ObEy: Quantifiable Object-based Explainability without Ground-Truth Annotations

Lennart Schulze*
Dept. of Computer Science
Columbia University
New York City, NY 10027
lennart.schulze@columbia.edu

William Ho*
Dept. of Computer Science
Columbia University
New York City, NY 10027
wh2529@columbia.edu

Richard Zemel
Dept. of Computer Science
Columbia University
New York City, NY 10027
zemel@cs.columbia.edu

Abstract

Neural networks are at the core of AI systems recently observing accelerated adoption in high-stakes environments. Consequently, understanding their black-box predictive behavior is paramount. Current explainable AI techniques, however, are limited to explaining a single prediction, rather than characterizing the inherent ability of the model to be explained, reducing their usefulness to manual inspection of samples. In this work, we offer a conceptual distinction between explanation methods and explainability. We use this motivation to propose Object-based Explainability (ObEy), a novel model explainability metric that collectively assesses model-produced saliency maps relative to objects in images, inspired by humans' perception of scenes. To render ObEy independent of the prediction task, we use full-image instance segmentations obtained from a foundation model, making the metric applicable on existing models in any setting. We demonstrate ObEy's immediate applicability to use cases in model inspection and comparison. As a result, we present new insights into the explainability of adversarially trained models from a quantitative perspective.

1 Introduction

Due to their predictive power, neural networks are at the center of the adoption of artificial intelligence (AI) in industry and society. This comes at the expense of their inherently intransparent, black-box predictive behavior. In response, the research community has spent significant efforts in the past years to devise methods that facilitate understanding of neural networks' behavior [5]. While explainable AI is demanded by all stakeholder groups, including in legislation and industry [6, 7], however, there remains a gap between scientific methods and explainability tools used by practitioners, with the former remaining largely unadopted outside the research community [8, 9].

Part of the reason for this deficit is the large variety of methods [1–3, 10–15], paired with uncertainty about the purpose with which to use them. This uncertainty can be attributed to the fact that current post-hoc explanation methods may mainly be used to manually inspect individual predictions on a small number of samples after training. They do not, however, offer to quantify and summarize the model's explainability. Thus, at model development time, still the task-specific performance metric, not the model's explainability, is used to guide decisions and development directions. This leads to concrete drawbacks for researchers and practitioners. For instance, opaque but well-performing models are preferred without regard to their lack of explainability and insights from analyzing the explainability are not used to improve models.

Consequently, in this work, we propose a new metric to quantify the explainability of a model, coined Object-based Explainability (ObEy). To this end, we follow arguments from the explanation

*Equal contribution. Order decided by coin flip.

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

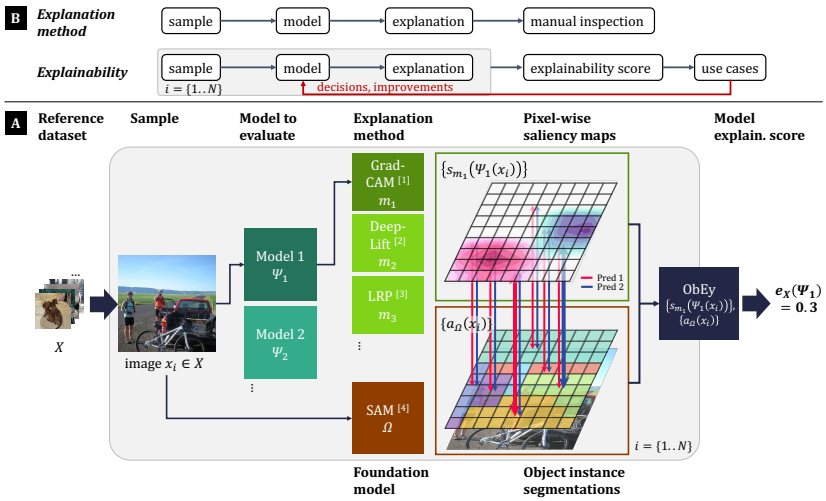

Figure 1: **ObEy overview:** A) Relative to a reference dataset, ObEy relates post-hoc saliency maps [1–3] that explain individual model predictions to generated [4] instance segmentation masks via a custom weighted intersection over union metric. The saliency map of each prediction is measured by its maximum congruence with any object, independent of its correctness. B) Going beyond per-sample explanations, ObEy offers use cases in model comparison and investigation.

method evaluation literature, and re-purpose them to construct a generalized method that captures to which extent vision-based models base their predictions on human-perceivable objects. Using ground-truth annotations for all objects, ObEy disentangles explainability from performance and is independent of the task. By leveraging recent advances in foundation models for object segmentation [4], our method can be used on any model, enabling adoption in any setting.

We show how using this metric to measure a model's explainability has immediate downstream use cases and benefits in the model development and adoption pipeline. Precisely, we demonstrate how ObEy can be used to compare and decide between models, and how it can be used as investigative tool to debug a single model. As a result of our experiments, we provide quantification for a previously identified aspect of adversarial training [16, 17] which is that adversarially trained models tend to be more explainable. In summary, our contributions are the following:

- We introduce a novel metric to quantify the explainability of any vision model based on saliency methods and Segment-Anything-generated all-object annotations.
- We propose immediate use cases of employing this metric in the model development and adoption process.
- Demonstrating these use cases on different models and datasets, we show that adversarially trained models are more explainable in certain task settings.

2 Related work

Explainability. The work on explainability in AI is vast and ambiguous [18]. We offer a new perspective and, throughout this work, differentiate explanation methods from explainability (Section 3.1). We use the former to refer to methods that aim to explain the model's prediction behavior on a single sample [19]. In contrast, we refer to the latter to describe a model's property. We adopt the definition of [5] to define explainability as the "ability a model has to make its functioning clearer to an audience", independent of individual samples. In this work, we quantify this ability by characterizing explanation method results collectively, as detailed below. For the purpose of our contribution, we consider interpretability to be subsumed under explainability.

The existing literature on explainability in the broader sense can be subdivided into explanation methods to explain a model's prediction and methods to (quantitatively) evaluate and compare those explanation methods in their approach. In this work, we do not offer new explanation

methods, nor method evaluation metrics, but use existing explanation methods to define a metric for a *model*'s explainability, following similar notions of goodness as method evaluation metrics.

Explanation methods. Post-hoc explanation methods generate an explanation for the prediction of a model on an individual sample. These can be grouped into white-box methods [1–3, 10, 11, 15] leveraging insight into the network or black-box methods [12, 20–25] exclusively using the systematic access to inputs and predictions to explain the model's behavior on a sample. White-box methods use the model internals, such as the gradients of the output with respect to the input or activated layers, to produce a saliency map (also heatmap, attribution map, importance map, explanation map) over the input. For images, each pixel of the input is attributed a positive, or sometimes negative, intensity value of contribution towards the prediction output.

Thorough explanation method reviews are provided in [5, 19, 26–29]. However, it was found that many of these methods produce saliency maps that are independent of the model. They only replicate or detect features of the input, such as edges, without explaining the computations in the model, thus failing their objective [30]. Methods explicitly found not to fall subject to this failure mode are simple input gradients [31, 32], or Grad-CAM [1]. Grad-CAM produces saliency maps that indicate which input pixels had a positive influence on the prediction of a class. At a target layer, it weighs the activation of the input by the gradients flowing back to that layer.

