


Figure 7. Instances of images with low saliency alignment, highlighting spurious features of water for ships, and branches and bird feeders for birds. (**Left**): Spurious features leading to misclassification (in red). (**Right**): Other instances of spurious features.

32×32 reduces RFS when averaged over all noise levels. The robustness ablation affirms that robust ResNets are much less relatively sensitive to foregrounds than standard ResNets, though the attack size seen in training does not seem to significantly affect RFS.

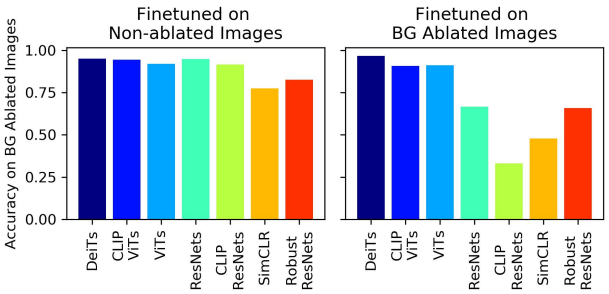


Figure 8. Model accuracy on images with backgrounds ablated via graying. The right plot shows accuracies for models finetuned on the images with backgrounds ablated. Only transformers can fit a linear layer on the features of background ablated images without compromising performance.

Moving away from comparing models, in Figure 5, we see **foreground sensitivity is largely affected by class**. In particular, across models of roughly equal size, ships and cats are often more sensitive to background noise, suggesting models learn to utilize background content more than foreground content in recognizing them. The class distinction is less pronounced in DeiTs and ViTs, with ViTs assigning high foreground sensitivity for all classes, and DeiTS having mixed sensitivity across classes, with many $iRFS$ scores larger than 0.

5.3. Removing Backgrounds Entirely

We also inspect the accuracy of models on images with backgrounds grayed out, similar to [60], though now considering ViTs, CLIP, and SimCLR, which had not been developed at the time of their study. Also, the rich annotations of RIVAL10 allow for going beyond foreground or background ablation (see the appendix for a discussion of attribute removal). Ablation via graying can be thought of as another kind of noise, where all pixels are smoothed to 0.5. In Figure 8, the left plot reveals that Robust ResNets and SimCLR see the largest drops in accuracy when evaluated on images with grayed backgrounds. Transformers do well on ablated images, consistent with the observation that transformers had high RFS at the largest noise levels. Furthermore, when we attempt to fit a linear layer to classify background-ablated images, only the features from transformer models are sufficiently informative to have high linear classification accuracy. Thus, while transformers make use of backgrounds, they still retain significant foreground information in their feature space. This result suggests transformers are much more robust to localized distribution shift. That is, distribution shift in one region (the background) may affect model perception of other unperturbed regions much less in transformers than ResNets.

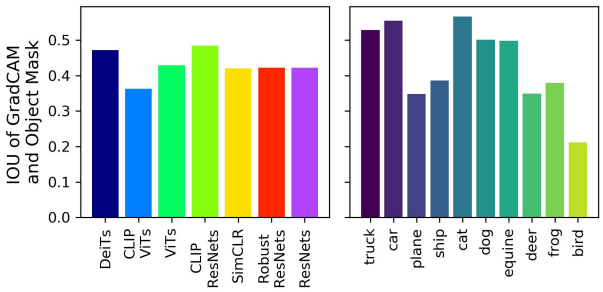


Figure 9. Alignment of binarized saliency maps with object segmentation masks, measured by intersection over union (IOU). Averaged over models (**left**) and object classes (**right**).

5.4. Saliency Alignment

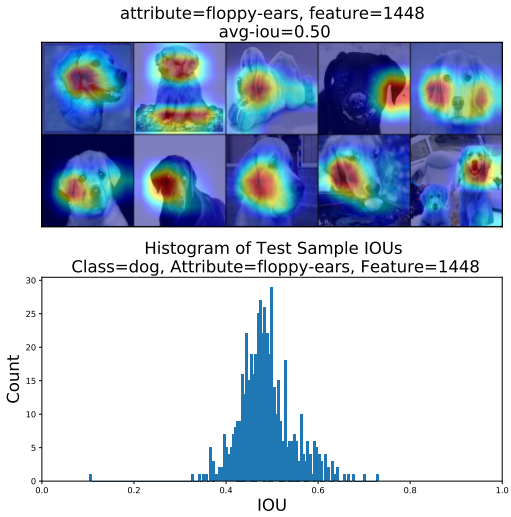


Figure 10. (**Top**): Example GradCAMs on test images with respect to the top feature identified by IOU in training set. (**Bottom**): Histograms of IOUs corresponding to this feature, attribute pair.

To complement the noise analysis, we use GradCAM [45] to assess the amount of saliency that models place on foreground pixels. RIVAL10's object segmentations allow us to automatically quantify saliency alignment with foregrounds, removing the need for human inspections. We also can easily detect failure modes, where models deem background regions as highly salient, by sorting samples based on saliency alignment. We present several metrics to assess saliency alignment in the appendix. We find that extracting samples with the lowest difference in average pixel saliency in foreground and background yield the most interesting failure modes. We present examples selected this way in Figure 7, highlighting spurious background features that contribute to the low RFS of ships and birds observed across models in Figure 5. Specifically, models look for water and coasts when classifying ships, and twigs and branches when classifying birds.

In Figure 9, we appeal to the standard metric intersection-over-union (IOU). Saliency maps are binarized using a threshold of 0.5 before being compared with object segmentation masks. On average, saliency alignment is similar across models, despite their being large differences in RFS identified in the noise analysis, suggesting saliency maps may give an incomplete picture of model sensitivity. In comparing saliency alignment across classes, we see much larger differences, emphasizing the result that **class matters** when it comes to background reliance.

6. Neural Node Attribution Analysis

The attribution of features in a neural network is a fundamental problem in modern machine learning work. Saliency, when computed with respect to a given feature, is a prominent approach for doing so [12, 26, 45, 51]. Although many works make claims of attribution based on saliency, to the best of our knowledge, quantitative validation is rarely given [63]. Here we propose to quantitatively evaluate node attribution via saliency through comparison with the ground-truth attribute localization in RIVAL10.

We propose the following procedure. Given a pretrained robust ResNet50 feature extractor and a class label, we identify the top 10 training images by activation with that label for each component in the feature layer (the penultimate layer). We then compute saliency using GradCAM at each neural feature on these top-10 images, and compare them against ground truth attribute localization. Saliencies are binarized at max-normalized threshold of $\tau = 0.5$. The intersection-over-union (IOU) with the ground truth attribute localization is then computed for each sample, and finally averaged. This obtains a score, which we interpret as measuring the quality of neural feature attribution based on saliency alignment to the attribute segmentations of the top-10 images. We then select the neural feature with highest alignment per attribute, identifying these features as the best candidates for node attribution. Note that searching by top IOU is only possible with ground truth attributes and localizations, as is the case with RIVAL10.

Next, we check if these neural features generalize to held-out data not used in the analysis, namely the test set of RIVAL10. Here we analyze one class-attribute pair and show additional results in the appendix. We visualize the GradCAMs of top testing samples with respect to the top features identified in the training set in Figure 10. We observe visually that the saliencies align well with the given attribute on these samples. We then compute the IOU scores on *all* images in the test set with the given class and attribute labels. We plot this histogram in Figure 10. We observe that IOU values are on average high (> 0.5) indicating that the neural features generalize well to held-out data for considered cases. We note that this analysis is just one approach for quantitatively evaluating feature attribution. We stress the importance of quantitative measurements rather than relying on just visualization, and envision that our RIVAL10 dataset may help refine the discourse around feature attribution.

7. Conclusion

We present **RI**ch **V**isual **A**ttributes with **L**ocalization (RIVAL10), and quantitatively assess sensitivities of state-of-the-art models under noise corruption. Specifically, we find adversarially or contrastively training ResNets leads to

reduced relative foreground sensitivity. Further, we observe transformers to adaptively raise foreground sensitivity as noise level increases, while ResNets do not. Applying automated alignment metrics to saliency maps reveals instances of spurious background features used by models. Lastly, we observe promising evidence that neural node attributions based on top activating images generalize to instances unseen during attribution. We hope RIVAL10's rich annotations lead future studies to gain new quantifiable insights on the behavior of deep image classifiers.

8. Acknowledgements

This project was supported in part by NSF CAREER AWARD 1942230, ONR grant 13370299, HR001119S0026, ARL grant W911NF2120076, and AWS Machine Learning Research Award.

