Optimizing Relevance Maps of Vision Transformers Improves Robustness

Revision 2. July 29th, 2022
Anonymous Author(s)
Affiliation
email

Abstract

It has been observed that visual classification models often rely mostly on the image background, neglecting the foreground, which hurts their robustness to distribution changes. To alleviate this shortcoming, we propose to monitor the model's relevancy signal and manipulate it such that the model is focused on the foreground object. This is done as a finetuning step, involving relatively few samples consisting of pairs of images and their associated foreground masks. Specifically, we encourage the model's relevancy map (i) to assign lower relevance to background regions, (ii) to consider as much information as possible from the foreground, and (iii) we encourage the decisions to have high confidence. When applied to Vision Transformer (ViT) models, a marked improvement in robustness to domain-shifts is observed. Moreover, the foreground masks can be obtained automatically, from a self-supervised variant of the ViT model itself; therefore no additional supervision is required.

1 Introduction

The reliance on simple image-level classification supervision, together with the sampling biases of object recognition datasets, leads to vision models that exhibit unintuitive behavior, as depicted in Fig 1. First, the models we tested (ViT [15], ViT AugReg [54], and DeiT [57]) tend to give disproportional high weight to the background of the image in the decision-making process. Second, the tested models occasionally regard a sparse subset of the pixels in the foreground object for the classification, disregarding much of the object's data. As argued by Geirhos et al. [20], and stated in [27] "image classification datasets contain 'spurious cues' or 'shortcuts' . For instance, cows tend to co-occur with green pastures, and even though the background is inessential to the identity of the object, models may predict 'cow', using primarily the green pasture background cue."

There is considerable evidence that context is a useful cue [34, 43]. However, many of the associated background elements and foreground shortcuts are only relevant to the specific data distribution, which leads to lack of robustness to distribution shifts [40, 47]. There are many methods for overcoming domain shifts, including domain adaptation techniques [17, 2] and methods that augment the training set or the training procedure [29, 41]. In this work, however, we opt for a direct approach, which monitors the relevancy score of the model for each image region, and manipulates the relevancy map to be focused on the regions within the foreground mask.

The method is based on a finetuning procedure, which is applied to a pretrained Vision Transformer (ViT) model. A relatively small set of samples, for which the foreground is given, is employed during this phase. In most of our experiments, we use three samples for half the classes, following work that examined the effect of transfer learning on half of the classes [65]. The ground-truth foreground mask is either human annotated [18], or estimated from a self-supervised ViT model [63].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

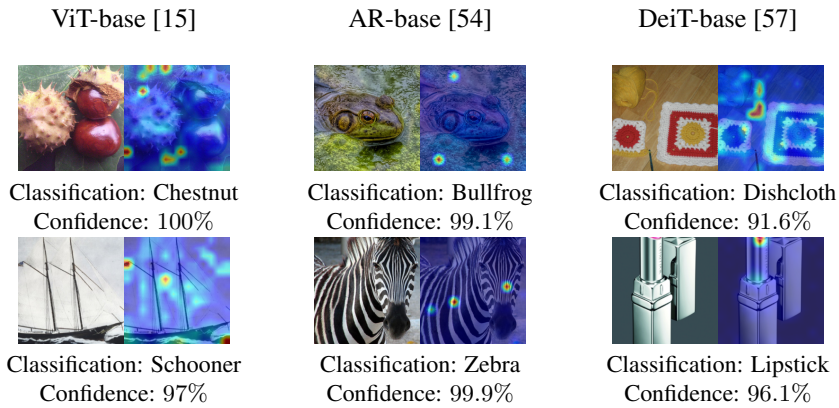


Figure 1: Examples of the salient issues with ViTs. Each pair depicts an input image and its corresponding relevance map. The first row demonstrates examples of background-centered relevance, the second row shows examples of sparse foreground relevance. Both issues occur in all models-ViT [15], ViT AugReg [54] (AR), and DeiT [57], even if the confidence of the model is above 90%.