Explanation method evaluation metrics. On top of explanation methods, evaluation metrics assess the goodness of an explanation method relative to an ideal for a good explanation. These evaluate different aspects such as explanation methods' ability to capture the parts of the image that are relevant for the model's prediction, called fidelity, their robustness, their intuitiveness, their ability to localize, and others [33]. Customary approaches to measure robustness as well as fidelity include, respectively, perturbing regions of the input and assessing the response in the explanation [23, 34, 35], as well as removing regions of the input [28, 36–38] or randomizing model weights [30]. Overviews of comparison metrics, critical discussions, and guiding frameworks are provided in [19, 39, 40]. These evaluation metrics are also used for the purpose of comparing and ranking multiple explanation methods [28, 30, 34–38, 40, 41]. Both for measuring deviation from the ideal within the metric or for comparing the evaluation results between explanation methods, mathematical similarity metrics, such as correlation, are used, as reviewed by [42]. Unlike ObEy, method evaluation metrics are not used to assess the explainability of the model.

However, similar to ObEy, object focus has been used in the context of method evaluation metrics. Particularly, the localization ability of an explanation, as ability to recover scene parts from the saliency map, is one aspect in which explanation methods are evaluated [23, 40, 41, 43]. The ideal may then be provided by bounding boxes or segmentation masks, often with respect to the predictable classes in the image only. This is problematic since models may focus on any feature of the image to produce a prediction [44, 45], so that ObEy uses all-object segmentations instead.

3 Method

3.1 Conceptual motivation

Explainability vs. explanation. A core concept of our contribution is the distinction between explanation methods and explainability (Figure 1), which are often used interchangeably in the literature. We consider most currently denoted explainability methods such as the aforementioned Grad-CAM to be explanation methods: methods which aim to explain a model's prediction on an individual input relative to the model's inner working and processing of the input at a time. These methods do not satisfy the definition of an explainability metric since they do not speak to the *ability* of the model to make its model clearer to an audience. That is, they do not capture to what extent the *model* can produce reasonable explanations based on its design and condition.

In contrast, we define explainability metrics as metrics that quantify the model's ability to be explained, using explanation methods. Particularly, we consider and characterize the goodness of the results of post-hoc explanation methods to conclude about the model's inherent ability to produce them. For example, a CNN image classifier may be explained via Grad-CAM as an explanation method, but the saliency maps produced may consistently be unfocused or exhibit low intensity. In this case, the model's individual decisions can be explained but the model has a

low explainability because those explanations are of low explanatory value. Therefore, applying an explanation method on a model does not render the model explainable, while, however, a model is asserted to be explainable via the behavior of explanation methods applied on it.

In this aspect, while both share that they quantify explanation results, explainability metrics are different from existing explanation method evaluation metrics. These metrics measure the soundness of the approach according to which explanation methods produce their explanations, to the end of comparing the methods rather than the models.

We consider explainability metrics desirable because they bridge the gap from theoretically motivated explanation methods, whose results can only be inspected visually post-hoc, to the model development pipeline by quantification. Thus, explainability metrics provide practical value in several aspects. 1) Optimizing: Quantifying explainability offers to optimize for explainability at training time. Without this ability, models are only optimized for their task performance. In light of the rising expectations, stakes, and legal requirements for many organizations, explainability will be as important of a development objective. 2) Informing model choices: When presented with multiple, similarly performing models to solve a task, knowing and being able to objectively compare their explainability can be a feature for decision-making. 3) Reporting: With a metric to quantify the models' explainability relative to its test distribution, reporting and documentation requirements can be satisfied such as mandated by policies and regulations.

Basing explainability on objects. ObEy uses explanation method results to infer the underlying model's explainability. To this end, the goodness of an explanation result needs to be defined. We base our argument for goodness on the finding that human beings, among other things, base their recognition of scenes on objects contained therein [46–48]. The more a model's prediction is based on understandable concepts as revealed by an explanation, the more the model is considered explainable [49, 50]. Therefore, if a model is shown to base its predictions on objects in an input scene similarly to human beings, these predictions are inherently explainable, independently of their correctness. Thus, importantly, objects correspond to all distinct entities in the image, irrespective of the model's task. Consequently, we define goodness to be the alignment of the visual explanation map with objects. We evaluate explainability as the extent to which a model is in a state such that running sound explanation methods produces this type of explanation results.

3.2 Object-based explainability (ObEy) metric

ObEy metric. To evaluate the extent to which saliency maps generated by explanation methods align with definable objects, we rely on annotations in the form of instance segmentation masks. We impose the assumption that these masks capture the majority of individually recognizable objects in the scene. This allows us to deduce that a high intensity of the saliency map inside a segmentation mask implies model explainability, while the opposite implies a lack thereof.

On a sample, we compute an intensity-weighted intersection over union (wIoU) between the saliency map and the per-object segmentation mask, for all objects in the image and the saliency maps from all predicted labels. For two pixel-aligned vectors $p, q \in [0,1]^K$ of K components each, the wIoU is given by

$$\frac{p \cap_w q}{p \cup_w q} = \frac{\sum_{j=1}^{K} p_j q_j}{\sum_{j=1}^{K} \max(p_j, q_j)}. \tag{1}$$

The weighted IoU is desirable due to the intensity disparity of certain saliency maps. The explanations of Grad-CAM tend to focus with high intensity on small image regions and with increasingly lower intensity on larger regions outside the focus. Hence, weighing by the intensity mitigates measurement distortions that would occur if the entire area of non-zero saliency contributed equally to the IoU as the few most salient pixels. A saliency map attributing uniformly maximal intensity within one object and zero outside it would thus maximize the wIoU.

For each predicted label for a single image, we retain the maximum wIoU of the corresponding saliency map with any object, since basing the prediction on any one object is sufficient to be explainable, regardless of that object's subjective relevance or correctness for the prediction (Section 3.1). Even for labels that correspond to groups of objects, the maximum with any constituent object will be reflected. If the mean over objects was used instead, models that focus on only one object in an image with many objects and annotations would be penalized.

The score of a sample is the mean of the max-wIoUs over all predictions. Finally, the Object-based Explainability (ObEy) of a model is the mean of the scores for all test samples (Fig. 1). For an explanation-method-compatible neural network Ψ the ObEy score is thus given by

$$e_{X_{test}}(\Psi) = \frac{1}{|X_{test}|} \sum_{1 \leq i \leq |X_{test}|} \left(\frac{1}{|S_i|} \sum_{s_m \in S_i} \left[\max_{a \in A_i} \frac{s_m \cap_w a}{s_m \cup_w a} \right] \right), \tag{2}$$

where i is the index of an image x_i from the test distribution X_{test}, s_m is a saliency map by its pixels with intensities in $[0, 1]$ generated by a post-hoc explanation method m, S_i is the set of saliency maps for image i with magnitude as large as the number of predictions from $\Psi(x_i)$, and A_i is the set of annotations for image i as object segmentations transformed into binary maps.

Generating segmentation masks. ObEy depends on instance segmentation masks, which are not readily available in practice. Even segmentation-mask-annotated datasets usually only mask objects pertaining to the classes of the prediction task, not all the objects in the sample. [4] recently released Segment-Anything (SAM), a foundation model that segments images in a zero-shot manner. We leverage SAM to generate all-object instance segmentation masks, enabling the use of ObEy in any setting. We demonstrate the efficacy of SAM by comparing SAM-generated masks against ground-truth masks in A.5. To our knowledge, this is the first explainability-oriented use of SAM, offering a new direction of application for foundation models.

Explanation method. Any explanation method producing saliency maps can be used interchangeably in the ObEy framework. In the following demonstrations, we use Grad-CAM due its spread, its positive evaluation of reflecting the model internals [30], and its dense saliency maps.