References

[1] Andreas Geiger Axel Sauer. Counterfactual generative networks. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[2] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Joshua B. Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*, 2019. 3

[3] Samyadeep Basu, Phillip Pope, and Soheil Feizi. Influence functions in deep learning are fragile. *CoRR*, abs/2006.14651, 2020. 3

[4] Philipp Benz, Soomin Ham, Chaoning Zhang, Adil Karjauv, and In So Kweon. Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. *arXiv preprint arXiv:2110.02797*, 2021. 5

[5] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. *CoRR*, abs/2103.14586, 2021. 5

[6] Chun-Hao Chang, George Alexandru Adam, and Anna Goldenberg. Towards robust classification model by counterfactual and invariant data generation. *arXiv preprint arXiv:2106.01127*, 2021. 3

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 4, 13, 14

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 2, 4

[10] 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. 1, 4, 13, 14

[11] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 14

[12] Gabriel G. Erion, Joseph D. Janizek, Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Learning explainable models using attribution priors. *CoRR*, abs/1906.10670, 2019. 8

[13] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 4

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. 3

[15] William Falcon and Kyunghyun Cho. A framework for contrastive self-supervised learning and designing a new approach. *arXiv preprint arXiv:2009.00104*, 2020. 14

[16] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009. 3

[17] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780, 2020. 3

[18] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688, Jul. 2019. 3

[19] Sven Gowal, Chongli Qin, Po-Sen Huang, A. Taylan Cemgil, Krishnamurthy Dvijotham, Timothy A. Mann, and Pushmeet Kohli. Achieving robustness in the wild via adversarial mixing with disentangled representations. *CoRR*, abs/1912.03192, 2019. 3

[20] 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. 1, 4, 13, 14

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 3

[22] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 3

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1

[24] Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and

Serge J. Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. *CoRR*, abs/2004.12276, 2020. 3

[25] Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. *CoRR*, abs/2012.04104, 2020. 3

[26] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 8

[27] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions, 2020. 3

[28] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 2

[29] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list. 4

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 3

[32] 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. 4, 13, 14

[33] Tanushree Mitra, C.J. Hutto, and Eric Gilbert. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1345–1354, New York, NY, USA, 2015. Association for Computing Machinery. 4

[34] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *CoRR*, abs/1602.03616, 2016. 3

[35] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 3

[36] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. 3

[37] Wanli Ouyang, Hongyang Li, Xingyu Zeng, and Xiaogang Wang. Learning deep representation with large-scale attributes. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 3

[38] 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. 14

[39] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners, 2021. 5

[40] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028, 2021. 3

[41] 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. 4, 14

[42] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016. 3

[43] Olga Russakovsky and Li Fei-Fei. Attribute learning in large-scale datasets. In *European Conference on Computer Vision*, pages 1–14. Springer, 2010. 3, 4

[44] Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *CoRR*, abs/1911.08731, 2019. 3

[45] 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. 2, 3, 8, 16

[46] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of visual transformers. *CoRR*, abs/2103.15670, 2021. 5

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[48] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning?, 2021. 3

[49] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[50] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017. 2

[51] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th Interna-*

tional Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017. 2, 8

[52] 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. 16

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1

[54] 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. 5, 14

[55] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations: Why and how to pass stress tests. *CoRR*, abs/2106.00545, 2021. 3

[56] Tianlu Wang, Diyi Yang, and Xuezhi Wang. Identifying and mitigating spurious correlations for improving robustness in nlp models, 2021. 6

[57] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 3

[58] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 14

[59] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11205–11216. PMLR, 18–24 Jul 2021. 3

[60] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *CoRR*, abs/2006.09994, 2020. 3, 6, 7

[61] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. *arXiv preprint arXiv:2106.04560*, 2021. 1

[62] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. *CoRR*, abs/1512.04150, 2015. 2

[63] Yilun Zhou, Serena Booth, Marco Tílio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? *CoRR*, abs/2104.14403, 2021. 8

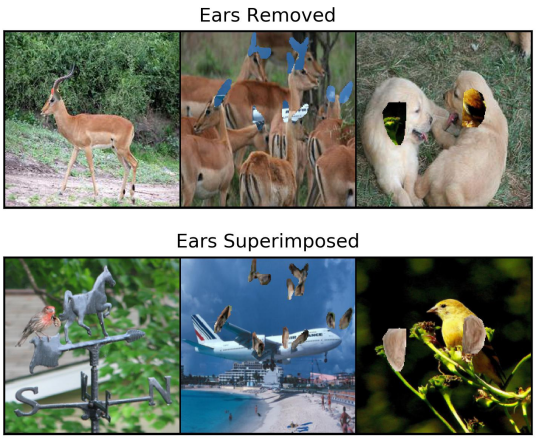


Figure 11. Examples of attribute-swapped inputs.

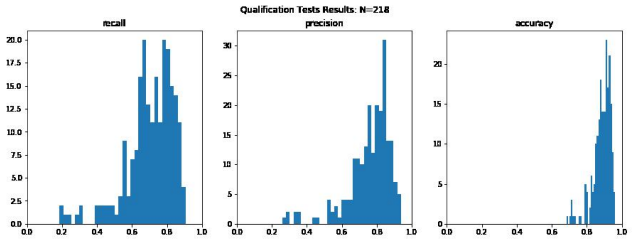


Figure 12. Histograms of Worker recall, precision, and accuracy scores on the qualification exam.

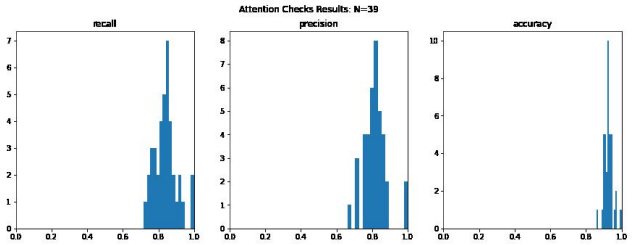


Figure 13. Histograms of per-Worker recall and precision on randomly placed attention checks during the main phase of data collection.

A. 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 25 and 26, 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 11, we display examples of challenging inputs yielded via attribute removal and insertion.

B. 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 segmentation 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 12.

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 12 (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

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.

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 13.

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 37 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 33, 34, 35, and 36 respectively. We have redacted identifying information of the authors appropriately.

C. 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.

C.1. Architectures and Training Procedures

Architecturally, we focus on ResNets [20] and Transformers [10]. 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 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 [32], 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 [7], two views of a single input are

Model	Pretraining Set	Parameter Count	RIVAL10 Accuracy	Source of Weights	Original Paper	Notes
ResNet18	IN-1k	11.4M	95.48	[38]	[20]	
ResNet50		23.9M	99.10			
ResNet101		42.8M	99.21			
ResNet152		58.5M	99.43			
Robust ResNet18	IN-1k	11.4M	91.80	[11]	[32]	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	[15]	[7]	RN50 backbone
CLIP ResNet50	YFCC100M	23.9M	96.34	[41]	[41]	
CLIP ResNet101		42.8M	96.27			
CLIP ViT-B/16		86M	99.17			Patch=16 $\times$ 16
CLIP ViT-B/32		87M	98.44			Patch=32 $\times$ 32
ViT (Tiny)	IN-21k + IN-1k	5M	94.82	[58]	[10]	Patch=16 $\times$ 16
ViT (Small)		22M	98.96			Patch=16 $\times$ 16
ViT (Base)		86M	99.64			Patch=16 $\times$ 16
ViT (Small)†		23M	97.86			Patch=32 $\times$ 32
ViT (Base)†		87M	99.26			Patch=32 $\times$ 32
DeiT (Tiny)	IN-1k	5M	96.42	[58]	[54]	Patch=16 $\times$ 16
DeiT (Small)		22M	99.30			Patch=16 $\times$ 16
DeiT (Base)		86M	99.74			Patch=16 $\times$ 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.

created via data augmentation. In CLIP [41], 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.

C.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.

C.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

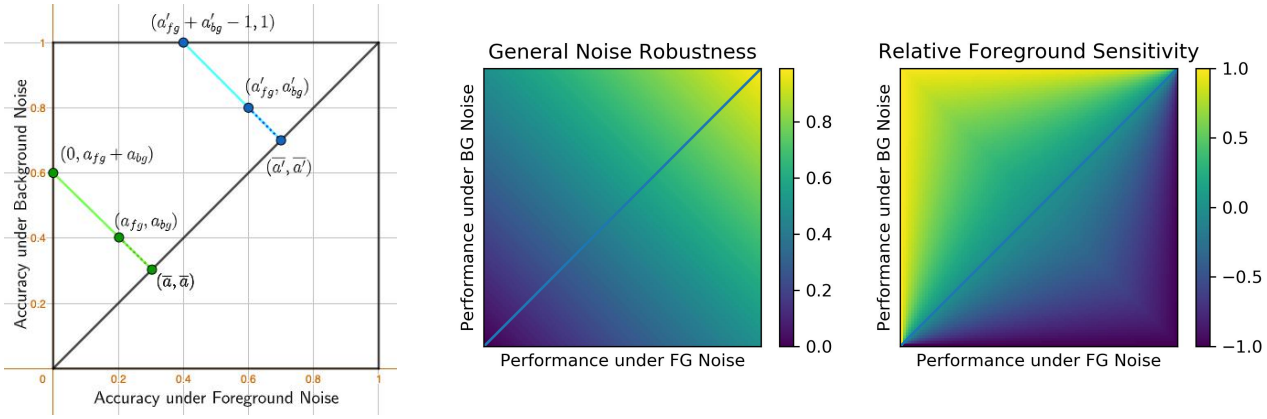


Figure 14. (**Left**) We demonstrate how RFS is derived as a ratio of the distance of (a_{fg}, a_{bg}) from the diagonal over the maximum distance to the diagonal for a point with fixed noise robustness $\overline{a} = 1/2(a_{fg} + a_{bg})$. (**Right**) Visualization of general noise robustness and relative 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.

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.

D. 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.

D.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 14 (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 14, we visualize both general noise robustness and RFS for all accuracies under foreground and background noise to add further context.

D.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.

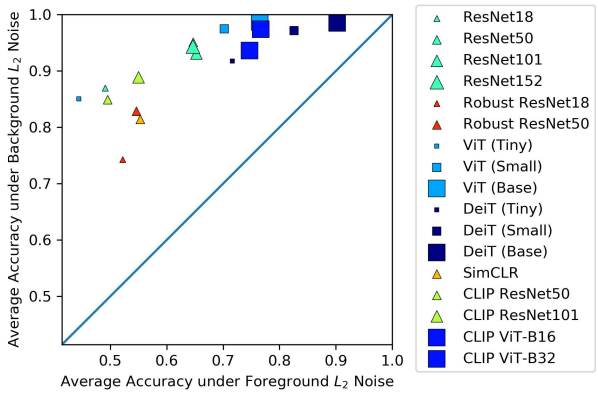


Figure 15. Accuracy under $\mathbf{L_2}$ normalized noise averaged over multiple noise levels. Marker size is proportional to parameter count. Models with higher relative foreground sensitivity lie further from the diagonal.