The finetuning procedure employs three loss terms. The first encourages the relevance map to assign lower values to the background of the image. The second term aims to remedy the sparse relevance issue by making larger parts of the foreground part of the relevancy map. The third term is a regularizer that ensures that the classification accuracy of the original model remains unimpaired.

Unsurprisingly, applying this method leads to a (modest) drop in accuracy on the original training dataset and on datasets with very similar distributions. In an extensive battery of experiments we show that (i) the classification accuracy on datasets from shifted domains increases considerably. This includes real-world unbiased and adversarial datasets, as well as synthetic ones that were created specifically to measure the robustness of the classification model, (ii) the resulting relevance maps demonstrate a significant improvement in focusing on the foreground of the image, i.e. the object, rather than on its background.

2 Related Work

Image classification datasets are becoming increasingly challenging, while at the same time models are growing more complex [13, 33, 55, 24]. With the rapid advancements in object recognition, the measuring stick is being narrowed down to a single number, accuracy [56]. By relying solely on accuracy, classifiers introduce biases, since they utilize shortcuts to select the right class [34, 20]. The shortcomings of models that rely on shortcuts have been demonstrated for domains other than vision, such as natural language processing [16] and multi-modal learning [19]. Therefore, it is becoming increasingly important to evaluate classifiers based on additional measures, such as robustness to distribution shifts, to ensure that the accuracy of the model does not conceal a problematic decision-making process.

Detecting shortcuts One way to assess the salient behavior of a model is to study Sufficient Input Subsets (SIS), i.e., the minimal number of pixels necessary for a confident prediction [8, 7]. Finding an SIS for a class can imply that the classifier has overinterpreted its input, since it can make a confident accurate decision using a small, sparse subset of pixels, which does not seem meaningful to humans. We study the SIS with gradients approach for ViT models, and find that it can be misleading. Specifically, SIS can be regarded as an adversarial method that can lead to high-confidence classification of *any label* from a sparse set of pixels, see Appendix A.

Explainability methods may be used to determine the reasons for the decisions made by classifiers. As an example, we may find that models tend to overlook objects with relevance maps (see Fig. 1). Gradients are a dominant and useful signal for model interpretation [49, 12, 37]. By adding input signals to the gradients, relevance maps were refined [22, 48, 52, 53]. Alternatives to gradient propagation for explanation include attribution propagation - a theory-driven method based on axioms, and permutation based on Shapley values[3, 38, 39, 48, 36]. For transformer architectures, the combination of gradients and attention values has been shown to produce a viable interpretation of the model's prediction [9, 10].

Reducing reliance on shortcuts Our method optimizes the relevancy maps of the model as a regularization term. Other works that investigated the use of relevancy to alleviate overfitting include Ross et al. [44], who introduced a regularization term for the input gradient, which reduces reliance on irrelevant cues, e.g., background pixels. Additional work in this vein has been conducted on medical data, studying how doctors classified a disease [50, 59]. Singh et al. [51] regularize feature representations of a category from its co-occurring context. Zhu et al. [66] enrich classifier representations by mimicking detection models. Importantly, unlike all the above methods, our method incorporates the foreground features and the classifier confidence, rather than considering only the background features. Furthermore, we apply our method not during training, but as a short finetuning process that is feasible for large models.

Reducing bias A related yet slightly different task is that of debiasing models (e.g., eliminating gender biases in language models). Several methods have been proposed to reduce biases. One approach is training with adversarial strategies [31, 62]. Other approaches suggest generating counterfactual samples to mitigate reliance on biased features, for example, masking gender features and preventing a captioning model from generating gender-related words [25, 67, 11].