4 Applications

The ObEy evaluation gives rise to immediate downstream use cases for researchers and practitioners (Figure 1), which we broadly group into use cases to compare different models regarding their explainability and investigative use cases to identify explainability properties of a given model.

Comparative use cases are those in which two or more models, for instance from different architectures, or multiple model conditions, resulting from different training procedures and stages, are compared in their ObEy score to draw conclusions about the effect of these differences on the explainability. Based on this insight, the quantification allows to select one model for subsequent steps taking into account its explainability, or to refine the less-explainable models accordingly. Here we demonstrate this use case to quantify the effect of adversarial training on the explainability between two otherwise equal models. Additionally, we inspect the explainability of under- and overfit models compared to well-fit ones.

Investigative use cases are those in which breaking down the explainability on the reference dataset may reveal new insights about the model's predictive behavior on its task, beyond the performance metric. These can be used to improve the model. For instance, analyzing the model's explainability between different data segments, such as by user groups or sample difficulty, offers novel insights on the deficits of the model. Here, we investigate a model by breaking down its explainability based on its predictive performance.

5 Experimental Setup

We conduct experiments on four datasets: MNIST [51] trained on a simple CNN inspired by [52], and Oxford-IIIT Pet [53], CUB-200-2011 [54], and Pascal-VOC [55], all trained on ResNet18 [56]. As Pascal-VOC is a multi-label classification dataset, we additionally engineer a custom MNIST multi-label (MNIST-ML) dataset by stacking every four samples to square images to assess multi-label performance more broadly. For MNIST-ML, we train a CNN with more convolutional layers. We use PGD [57] as adversarial attack both for assessing the adversarial robustness of the models and for adversarial training. We employ L_2 and L_∞ attacks on all models and datasets.

To simulate overfitting, we do not consider any validation set and choose the best model solely based on training performance. We reduce the training set to include $\frac{1}{10}$ of the instances per class,

and train for twice the number of epochs to ensure the highest training performance possible. To underfit models, we train for a single epoch on each dataset. Details about the datasets, models, attacks, and hyperparameters can be found in the supplementary material A.7.

6 Results and Discussion

6.1 Comparative use cases

Compare training techniques: Adversarial training's effect on explainability. We present the ObEy score for normally and adversarially trained models using PGD with L_2 and L_∞ perturbations in Table 1. Details on the classification performance can be found in Table 6.

In Table 1, it can be observed that the ObEy scores increase significantly for the single-label datasets for both perturbations, where for CUB-200-2011 the size is marginally lower. This indicates that adversarially trained models focus more on the segmented objects in the scene or attribute higher saliency intensities to the overlapping regions, compared to regularly trained models. The latter is reflected in the changes in intensity in Table 4. This effect is also visible in Figure 3, where the intensities are higher for the adversarially trained models in the last two columns as compared to the first two. The shapes of the saliency maps change less uniformly, sometimes contributing to larger or smaller overlaps. These changes may also lead to a change of the corresponding object part, indicating that adversarial training can shift the focus to more precise parts. For L_∞ training on CUB-200-2011, the mean intensity decreases while the better aligned shapes lead to a total ObEy improvement, which also holds for Pascal-VOC.

For multi-label datasets, it can be observed in Table 1 that in most cases the ObEy decreases after adversarial training. Notably, except for MNIST-ML L_∞, this outcome is driven by the decrease in intensity (Tables 4,12). In contrast, the shapes of the saliency maps, as visible in Figure 2, in majority become more overlapping with the maximizing object segmentation masks.

This suggests that adversarial training affects multi-label classifiers differently than single-label classifiers. In the majority of cases, for the former, adversarial training updates the CNN weights such that activations weighted by the gradients decrease, whereas the opposite holds for single-label classifiers. Overall, the increase in ObEy for single-label datasets provides new quantitative evidence for visually observed phenomena in [16, 17]. For multi-label datasets, the decrease in intensity prevents an increase in ObEy, despite more object-aligned saliency maps in most cases.

Table 1: ObEy scores for regularly trained and adversarially trained models.

Dataset	Model	Attack	Normal	Adv.Train	Δ	Rel. Δ
MNIST	SimpleCNN	L_∞	0.0016	0.0029	0.0013	81.3%
MNIST	SimpleCNN	L_2	0.0016	0.0031	0.0015	93.8%
MNIST-ML	DeepCNN	L_∞	0.0065	0.0028	-0.0037	-56.9%
MNIST-ML	DeepCNN	L_2	0.0065	0.0017	-0.0048	-73.8%
Oxford-IIIT Pet	ResNet18	L_∞	0.1963	0.3314	0.1351	68.8%
Oxford-IIIT Pet	ResNet18	L_2	0.1963	0.3121	0.1158	59.0%
CUB-200-2011	ResNet18	L_∞	0.1211	0.1377	0.0166	13.7%
CUB-200-2011	ResNet18	L_2	0.1211	0.2162	0.0951	78.5%
Pascal-VOC	ResNet18	L_∞	0.1014	0.1050	0.0036	3.6%
Pascal-VOC	ResNet18	L_2	0.1014	0.0878	-0.0136	-13.4%

Compare training stages: Under- and overfitting's effect on explainability. The ObEy scores for under- and overfit models compared to well-trained models are presented in Table 2. The ObEy decreases significantly for all under- and overfit models, except for the overfit MNIST-ML and Pascal-VOC models. The decrease indicates that, when insufficiently or too much trained on the training distribution, models fail to capture the object-related parts of the images in the test distribution and produce lower-intensity saliency maps. In the case of the multi-label datasets, the ObEy improvement suggests that, due to the additional complexity, training beyond test performance satisfaction may still improve the ability to focus on objects and make predictions explainable, without aiding the prediction task. Using the ObEy insight thus provides an additional means to assess the fitness, and trade-offs, of the models during development.

Table 2: ObEy scores for over- and underfit models as compared to a well-trained models.

Dataset	Model	Fit	Normal Training	Post Fit	Δ	Rel. Δ
MNIST	SimpleCNN	Over	0.0016	0.0009	-0.0007	-43.8%
MNIST	SimpleCNN	Under	0.0016	0.0006	-0.0010	-62.5%
MNIST-ML	DeepCNN	Over	0.0065	0.0081	0.0016	24.6%
MNIST-ML	DeepCNN	Under	0.0065	0.0015	-0.0050	-76.9%
Oxford-IIIT Pet	ResNet18	Over	0.1963	0.0959	-0.1004	-51.1%
Oxford-IIIT Pet	ResNet18	Under	0.1963	0.0649	-0.1314	-66.9%
CUB-200-2011	ResNet18	Over	0.1211	0.0506	-0.0705	-58.2%
CUB-200-2011	ResNet18	Under	0.1211	0.1039	-0.0172	-14.2%
Pascal-VOC	ResNet18	Over	0.1014	0.1148	0.0134	13.2%
Pascal-VOC	ResNet18	Under	0.1014	0.0538	-0.0476	-46.9%

6.2 Investigative use case

Investigate performance: Explainability on correct vs. incorrect predictions. We investigate the reasons for model performance by breaking down the explainability between correct and incorrect predictions. Additional details regarding the setup are available in A.9. In Table 3, it can be seen that for Oxford-IIIT Pet, CUB-200-2011, and Pacal-VOC, the wrong predictions also have lower ObEy scores. Hence, regardless of whether the focus is on the correct or incorrect object, it is weaker than the object focus in the correctly predicted samples. Likely, the low performance thus stems from lower activations from those samples, indicative of higher uncertainty or the inability to recognize features in the sample. Conversely, the model did not learn a wrong short cut, such as predicting sheep based on grass, on which it would rely too much.