We find that the trends are near identical. The one small difference is that the class distinctions in Figure 17 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.

E. Saliency Alignment

To complement the noise analysis, we inspected saliency maps obtained via GradCAM [45], 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 foreground. Another interpretation of this metric is the fraction of total saliency in the foreground, similar to [52].

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

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.

E.1. Empirical Observations

We present complete quantitative saliency alignment results in Figure 18. 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

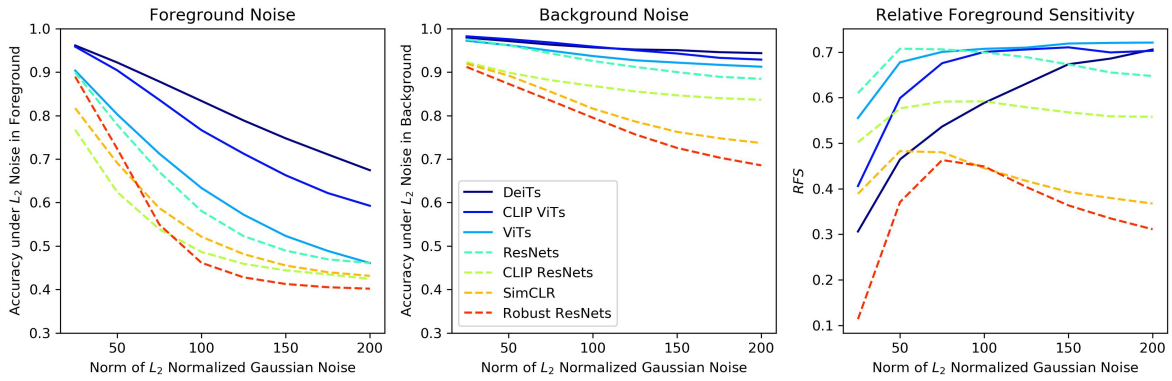


Figure 16. Accuracy under $\mathbf{L_2}$ normalized noise in foreground (**left**) and background (**middle**) at various noise levels. Models are grouped by architecture and training procedure, with a curve corresponding to the average over all models in a group. (**Right**): RFS by group.

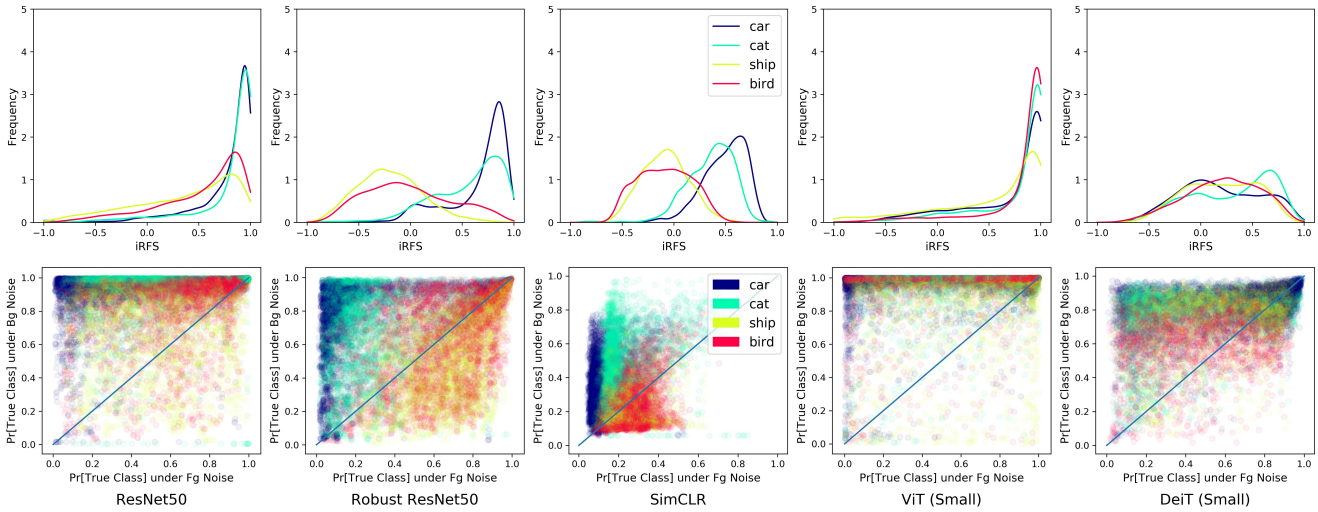


Figure 17. Relative foreground sensitivity per instance for four classes and five models of roughly equal size, computed using $\mathbf{L_2}$ normalized noise corruption. (**Top**): Histogram of $iRFS$; positive denotes greater foreground sensitivity. (**Bottom**): Scatter; top left indicates high relative foreground sensitivity. Class distinction is slightly less pronounced than with L_∞ noise, but still substantial.

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 GradCAMs 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.

F. 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

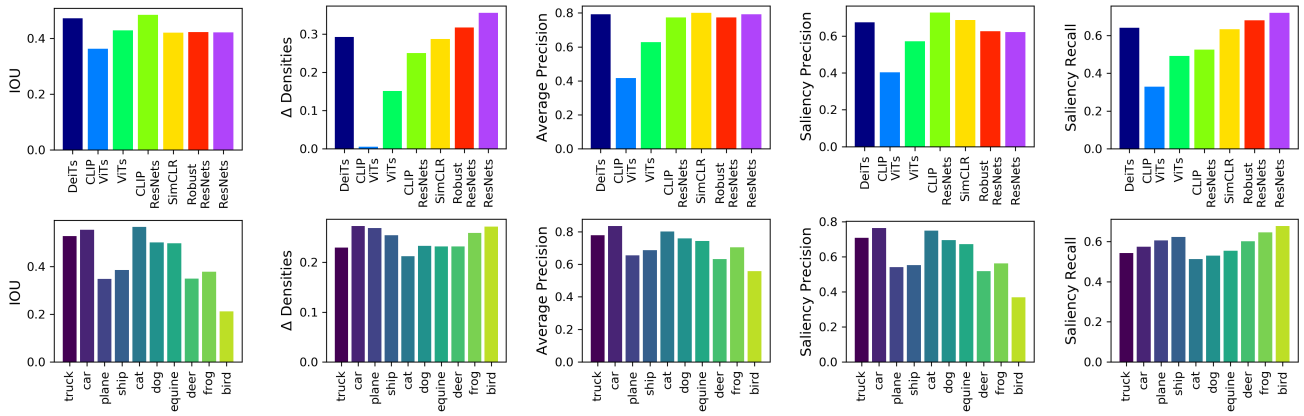


Figure 18. Saliency alignment averaged over model categories (top) and object classes (bottom) for five alignment metrics.

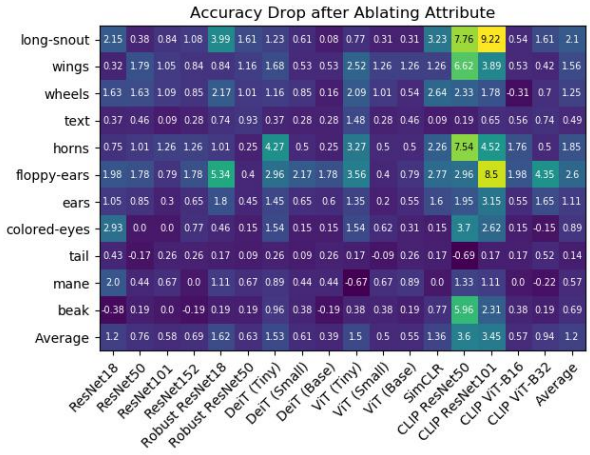


Figure 19. Degradation to model performance due to attribute ablation (via graying), as measured by accuracy.

in performing RIVAL10 classification are not very important to deep classifiers.

G. Additional Qualitative Examples of Background Sensitivity

We provide additional examples qualitatively demonstrating instances where models have high background sensitivity. Figure 20 shows GradCAMs where saliencies have worst alignment with foreground, as measured by Δ Densities, for a Robust ResNet50. In Figure 21, we display instances where noise corruption reveals greater background sensitivity for Robust ResNet50 and DeiT (Small).

H. Additional Visualizations for Neural Node Attribution

We show GradCAMs and IOU histograms for another top feature attribute pair, cars and wheels, in Figure 22. 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 23 and 24 respectively. Interestingly, feature activation value and saliency alignment (as measured by IOU) do not seem to be strongly related.