3 Method

Our approach aims to direct vision models such that their decision will be based on the features of the object rather than on other supportive background features. To achieve this, we employ additional supervision to distinguish between the foreground and background features. The method finetunes the model in a way that encourages the class relevance map, obtained through a relevance computation method, to roughly resemble the segmentation map. This way, the decision-making process is focused on the foreground. The relevance map employed by our method is calculated using a recent advancement in explainability for transformer-based architectures [9]. A brief introduction of the explainability method used is provided in Appendix B.

The method employs a small set of labeled segmentation maps for distinguishing between the foreground and background of the input image. Our first loss term discourages the model from considering mostly the background:

$$\mathcal{L}_{\text{bg}} = \text{MSE}\left(\mathbf{R}(i) \odot \bar{\mathbf{S}}(i), 0\right), \tag{1}$$

where i is the input image, $\mathbf{R}(i)$ is the relevance map produced for i, $\bar{\mathbf{S}}(i)$ is the inverse of the segmentation map for i, and $\odot$ is the Hadamard product. Put differently, $\mathcal{L}_{\text{bg}}$ extracts the relevance values assigned to the background using the provided segmentation, and encourages those values to be close to 0, which is the minimal possible relevance value.

Our second loss term encourages the model to consider as much information as possible from the foreground of the image:

$$\mathcal{L}_{\text{fg}} = \text{MSE}\left(\mathbf{R}(i) \odot \mathbf{S}(i), 1\right), \tag{2}$$

where $\mathbf{S}(i)$ is the foreground mask. This loss encourages the relevance of pixels inside the segmentation to be higher (1 is the maximal achievable relevance value). The overall explainability loss is constructed as follows:

$$\mathcal{L}_{\text{relevance}} = \lambda_{\text{bg}} \cdot \mathcal{L}_{\text{bg}} + \lambda_{\text{fg}} \cdot \mathcal{L}_{\text{fg}}, \tag{3}$$

where $\lambda_{\text{bg}}, \lambda_{\text{fg}}$ are hyperparameters. All our experiments apply the same choice of $\lambda_{\text{bg}} = 2, \lambda_{\text{fg}} = 0.3$. Note that the coefficient λ_{fg} is much smaller than λ_{bg}. The reason is two-fold: (i) we find that the issue of overinterpreting the background is more common than the issue of partial relevance of foreground pixels, and (ii) $\mathcal{L}_{\text{fg}}$ implies a uniform relevance of 1 for all foreground pixels, which may be detrimental, as we wish to allow the model to be able to focus on specific features of the object.

Finally, in the absence of an additional regularization loss, the finetuning results in explanations that resemble the ground-truth segmentation, while the accuracy plummets due to the absence of encouragement to maintain high accuracy. Therefore, one must apply an additional loss term to ensure that the output distribution of the model remains similar to the original model. We opt to use a confidence-boosting loss for this purpose, which is constructed as follows:

$$\mathcal{L}_{\text{classification}} = \text{CE}\left(\mathcal{M}(i), \arg\max(\mathcal{M}(i))\right), \tag{4}$$

where $\mathcal{M}$ notates the vision model, and $\arg\max(\mathcal{M}(i))$ is the class predicted by $\mathcal{M}$ for the input image i. $\mathcal{L}_{\text{classification}}$ calculates the cross-entropy loss between the output distribution of $\mathcal{M}$ and the one-hot distribution where the predicted class is assigned a probability of 1. In other words, this loss encourages the confidence of the predicted class to increase.

The overall loss for the finetuning process is, therefore:

$$\mathcal{L} = \lambda_{\text{relevance}} \cdot \mathcal{L}_{\text{relevance}} + \lambda_{\text{classification}} \cdot \mathcal{L}_{\text{classification}}, \tag{5}$$

where $\lambda_{\text{relevance}} = 0.8$, and $\lambda_{\text{classification}} = 0.2$ remain constant in all our experiments.

4 Experiments

The main hypothesis of this work is that improving the salient maps of ViTs trained on ImageNet will result in reduced overfitting, and better generalization to data from unseen distributions. We present a wide range of tests to confirm our hypothesis. Importantly, all the models we experiment on have been fine-tuned on ImageNet-1k, which is also the dataset we employ in our fine-tuning process.