In contrast, the ObEy increase for wrong predictions for MNIST-ML suggests that the model focuses indeed on objects, but in an ineffective way: Either the focus is strongly on the wrong object, whose misidentification leads to the prediction of a wrong class. Or, the focus is strongly on the correct object, but the wrong prediction is made, implying that something within the object is recognized differently than in the correct samples. Since there are only prediction-relevant objects in MNIST-ML (digits), the latter is the case.

Table 3: Performance and total, correct-prediction, and incorrect-prediction ObEy of models. Performance is measured in accuracy for single-label datasets and mAP [55] for multi-label.

Dataset	Performance	Overall	Correct	Incorrect	Rel. Δ Correct	Rel. Δ Incorrect
MNIST	97.0%	0.0016	0.0016	0.0017	0.0%	6.2%
MNIST-ML	98.8%	0.0065	0.0063	0.0074	-3.1%	13.8%
Oxford-IIIT Pet	89.0%	0.1963	0.2032	0.1405	3.5%	-28.4%
CUB-200-2011	69.5%	0.1211	0.1295	0.1021	6.9%	-15.7%
Pascal-VOC	79.4%	0.0959	0.1013	0.0599	5.6%	-37.6%

7 Conclusion

We discuss the notion of explainability in contrast to explanation methods and propose ObEy as a novel metric to quantify the explainability of a model. Inspired by humans' perception of scenes, ObEy evaluates the model's ability to produce saliency maps that are aligned with any definable objects in images, using segmentation masks. We circumvent the unavailability of these masks in practice by using the SAM model, enabling ObEy to be used with any vision model and dataset.

We show how ObEy can be applied to practical downstream use cases. Comparing the explainability between training stages may reveal how features between the training and test distribution are recognized differently over training time. Comparisons between training techniques may expose which ones lead to more explainable models, irrespective of performance. In particular, we find that adversarial training on single-label image classifiers improves explainability. Finally, breaking down explainability by segments in the reference distribution may be useful to investigate and debug the model's behavior on poor-performing segments. We are hopeful to see future work employing ObEy to compare a wide range of models and to optimize for explainability.

References

[1] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," pp. 618–626, 2017.

[2] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2017.

[3] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, e0130140, 2015.

[4] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," pp. 4015–4026, Oct. 2023.

[5] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[6] A. Jobin, M. Ienca, and E. Vayena, "The global landscape of ai ethics guidelines," *Nature machine intelligence*, vol. 1, no. 9, pp. 389–399, 2019.

[7] N. Maslej, L. Fattorini, E. Brynjolfsson, J. Etchemendy, K. Ligett, T. Lyons, J. Manyika, H. Ngo, J. C. Niebles, V. Parli, *et al.*, "Artificial intelligence index report 2023," *arXiv preprint arXiv:2310.03715*, 2023.

[8] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. Moura, and P. Eckersley, "Explainable machine learning in deployment," in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 648–657.

[9] I. Ahmed, G. Jeon, and F. Piccialli, "From artificial intelligence to explainable artificial intelligence in industry 4.0: A survey on what, how, and where," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5031–5042, 2022.

[10] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds., pp. 3319–3328, Jun. 2017.

[11] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: Removing noise by adding noise," 2017.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," KDD '16, pp. 1135–1144, 2016.

[13] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," 2017.

[14] H. Chen, S. M. Lundberg, and S.-I. Lee, "Explaining a series of models by propagating shapley values," *Nature Communications*, vol. 13, no. 1, p. 4512, Aug. 2022.

[15] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2015.

[16] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb, "On the connection between adversarial robustness and saliency map interpretability," 2019.

[17] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," 2019.

[18] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[19] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," *Information Fusion*, vol. 76, pp. 89–106, 2021.

[20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., pp. 818–833, 2014.

[21] C. Agarwal and A. Nguyen, "Explaining image classifiers by removing input features using generative models," Nov. 2020.

[22] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017.

[23] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," Oct. 2017.

[24] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," 2019.

[25] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," 2018.

[26] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital signal processing*, vol. 73, pp. 1–15, 2018.

[27] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[28] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," 2018.

[29] A. Warnecke, D. Arp, C. Wressnegger, and K. Rieck, "Evaluating explanation methods for deep learning in security," pp. 158–174, 2020.

[30] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems*, vol. 31, 2018.

[31] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *The Journal of Machine Learning Research*, vol. 11, pp. 1803–1831, 2010.

[32] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2014.

[33] A. Hedström, L. Weber, D. Krakowczyk, D. Bareeva, F. Motzkus, W. Samek, S. Lapuschkin, and M. M. M. Höhne, "Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond," *Journal of Machine Learning Research*, vol. 24, no. 34, pp. 1–11, 2023.

[34] Z. Q. Lin, M. J. Shafiee, S. Bochkarev, M. S. Jules, X. Y. Wang, and A. Wong, "Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms," *arXiv preprint arXiv:1910.07387*, 2019.

[35] A. Ghorbani, A. Abid, and J. Zou, "Interpretation of neural networks is fragile," vol. 33, no. 01, pp. 3681–3688, 2019.

[36] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, "A benchmark for interpretability methods in deep neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[37] Y. Rong, T. Leemann, V. Borisov, G. Kasneci, and E. Kasneci, "A consistent and efficient evaluation strategy for attribution methods," *arXiv preprint arXiv:2202.00449*, 2022.

[38] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, "Evaluating the visualization of what a deep neural network has learned," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

[39] R. Tomsett, D. Harborne, S. Chakraborty, P. Gurram, and A. Preece, "Sanity checks for saliency metrics," vol. 34, no. 04, pp. 6021–6029, 2020.

[40] L. Arras, A. Osman, and W. Samek, "Clevr-xai: A benchmark dataset for the ground truth evaluation of neural network explanations," *Information Fusion*, vol. 81, pp. 14–40, 2022.

[41] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *International Journal of Computer Vision*, vol. 126, no. 10, pp. 1084–1102, Oct. 2018.

[42] O. Le Meur and T. Baccino, "Methods for comparing scanpaths and saliency maps: Strengths and weaknesses," *Behavior research methods*, vol. 45, no. 1, pp. 251–266, 2013.

[43] J. Oramas, K. Wang, and T. Tuytelaars, "Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks," *arXiv preprint arXiv:1712.06302*, 2017.

[44] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller, "Unmasking clever hans predictors and assessing what machines really learn," *Nature communications*, vol. 10, no. 1, p. 1096, 2019.

[45] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[46] I. Biederman, "Perceiving real-world scenes," *Science*, vol. 177, no. 4043, pp. 77–80, 1972.

[47] J. M. Henderson and A. Hollingworth, "High-level scene perception," *Annual review of psychology*, vol. 50, no. 1, pp. 243–271, 1999.

[48] R. A. Epstein and C. I. Baker, "Scene perception in the human brain," *Annual Review of Vision Science*, vol. 5, no. 1, pp. 373–397, 2019.

[49] U. Ehsan, P. Tambwekar, L. Chan, B. Harrison, and M. O. Riedl, "Automated rationale generation: A technique for explainable ai and its effects on human perceptions," IUI '19, pp. 263–274, 2019.

[50] N. Tintarev and J. Masthoff, "Designing and evaluating explanations for recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 479–510.

[51] Y. LeCun, C. Cortes, and C. Burges, "The mnist database of handwritten digits,," 2010.