I. 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 27, 28, 29, 30. 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 31 and 32. 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

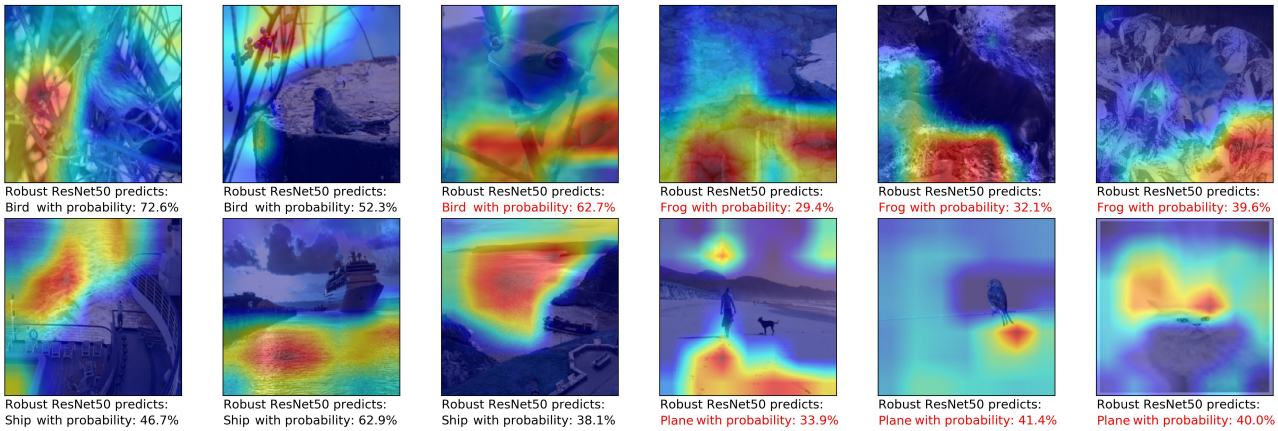


Figure 20. Additional examples of spurious features used by a Robust ResNet50 observed via sorting images by saliency alignment (Δ Densities). Misclassifications are in red text. Spurious features include branches, dry leaves, water, and sky.

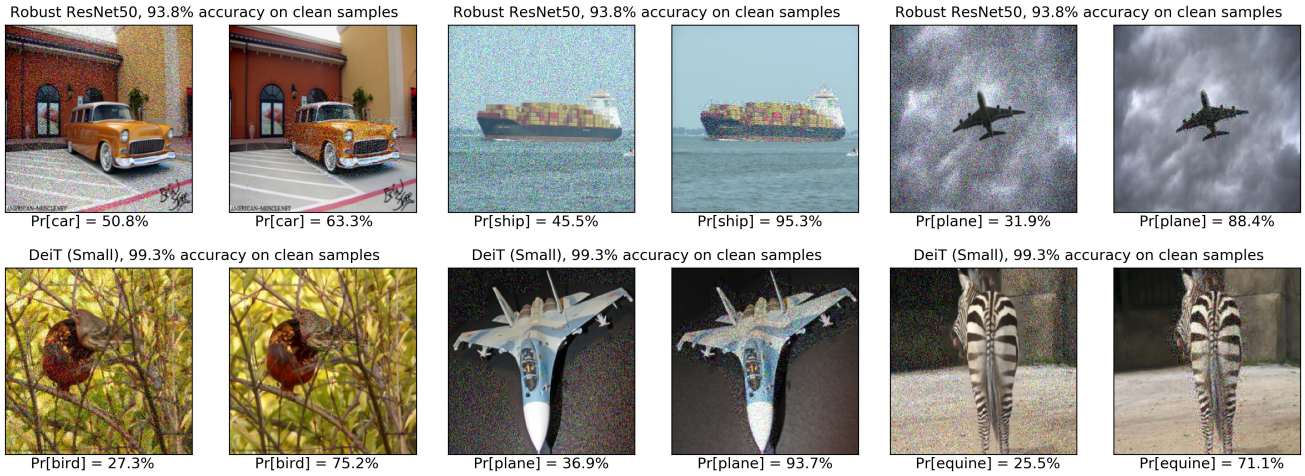


Figure 21. Additional examples where background noise degrades performance of highly accurate models more than foreground noise. (**top**): Robust ResNet50, (**bottom**): DeiT (Small). Gaussian ℓ_∞ noise with standard deviation $\sigma = 0.12$ shown. Probabilities are averaged over ten trials.

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.

J. 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 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.

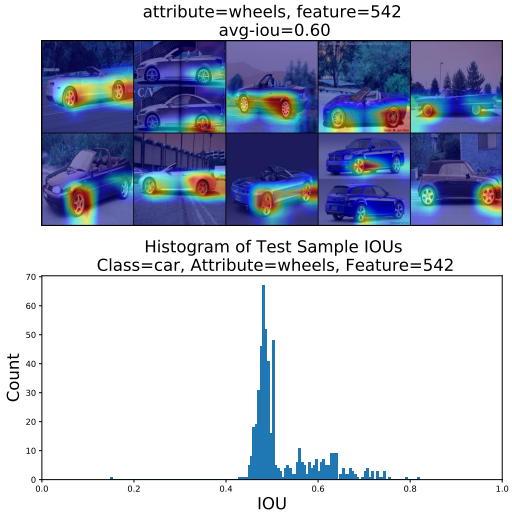


Figure 22. (**Top**): Example GradCAMs on test images with respect to the top feature identified by IOU in training set. (**Bottom**): Histograms of IOUs corresponding to this feature, attribute pair.

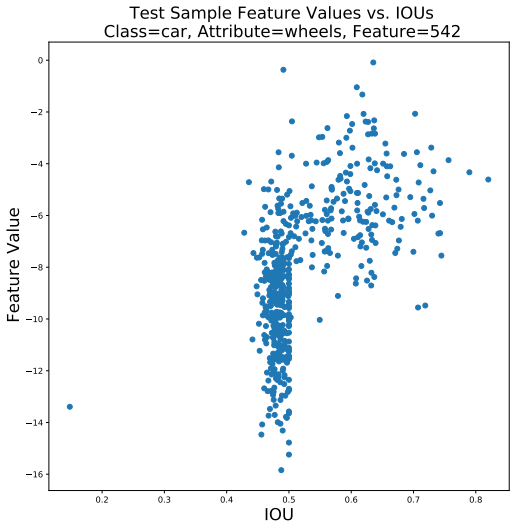


Figure 23. Feature values vs IOU scores for class-attribute pair car and wheels.

K. 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.

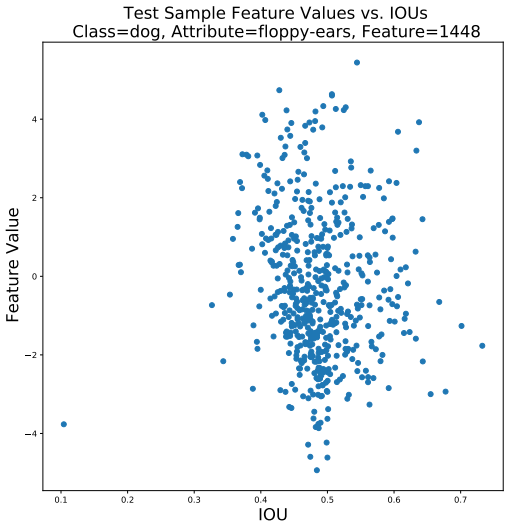


Figure 24. Feature values vs IOU scores for class-attribute pair dog and floppy-ears.

L. 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.

M. 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.

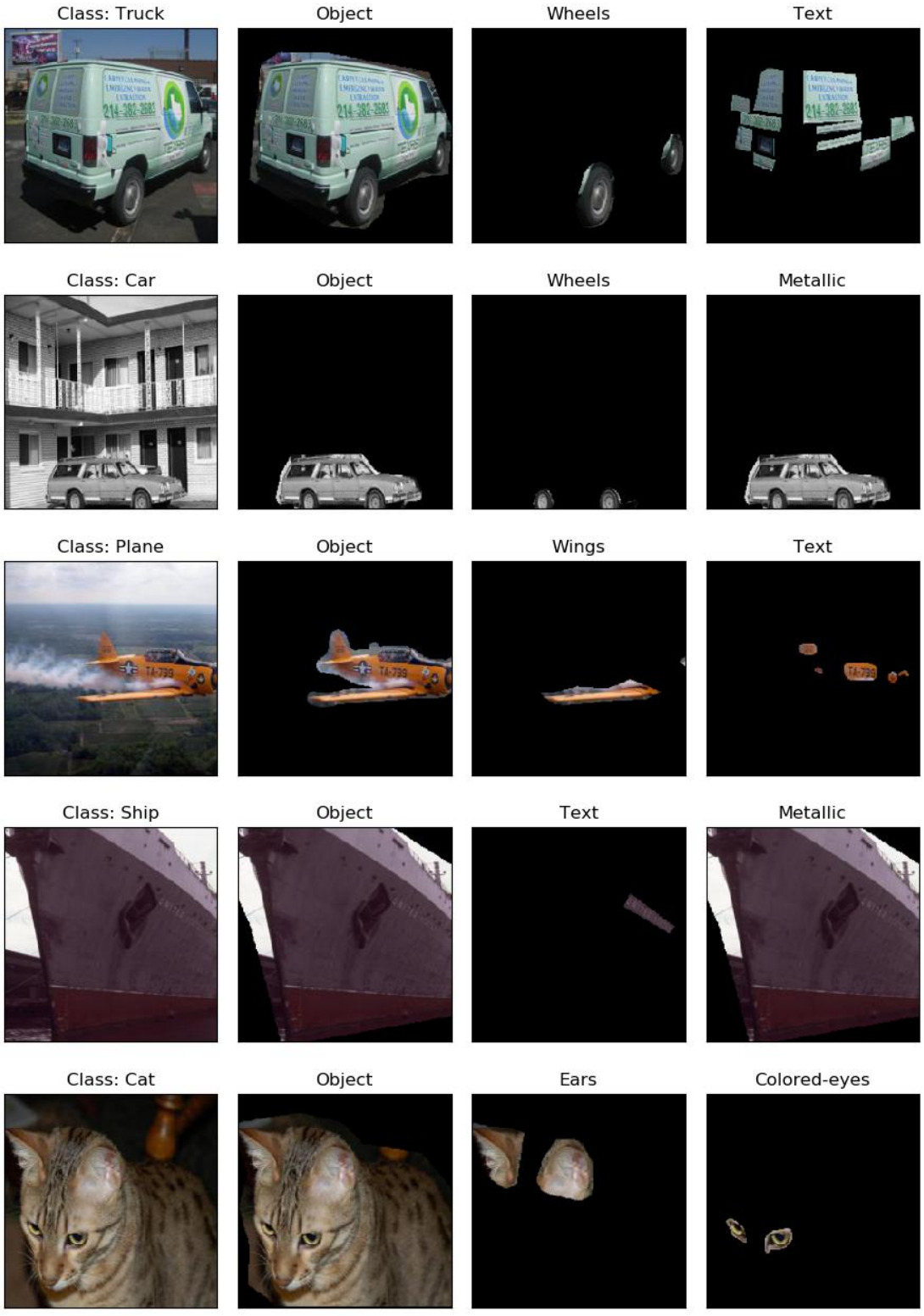


Figure 25. RIVAL10 examples. Left column has original image. Next column shows object mask applied onto the original image. The following two columns show attribute masks applied onto the original image.