First, we evaluate the improvement in robustness, i.e., the ability to maintain high accuracy under distribution shifts. The datasets with shifted distributions are only used for evaluation and contain both real-world datasets and synthetic ones. Second, we conduct segmentation tests following [10] to assess the effect of our method on the level of agreement between the relevancy maps and the foreground segmentation maps. Third, following [65], our method employs samples from a subset of the classes during training, so that we can check whether the training classes (set A of the labels) differ from the set of classes not used during training (set B). Ideally, the method would have a positive effect on test samples from both sets A and B. Furthermore, in Appendix G we evaluate how sensitive the method is to the number of samples per class, and to the number of classes in set A.

Robustness benchmarks Several alternatives have been proposed to ImageNet [45] to examine robustness to distribution shifts: (i) ImageNet-v2 [42], a new test set sampled from the same distribution, which reduces adaptive overfitting; (ii) ImageNet-A [27], a test set of natural adversarial samples; (iii) ImageNet-R [26] and ImageNet-Sketch [61], which contain renditions of objects (e.g., art, sculptures, sketches); (iv) ObjectNet [5], a real-world set with controls on object locations and view points, and (v) SI-Score [14], which is a synthetic dataset designed specifically for testing robustness to object location, rotation and size. We employ all the aforementioned datasets to evaluate accuracy when facing distribution shifts, and assess the model's resilience and ability to generalize.

Baseline Methods We focus on methods that resemble ours, i.e. methods that strive to correct overfitting by manipulating the saliency maps of the model. Our baselines include GradMask [50], and Right for the Right Reasons [44] (RRR). Both GradMask and RRR were originally applied during training; in this work, however, we focus on large vision models that require significant resources for training, which makes this approach computationally impossible for us. Therefore, we apply their accuracy and relevancy losses within a finetuning process, similar to ours, in order to test their success in a fair way, while adhering to computational limitations.

Both GradMask and RRR employ two loss functions. First, a classic cross-entropy loss with the ground-truth labels to ensure correct labeling, and second, a loss limiting the values of the gradients of irrelevant parts of the input. The latter resembles our background loss (Eq. 1), with gradients as the relevance map. We refer the reader to Appendix C for the full description of the applied losses.

We note that while using the gradient of the output w.r.t. the input is common practice for interpreting CNNs, these gradients are less stable for transformer-based models. For example, results presented in [35] demonstrate that for transformer-based models the classic Input$\times$Gradient method violates faithfulness. We found it difficult to grid-search hyperparameters to fit both accuracy and relevancy losses simultaneously for the baselines. Furthermore, we had to tune the hyperparameters for each model separately to obtain an improved relevance loss. Our method, on the other hand, uses the same hyperparameter choice (see Sec. 3) for all models, which makes it far more stable to use, thus allowing us to run experiments on large models as well (i.e. ViT-L, ViT AugReg-L). We refer the reader to Appendix D for the full description of hyperparameters used in ours experiments.

Models and Training To demonstrate the effectiveness of our method, we experiment on three types of ViT-based models: vanilla ViT [15], ViT AugReg (AR) [54], and DeiT [58] which presents techniques for efficiently training ViTs. For each model type, we experiment with different model