[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[53] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "The oxford-iiit pet dataset,"

[54] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "Caltech-ucsd birds-200-2011," no. CNS-TR-2011-001, 2011.

[55] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[57] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2018.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," pp. 248–255, 2009.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

A Appendix

A.1 Results (comparative): Grad-CAM saliency maps and SAM segmentation mask for normal vs. adversarially robust models.

Figure 2 and Figure 3 show saliency maps for one Pascal-VOC and two Oxford-IIIT Pet samples respectively, all produced using Grad-CAM on ResNet18 when targeting the predicted class. The second and fourth rows depict the SAM segmentation masks that maximize the wIoUs of the given saliency maps. The first and third columns show saliency maps explaining predictions on regular samples, while the second and fourth columns show saliency maps generated on inputs perturbed using PGD L_∞ attacks. The first two columns show a regularly trained model, while the last two columns show a model that was adversarially trained on L_∞ attacks.

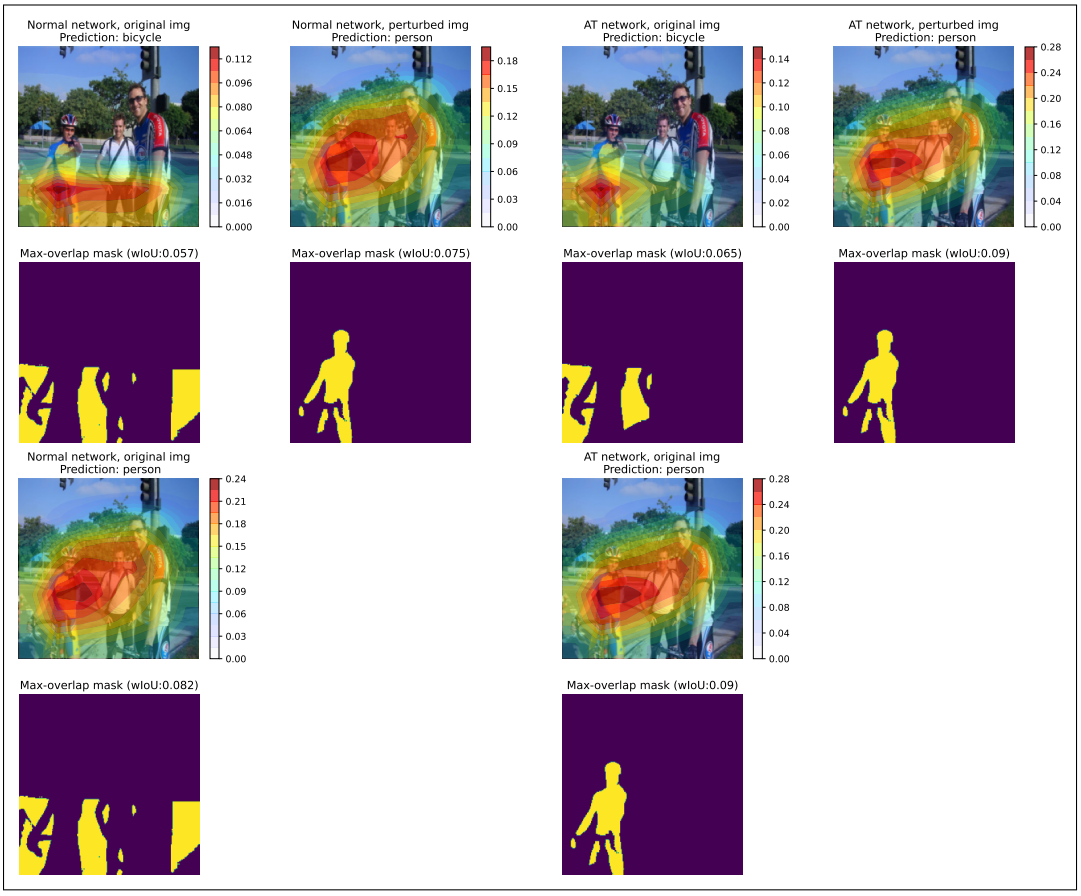

Figure 2: **Pascal-VOC sample after adversarial training:** Saliency maps for all predictions produced by regularly trained (left two columns) and L_∞-trained (right two columns) ResNet18 using Grad-CAM targeting the predicted class as well as segmentation masks of object with highest wIoU, for one Pascal-VOC sample. It can be observed that in response to adversarial training, the intensity increases and the shape of the center of focus changes between the same-predicted classes. This may also invoke the assignment of a new object that maximizes the wIoU for the prediction. Apart, adversarial samples (columns 2, 4) receive fewer predictions. Note that ObEy only considers the non-perturbed samples.

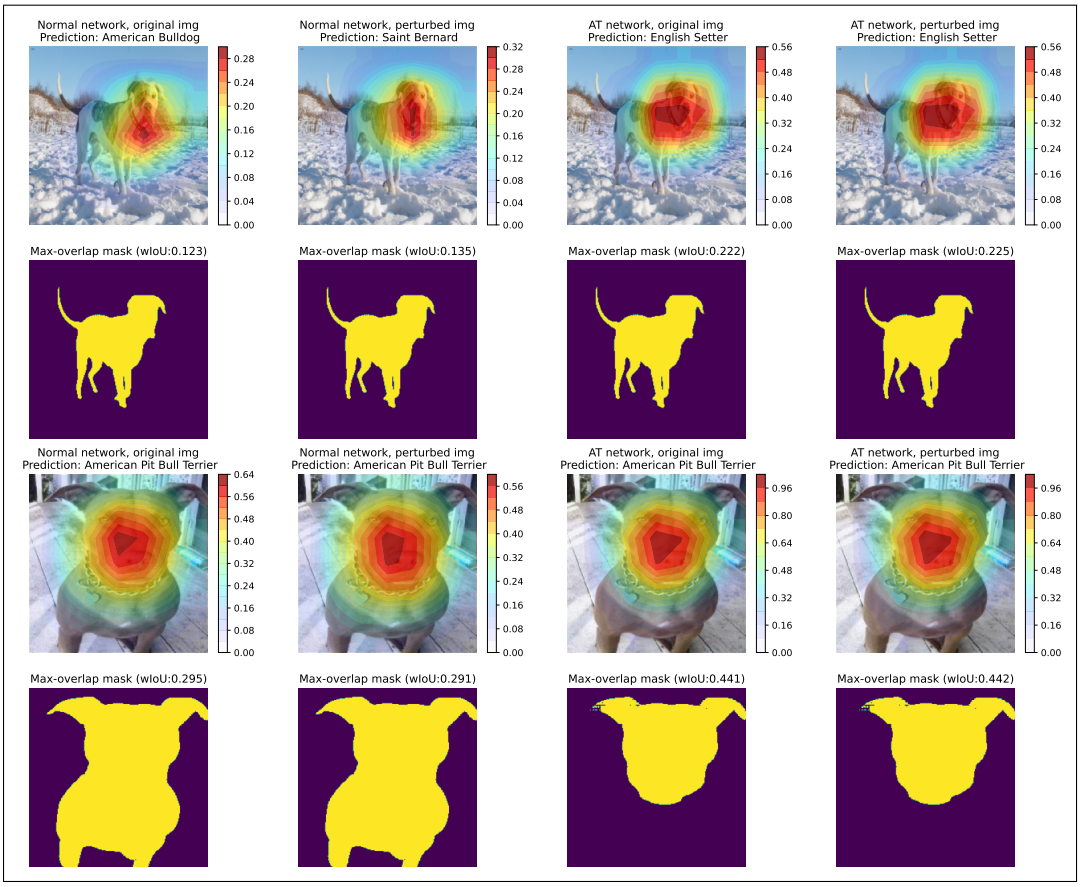

Figure 3: **Oxford-IIIT samples after adversarial training:** Saliency maps for the single-label prediction produced by regularly trained (left two columns) and L_∞-trained (right two columns) ResNet18 using Grad-CAM targeting the predicted class as well as segmentation masks of object with highest wIoU, for two Oxford-IIIT Pet samples. After adversarial training, the saliency intensities increase significantly and the shapes of the focus of saliency become more aligned with the body parts of the dogs. For the second sample, this additionally causes the assignment of a smaller body part whose mask maximizes the wIoU with the new saliency map. Note that ObEy only considers the non-perturbed samples.