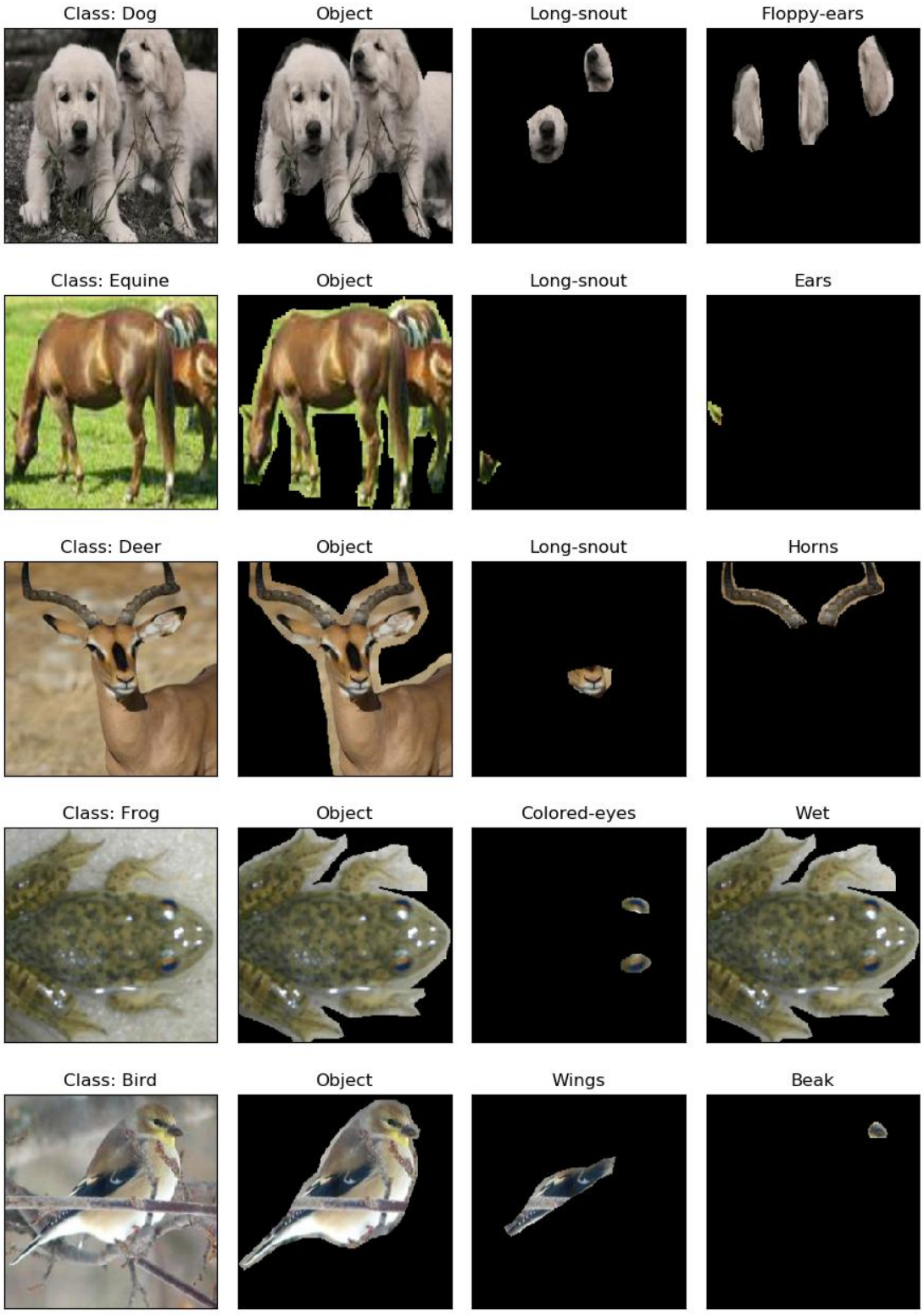
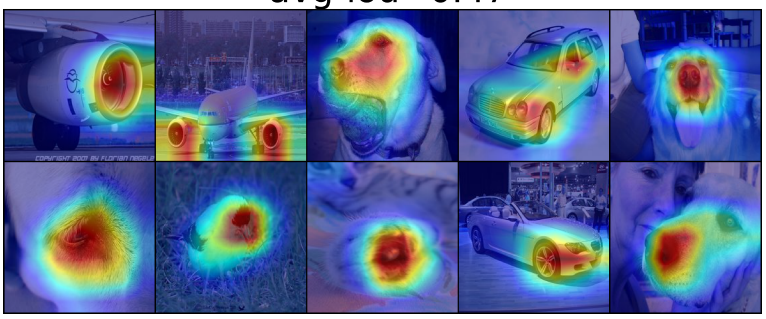
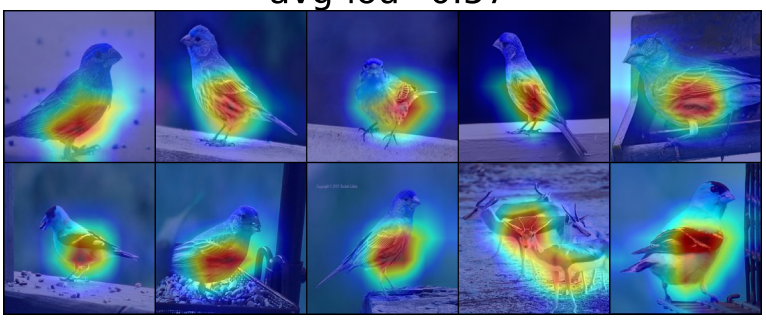


Figure 26. RIVAL10 examples. Left column has original image. Next column shows object mask applied onto the original image. The following two columns show attribute masks applied onto the original image.

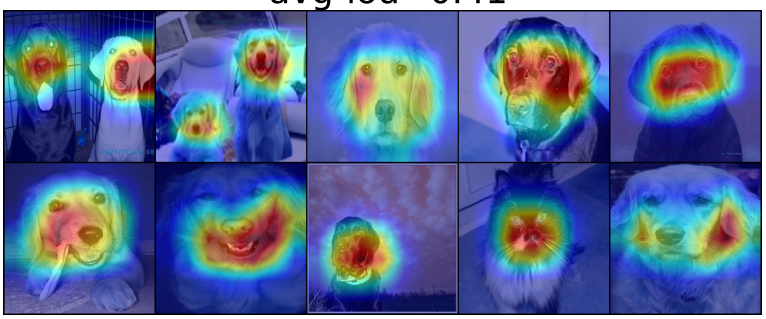
attribute=long-snout, feature=774
avg-iou=0.47



attribute=wings, feature=618
avg-iou=0.57



attribute=wheels, feature=542
avg-iou=0.41



attribute=text, feature=1890
avg-iou=0.59

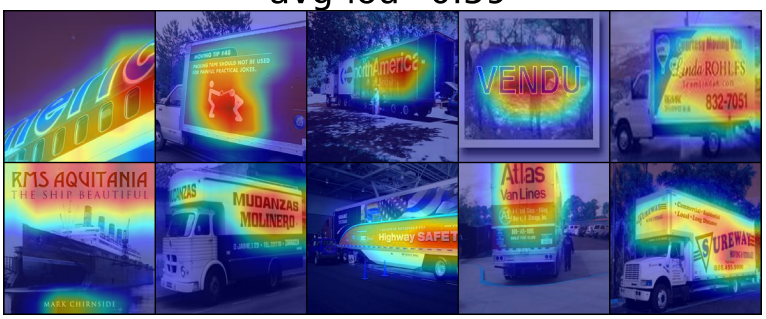


Figure 27. Saliency for top feature attribute pairs by IOU. First quarter of results shown here.

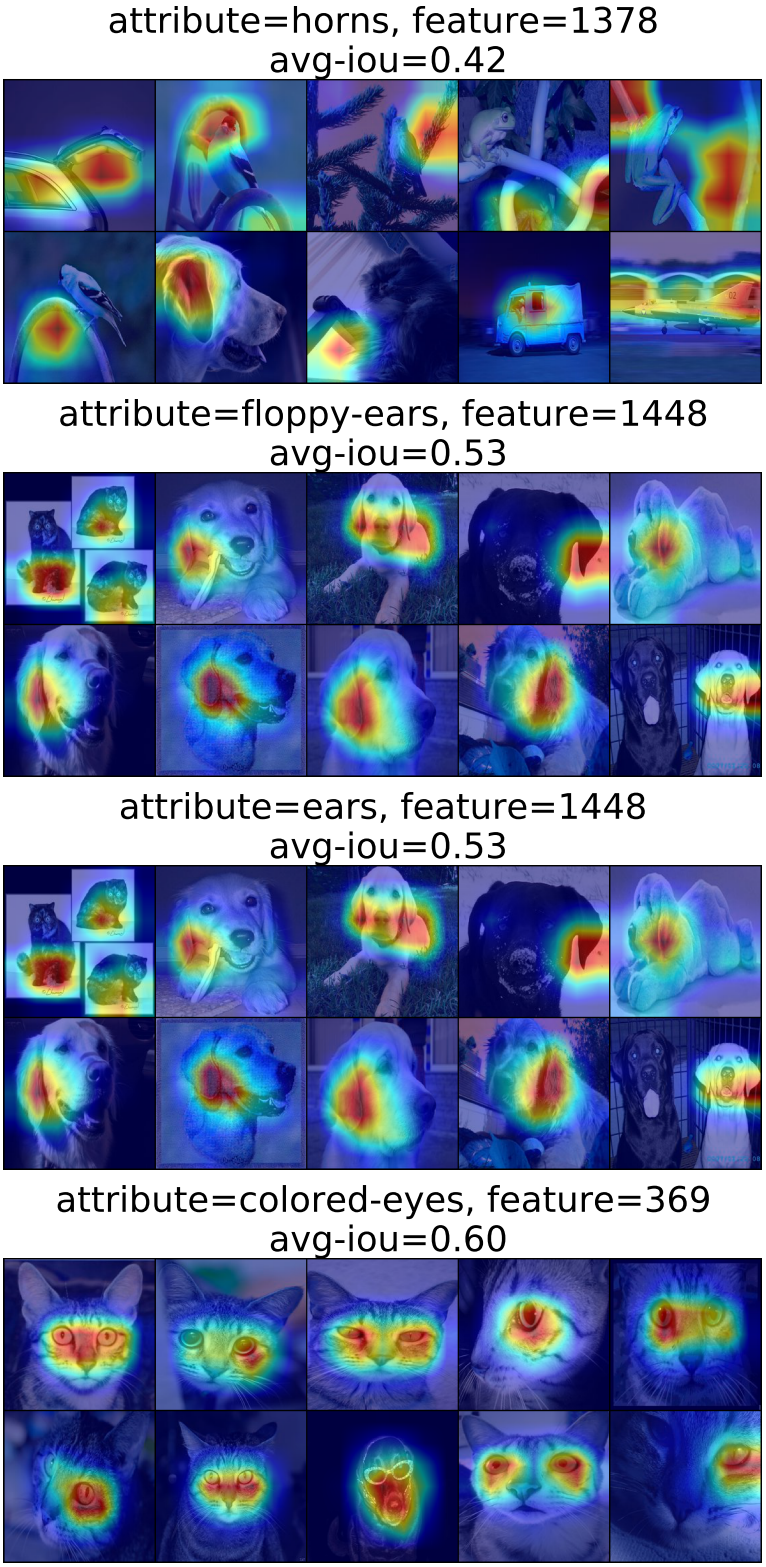


Figure 28. Saliency for top feature attribute pairs by IOU. Second quarter of results shown here.

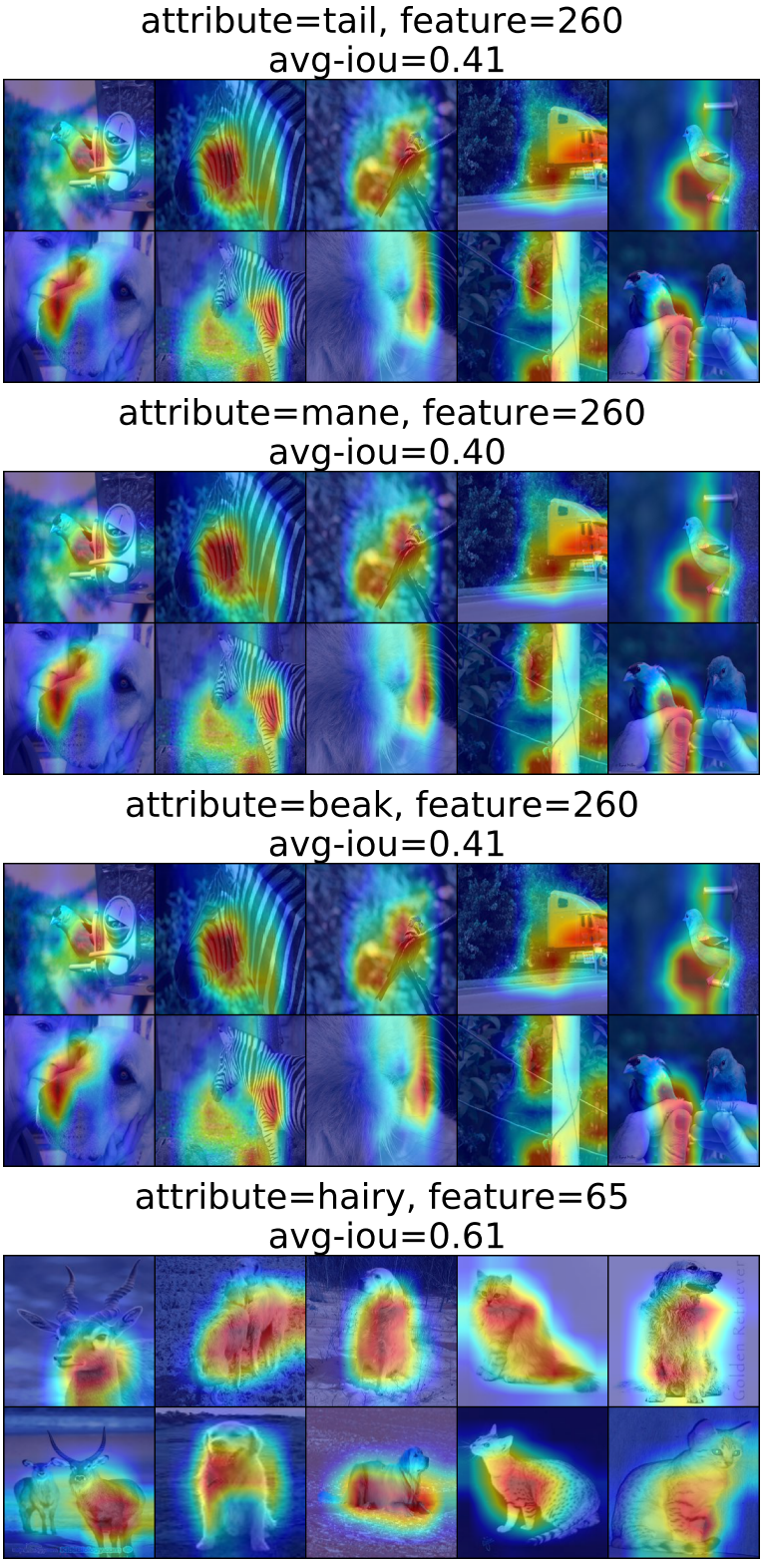
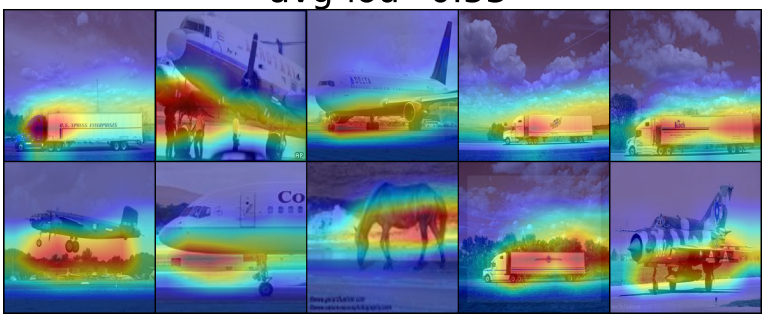
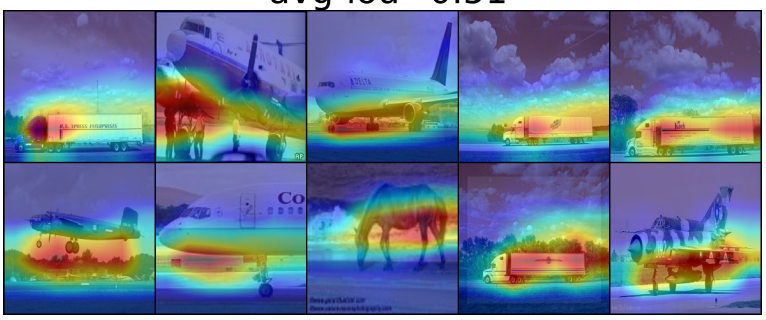


Figure 29. Saliency for top feature attribute pairs by IOU. Third quarter of results shown here.