A.2 Results (comparative): Mean saliency map intensities

Table 4 and Table 5 show the mean intensities for the saliency maps generated by each model over the whole dataset.

Table 4: Mean intensity of saliency maps for regularly trained and adversarially trained models.

Dataset	Model	Attack	Normal	Adv. Train	Δ	Rel. Δ
MNIST	SimpleCNN	L_∞	0.0008	0.0012	0.0004	43.2%
MNIST	SimpleCNN	L_2	0.0008	0.0013	0.0005	58.2%
MNIST-ML	DeepCNN	L_∞	0.0013	0.0006	-0.0007	-56.0%
MNIST-ML	DeepCNN	L_2	0.0013	0.0003	-0.0010	-79.0%
Oxford-IIIT Pet	ResNet18	L_∞	0.1307	0.2176	0.0869	66.5%
Oxford-IIIT Pet	ResNet18	L_2	0.1307	0.2048	0.0742	56.8%
CUB-200-2011	ResNet18	L_∞	0.1177	0.1012	-0.0165	-14.1%
CUB-200-2011	ResNet18	L_2	0.1177	0.2058	0.0881	74.8%
Pascal-VOC	ResNet18	L_∞	0.0739	0.0733	-0.0006	-0.8%
Pascal-VOC	ResNet18	L_2	0.0739	0.0630	-0.0109	-14.8%

Table 5: Mean intensity of saliency maps for well-trained models compared to over- and underfit models.

Dataset	Model	Fit	Normal Training	Post Fit	Δ	Rel. Δ
MNIST	SimpleCNN	Overfit	0.0008	0.0001	-0.0007	-83.7%
MNIST	SimpleCNN	Underfit	0.0008	0.0003	-0.0006	-69.1%
MNIST-ML	DeepCNN	Overfit	0.0013	0.0018	0.0005	38.9%
MNIST-ML	DeepCNN	Underfit	0.0013	0.0004	-0.0010	-72.4%
Oxford-IIIT Pet	ResNet18	Overfit	0.1307	0.0558	-0.0748	-57.3%
Oxford-IIIT Pet	ResNet18	Underfit	0.1307	0.0386	-0.0920	-70.5%
CUB-200-2011	ResNet18	Overfit	0.1177	0.0396	-0.0781	-66.3%
CUB-200-2011	ResNet18	Underfit	0.1177	0.0745	-0.0432	-36.7%
Pascal-VOC	ResNet18	Overfit	0.0739	0.0873	0.0134	18.2%
Pascal-VOC	ResNet18	Underfit	0.0739	0.0410	-0.0329	-44.6%

A.3 Results (comparative): Predictive performance after adversarial training

Table 4 shows the predictive performance of normally and adversarially trained models. We note that the performance of the perturbed samples on a normal model is non-zero. This is because we restrict the perturbation sizes to enforce the constraint that images shall still be realistic after perturbation. That is, the perturbation should be minimally perceivable. Two examples of perturbed images are shown in Figure 4.

Table 6: Training and robustness performance on normal and adversarially trained models. MNIST-ML and Pascal-VOC are measured in mAP [55], while the other datasets are measured in accuracy. AT denotes adversarial training

Dataset	Model	Attack	Non-pert.	Pert.	AT+Non-pert.	AT+Pert.
MNIST	SimpleCNN	L_∞	0.99	0.16	0.96	0.70
MNIST	SimpleCNN	L_2	0.99	0.02	0.97	0.80
MNIST-ML	DeepCNN	L_∞	0.98	0.40	0.98	0.90
MNIST-ML	DeepCNN	L_2	0.98	0.59	0.97	0.82
Oxford-IIIT Pet	ResNet18	L_∞	0.88	0.40	0.87	0.70
Oxford-IIIT Pet	ResNet18	L_2	0.88	0.14	0.87	0.78
CUB-200-2011	ResNet18	L_∞	0.69	0.22	0.67	0.54
CUB-200-2011	ResNet18	L_2	0.69	0.14	0.66	0.46
Pascal-VOC	ResNet18	L_∞	0.80	0.24	0.76	0.61
Pascal-VOC	ResNet18	L_2	0.80	0.24	0.59	0.49

A.4 Results (investigative): ObEy breakdown by class

As an additional investigative use case of ObEy, a model can be inspected by breaking down its explainability by the class membership of samples. Table 7 shows the accuracy and ObEy score for each class of the Oxford-IIIT Pet dataset, and the ObEy divided by accuracy. Table 8 shows the AP and ObEy score for each class of the Pascal-VOC dataset, and the ObEy divided by AP.

Table 7: Accuracy and ObEy for each class computed for Oxford-IIIT Pet.

Class	Accuracy	ObEy	ObEy / Accuracy
Abyssinian	0.91	0.1797	0.1979
American Bulldog	0.88	0.1848	0.2100
American Pit Bull Terrier	0.52	0.1574	0.3027
Basset Hound	0.87	0.1972	0.2266
Beagle	0.94	0.2033	0.2163
Bengal	0.93	0.1834	0.1972
Birman	0.81	0.2160	0.2667
Bombay	1.00	0.2196	0.2196
Boxer	0.91	0.1942	0.2136
British Shorthair	0.80	0.2146	0.2683
Chihuahua	0.86	0.2006	0.2332
Egyptian Mau	0.84	0.2094	0.2508
English Cocker Spaniel	0.95	0.2359	0.2483
English Setter	0.85	0.1955	0.2300
German Shorthaired	0.99	0.1982	0.2002
Great Pyrenees	0.94	0.1829	0.1946
Havanese	0.95	0.1719	0.1809
Japanese Chin	1.00	0.2178	0.2178
Keeshond	0.98	0.2056	0.2098
Leonberger	0.97	0.2019	0.2081
Maine Coon	0.82	0.1651	0.2013
Miniature Pinscher	0.92	0.1918	0.2084
Newfoundland	0.97	0.1862	0.1920
Persian	0.83	0.2054	0.2475
Pomeranian	0.91	0.2091	0.2298
Pug	0.88	0.1966	0.2234
Ragdoll	0.68	0.1751	0.2576
Russian Blue	0.83	0.1869	0.2251
Saint Bernard	0.98	0.2046	0.2088
Samoyed	0.97	0.2037	0.2100
Scottish Terrier	0.99	0.2269	0.2292
Shiba Inu	0.97	0.1989	0.2051
Siamese	0.89	0.2197	0.2468
Sphynx	0.92	0.1555	0.1690
Staffordshire Bull Terrier	0.56	0.1524	0.2712
Wheaten Terrier	0.92	0.1904	0.2070
Yorkshire Terrier	0.98	0.2237	0.2283

Table 8: AP and ObEy for each class computed for Pascal-VOC.

Class	AP	ObEy	ObEy / AP
Aeroplane	0.95	0.0936	0.0986
Bicycle	0.80	0.0974	0.1225
Bird	0.92	0.0823	0.0896
Boat	0.86	0.0960	0.1119
Bottle	0.58	0.0790	0.1358
Bus	0.92	0.1244	0.1352
Car	0.79	0.0870	0.1106
Cat	0.93	0.1401	0.1503
Chair	0.70	0.0642	0.0923
Cow	0.68	0.0734	0.1085
Diningtable	0.60	0.0680	0.1132
Dog	0.87	0.0998	0.1154
Horse	0.80	0.0896	0.1124
Motorbike	0.86	0.1138	0.1321
Person	0.95	0.0993	0.1043
Pottedplant	0.53	0.0610	0.1156
Sheep	0.82	0.0955	0.1169
Sofa	0.59	0.0697	0.1186
Train	0.92	0.1227	0.1336
Tvmonitor	0.84	0.1120	0.1330

A.5 Results: ObEy from SAM-generated vs. ground-truth segmentation masks

We compare the ObEy scores computed using SAM-generated segmentation masks against human-labeled ground-truth (GT) segmentation masks where those are provided among the chosen datasets. Additionally, we manually engineer the segmentation masks for the two MNIST datasets by thresholding objects found in the image into binary masks. This exclusively serves informational purposes, since in the case of the natural-image datasets, the ground-truth masks do not capture all recognizable objects in the scene and instead only segment the objects that correspond to the classes to predict. Since we argue that explainability is independent of correctness, and basing a prediction on an intuitively irrelevant but recognizable object is as explainable as basing it on the correct object, these ground-truth masks do not fully satisfy our requirement to be used in the explainability metric.

Tables 9, 10, and 11 show ObEy scores computed using both SAM segmentation masks and ground-truth segmentation masks. Table 9 shows the normally trained models, while Table 10 shows adversarially trained models using L_2 and L_∞ PGD attacks, and Table 11 shows over -and underfit models. Here, the SAM-based and ground-truth-based ObEy scores for Pascal-VOC use a different, smaller test split of the Pascal-VOC dataset for which the ground-truth masks are provided, instead of the main test split.

For MNIST and MNIST-ML, in several cases, we observe a noticeable discrepancy above 15% from the ObEy scores computed on SAM masks to the manually generated ground-truth masks. This is especially evident for MNIST-ML, but less for MNIST, after adversarial training and over- and underfitting. This may be indicative of inadequate SAM masks only for some of the MNIST samples. Since these images have a low resolution and no natural foreground and background, SAM predicts too many objects that are only parts of the digit, leading to a lower ObEy.

For the three datasets containing natural images, Oxford-IIIT Pet, CUB-200-2011, and Pascal-VOC, all ObEy scores for the SAM segmentation masks are within 10% of the ground-truth segmentation masks, with the exception of one experiment. Within each dataset, the relative difference in ObEy between the two different mask types always has the same signs, again with the exception of a single experiment. This indicates a proportional difference between the masks. Furthermore, the sign of the delta between normally and adversarially trained models is the same across both mask types for each dataset. This also holds for the deltas from normal fit to under- and over-fit, respectively, with the exception of CUB-200-2011 underfit.

This correspondence between SAM masks and ground-truth masks indicates that many of the saliency maps indeed focus on the objects segmented in the ground-truth annotations. These,

however, would not be able to establish the explainability from short-cut [44, 45] or more fine-grained object part learned predictions. Consequently, all-object masks produced by SAM remain preferred due to their complete segmentation of the image, as necessitated by our definition of explainability.

Table 9: ObEy scores comparison between SAM segmentation masks and ground-truth segmentation masks for regularly trained models.

Dataset	Model	Normal Train (SAM Mask)	Normal Train (GT Mask)	Rel. Diff.
MNIST	SimpleCNN	0.0016	0.0013	-18.8%
MNIST-ML	DeepCNN	0.0065	0.0048	-26.2%
Oxford-IIIT Pet	ResNet18	0.1963	0.1860	-5.5%
CUB-200-2011	ResNet18	0.1211	0.1262	4.0%
Pascal-VOC	ResNet18	0.1006	0.0927	-7.9%

Table 10: ObEy scores comparison between SAM segmentation masks and ground-truth segmentation masks for adversarially trained models.

Dataset	Model	Attack	Adv.Train (SAM Mask)	Adv. Train (GT Mask)	Rel. Diff.
MNIST	SimpleCNN	L_∞	0.0029	0.0031	6.9%
MNIST	SimpleCNN	L_2	0.0031	0.0033	6.5%
MNIST-ML	DeepCNN	L_∞	0.0028	0.0019	-32.1%
MNIST-ML	DeepCNN	L_2	0.0017	0.0013	-23.5%
Oxford-IIIT Pet	ResNet18	L_∞	0.3314	0.3066	-7.5%
Oxford-IIIT Pet	ResNet18	L_2	0.3121	0.2895	-7.2%
CUB-200-2011	ResNet18	L_∞	0.1377	0.1465	6.4%
CUB-200-2011	ResNet18	L_2	0.2162	0.2341	8.3%
Pascal-VOC	ResNet18	L_∞	0.1066	0.0973	-8.7%
Pascal-VOC	ResNet18	L_2	0.0868	0.0761	-12.3%

Table 11: ObEy scores comparison between SAM segmentation masks and ground-truth segmentation masks for over -and underfit models

Dataset	Model	Fit	Post Fit (SAM Mask)	Post Fit (GT Mask)	Rel. Diff.
MNIST	SimpleCNN	Over	0.0009	0.0010	11.1%
MNIST	SimpleCNN	Under	0.0006	0.0006	0.0%
MNIST-ML	DeepCNN	Over	0.0081	0.0057	-29.6%
MNIST-ML	DeepCNN	Under	0.0015	0.0010	-33.3%
Oxford-IIIT Pet	ResNet18	Over	0.0959	0.0908	-5.3%
Oxford-IIIT Pet	ResNet18	Under	0.0649	0.0607	-6.5%
CUB-200-2011	ResNet18	Over	0.0506	0.0459	-9.3%
CUB-200-2011	ResNet18	Under	0.1039	0.1072	3.2%
Pascal-VOC	ResNet18	Over	0.1167	0.1051	-9,9%
Pascal-VOC	ResNet18	Under	0.0546	0.0510	-5.5%

A.6 Results: Unweighted-ObEy (IoU) between Saliency Maps and SAM Segmentation masks

We compute the mean unweighted IoU between the saliency maps and SAM segmentation masks over all samples to assess the unweighted object-alignment of the shapes of the salient regions. Note that we still only consider the single segmentation mask with the maximum IoU to the saliency map when computing the score per image. Tables 12, 13, and 14 show the mean unweighted IoU for each of the three main use cases considered in this work.

For the comparative use cases in Tables 12 and 13, we observe that the relative difference between the normal and adjusted model states are all considerably smaller for the datasets containing natural images. Comparing these results with the weighted ObEy scores in Tables 1 and 2, we notice that the difference in relative Δ between the unweighted IoU and weighted ObEy scores are significant. This further confirms that changing model states affect the intensities of the saliency maps more than the covered area. We note that the changes in assignment of wIoU-maximizing object per saliency map affects these deltas.

Table 12: Mean unweighted IoU for regularly trained and adversarially trained models.