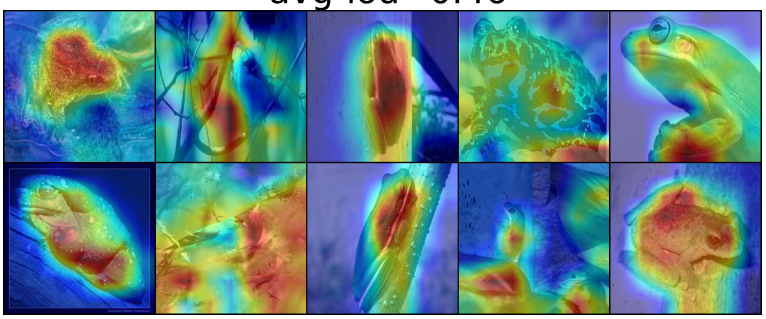
attribute=metallic, feature=1237
avg-iou=0.53



attribute=rectangular, feature=1237
avg-iou=0.51



attribute=wet, feature=1580
avg-iou=0.46



attribute=long, feature=1237
avg-iou=0.53

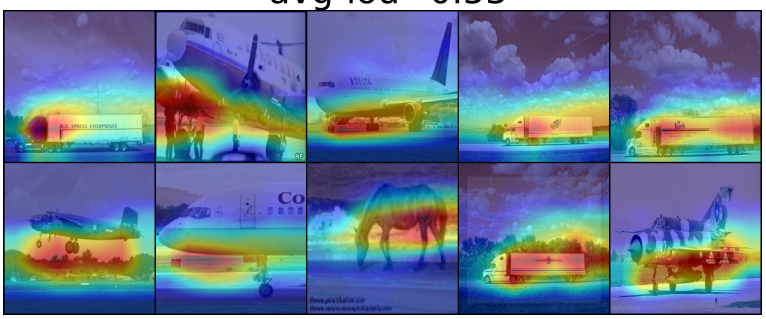


Figure 30. Saliency for top feature attribute pairs by IOU. Fourth quarter of results shown here.

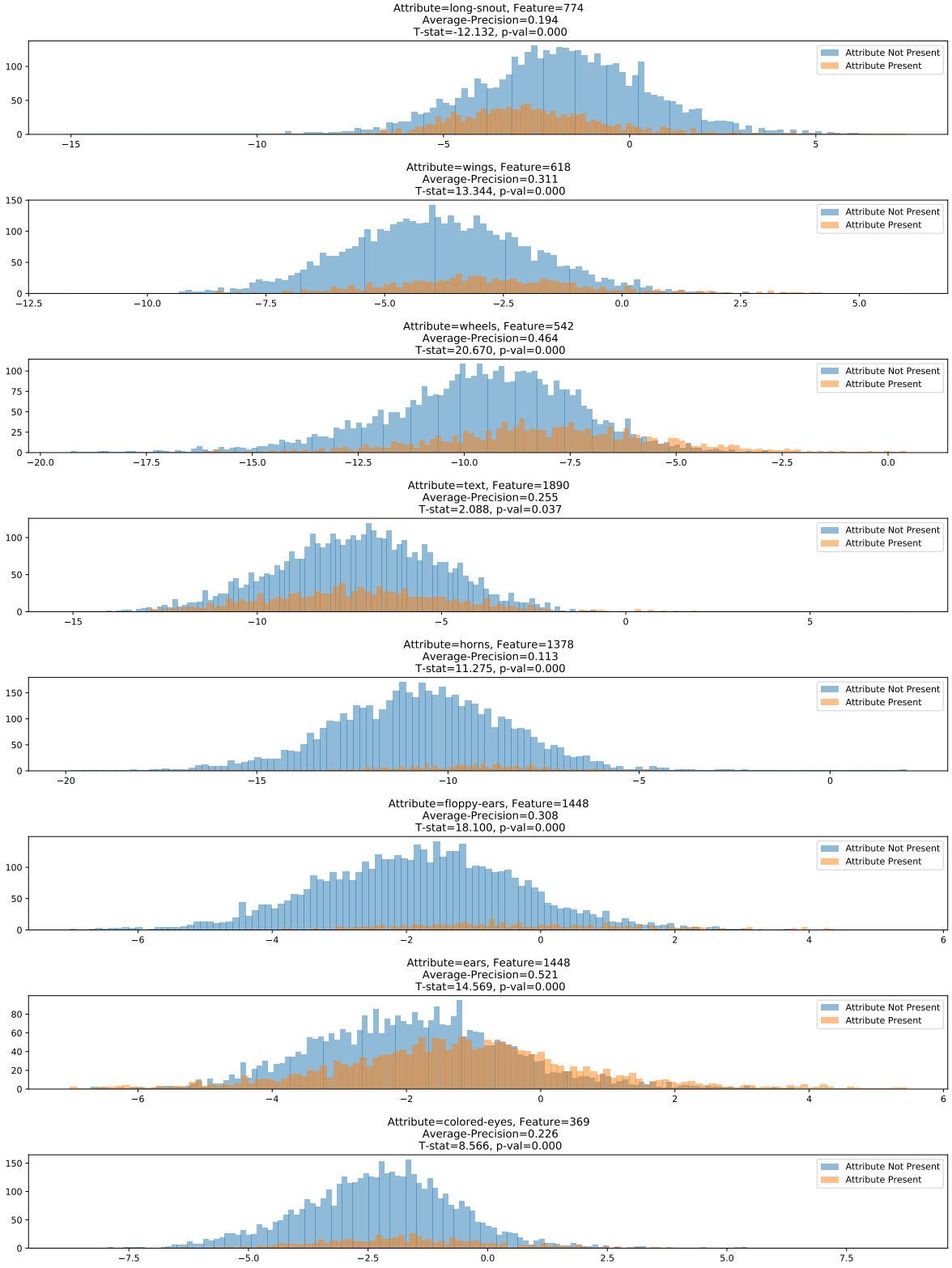


Figure 31. Top feature histograms for the top attribute feature pairs. First half shown here.

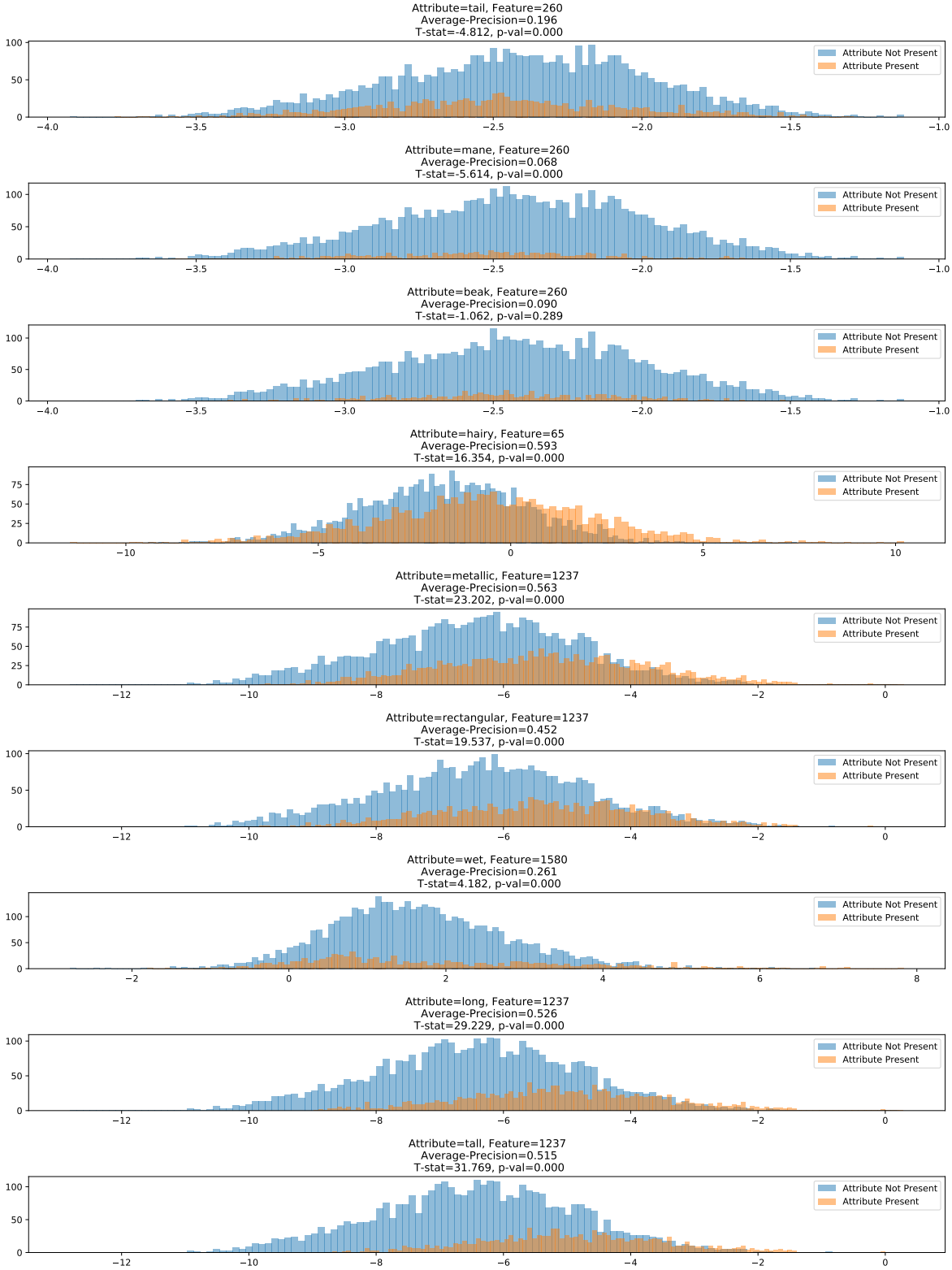


Figure 32. Top feature histograms for the top attribute feature pairs. Second half shown here.