Dataset	Model	Attack	Normal	Adv.Train	Δ	Rel. Δ
MNIST	SimpleCNN	L_∞	0.4042	0.5204	0.1162	28.7%
MNIST	SimpleCNN	L_2	0.4042	0.5687	0.1645	40.7%
MNIST-ML	DeepCNN	L_∞	0.2508	0.2414	-0.0094	-3.7%
MNIST-ML	DeepCNN	L_2	0.2508	0.3562	0.1054	42.0%
Oxford-IIIT Pet	ResNet18	L_∞	0.5153	0.5427	0.0274	5.3%
Oxford-IIIT Pet	ResNet18	L_2	0.5153	0.5435	0.0282	5.5%
CUB-200-2011	ResNet18	L_∞	0.6544	0.6568	0.0024	0.4%
CUB-200-2011	ResNet18	L_2	0.6544	0.6568	0.0024	0.4%
Pascal-VOC	ResNet18	L_∞	0.4385	0.4411	0.0026	0.6%
Pascal-VOC	ResNet18	L_2	0.4385	0.4447	0.0062	1.4%

Table 13: Mean unweighted IoU for well-trained models compared to over- and underfit models.

Dataset	Model	Fit	Normal Training	Post Fit	Δ	Rel. Δ
MNIST	SimpleCNN	Overfit	0.4042	0.9509	0.5467	135.3%
MNIST	SimpleCNN	Underfit	0.4042	0.3879	-0.0163	-4.0%
MNIST-ML	DeepCNN	Overfit	0.2508	0.2800	0.0292	11.6%
MNIST-ML	DeepCNN	Underfit	0.2508	0.1481	-0.1027	-40.9%
Oxford-IIIT Pet	ResNet18	Overfit	0.5153	0.5150	-0.0003	-0.1%
Oxford-IIIT Pet	ResNet18	Underfit	0.5153	0.5128	-0.0025	-0.5%
CUB-200-2011	ResNet18	Overfit	0.6544	0.6527	-0.0017	-0.3%
CUB-200-2011	ResNet18	Underfit	0.6544	0.6537	-0.0007	-0.1%
Pascal-VOC	ResNet18	Overfit	0.4385	0.4458	0.0073	1.7%
Pascal-VOC	ResNet18	Underfit	0.4385	0.4118	-0.0267	-6.1%

Table 14: Performance and total, correct-prediction, and incorrect-prediction mean unweighted IoU of models. Performance is measured in accuracy for single-label datasets and mAP [55] for multi-label.

Dataset	Performance	Overall	Correct	Incorrect	Rel. Δ Correct	Rel. Δ Incorrect
MNIST	97.0%	0.4042	0.4023	0.4671	-0.5%	15.6%
MNIST-ML	98.8%	0.2456	0.2492	0.1696	1.5%	-30.9%
Oxford-IIIT Pet	89.0%	0.5153	0.5177	0.4958	0.5%	-3.8%
CUB-200-2011	69.5%	0.6544	0.6518	0.6602	-0.4%	0.9%
Pascal-VOC	79.4%	0.4166	0.4243	0.3645	1.8%	-12.5%

A.7 Experimental Setup: Details

A.7.1 Models

The classifier used for experiments on the MNIST dataset is a CNN inspired by [52] with two convolutional layers and linear layers, which we train from scratch. For MNIST-ML, we expand the model to eight convolutional layers to make the model better suited for the more complex task. For the remaining datasets, we use ResNet18 [56] pre-trained on ImageNet [58] and replace the output layer to match the respective number of classes for each dataset. We fine-tune all the networks in two stages using the Adam optimizer [59] with a default learning rate of $1e-3$. First, the final classification layer is fine-tuned while keeping all other layers frozen. Second, the weights of all layers are fine-tuned. For adversarial training, we then additionally train all layers with a learning rate of $1e-4$. In each of the aforementioned stages, the models are trained for 10 epochs, and we use a multi-step learning rate scheduler with a decay of 0.1 after 4 and 6 epochs.

A.7.2 Adversarial attacks setup

We use PGD [57] as the main form of adversarial attack both for assessing the robustness of our models and for adversarial training. We employ both L_∞ and L_2 attacks. For L_∞ attacks, we constrain the adversarial samples to lie within an L_∞ ball of radius $\epsilon = 0.3$ for the MNIST datasets, and $\epsilon = \frac{2}{255}$ for the others. As for the L_2 attacks, we use $\epsilon = 1$ for the two MNIST datasets and $\epsilon = 0.3$ for the remaining datasets.

A.7.3 SAM segmentation mask generation

We use SAM-ViT-Huge [4] to generate segmentation masks for all datasets. We use the default hyperparameters, with the following modifications. We query 56 points per image side for MNIST and 224 for all other datasets. The prediction thresholds are 0.8 for IoU and 0.8 for the stability score. We further apply a custom post-processing to only retain masks of objects that cover an area of at least 1% of the total image area. The predicted masks from this process may overlap.

A.8 Experimental Setup: Perturbed image example

Considering the objective of adversarial samples to produce reliable misclassification while minimally visually perturbing the input to prevent the attack from being observed, we probe different strengths of perturbation to find this balance. Table 2 shows two examples of perturbed images using the hyperparameters from Section A.7.2.

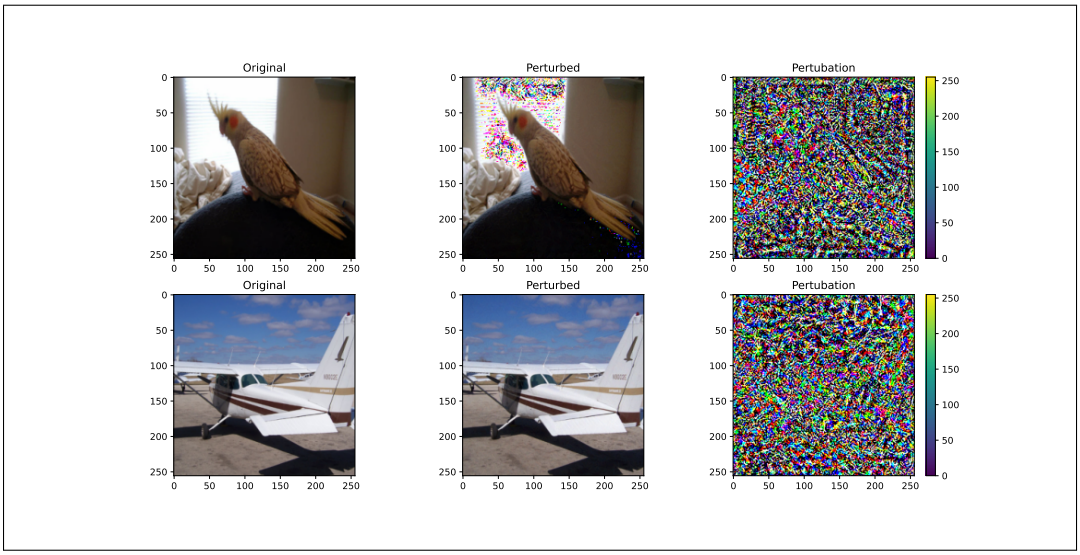

Figure 4: Examples of perturbed images from the Pascal-VOC dataset using PGD L_∞ attack.

A.9 Experimental Setup: Investigative use case details

For the investigative use case, we compare the ObEy score of correct and incorrect predictions. This is unambiguous for single-label datasets. For a multi-label sample, the ObEy score is defined such that it is averaged over all predictions produced for that sample. However, these predictions can simultaneously contain correct and incorrect ones. To break down ObEy by predictive performance for multi-label datasets, we, therefore, compute the ObEy score for each individual prediction and mark it as correct if the prediction corresponds to a class that is labeled in the given image, while predicted classes that do not appear in the label are marked as incorrect. The ObEy score is then computed by the mean over predictions, not over samples.