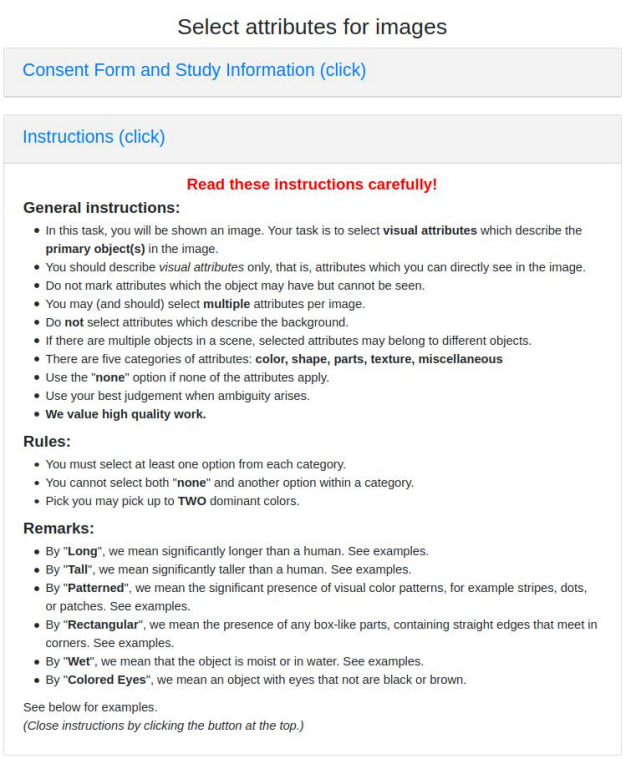


Figure 33. Screenshot of instructions page shown to workers

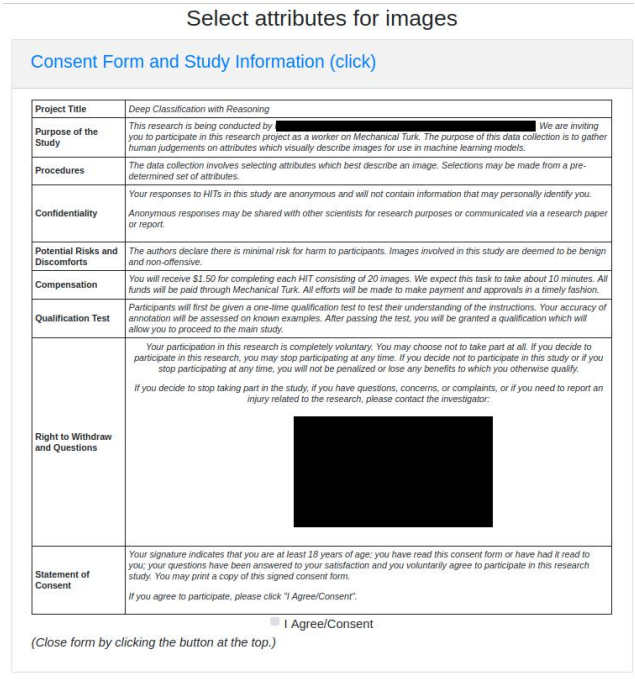


Figure 34. Screenshot of consent page shown to workers

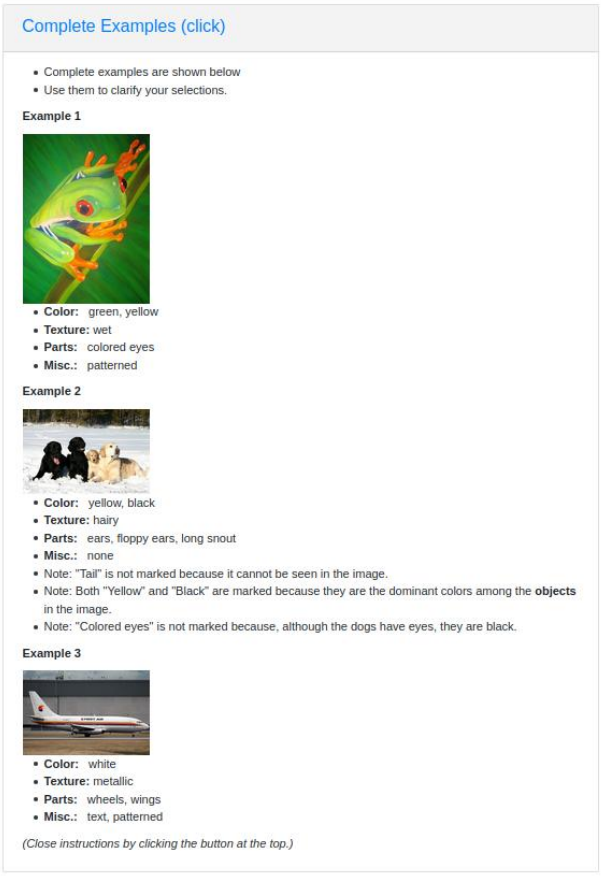


Figure 35. Screenshot of examples page shown to workers

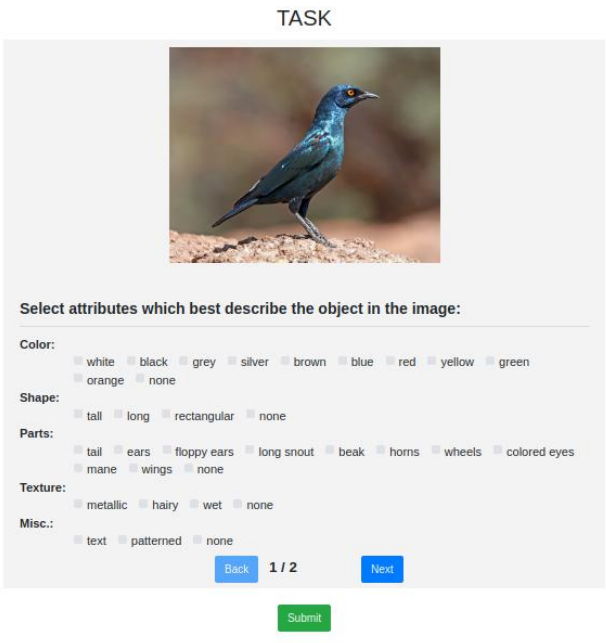


Figure 36. Screenshot of annotation form shown to workers

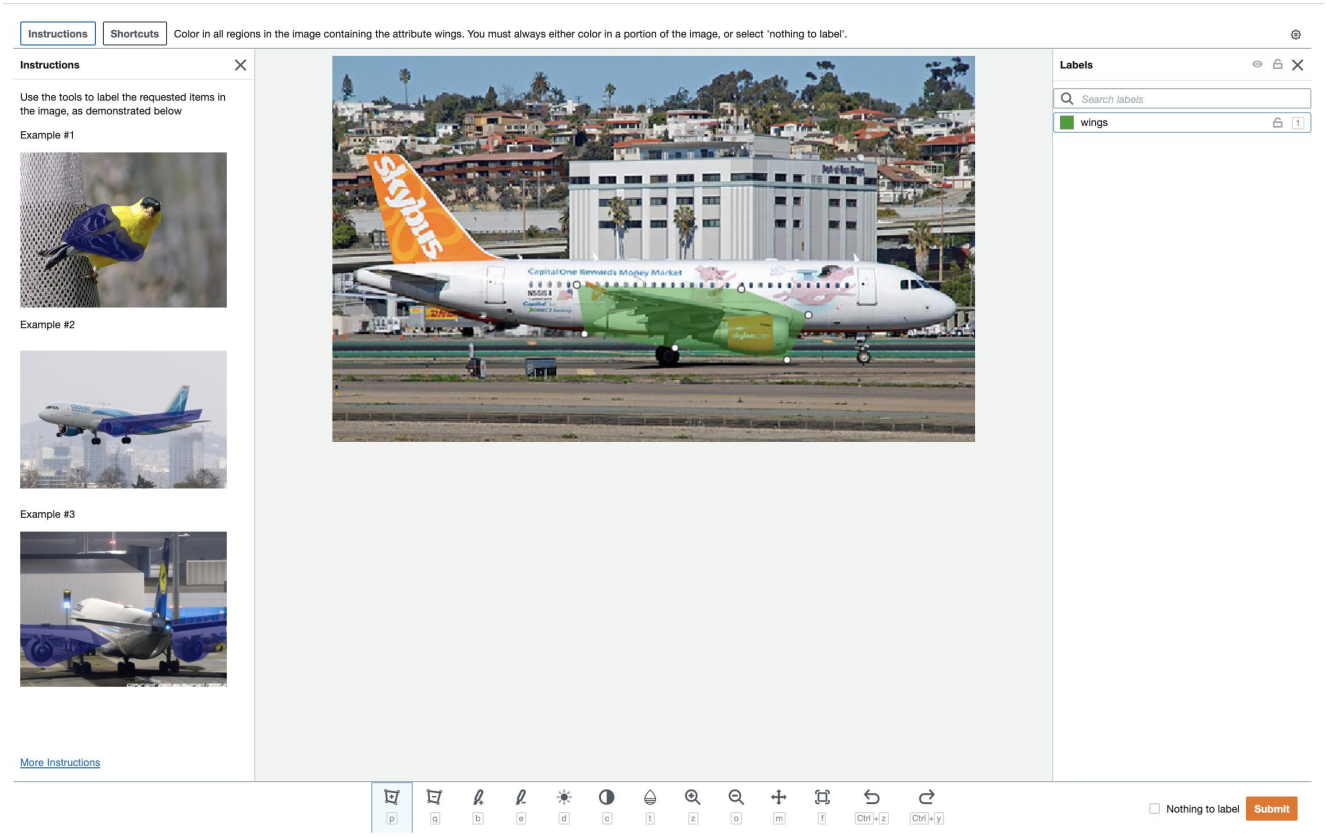


Figure 37. Screenshot of annotation form and tools for completing segmentations.