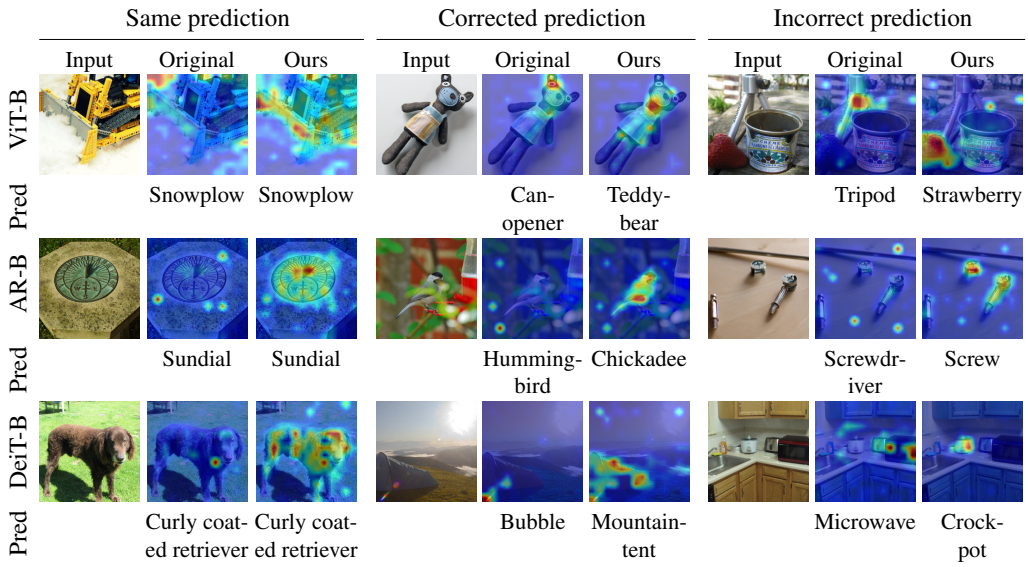


Figure 2: Examples from the ImageNet validation set of cases where our method does not change the prediction, corrects the prediction, and ruins the prediction. Even in cases where our method changes a correct prediction, there is often a rationale behind the modified prediction.

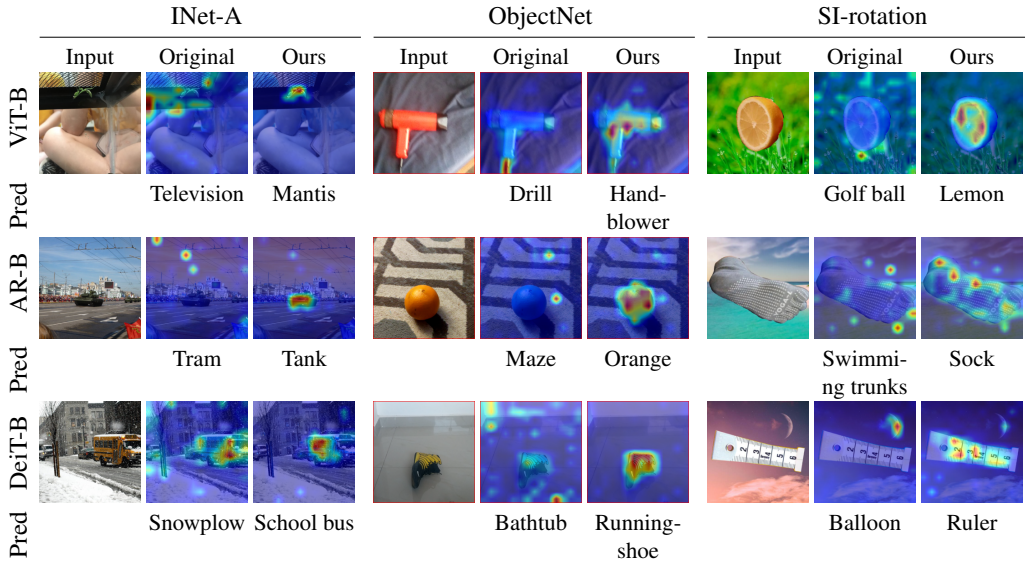


Figure 3: Examples of cases where our method corrects wrong predictions, alongside the original and modified (after finetuning) explainability maps. The “Pred" row specifies the predictions before and after our finetuning. The original classifiers focus on sparse or irrelevant data. The examples are presented for the base models of ViT [15], ViT AugReg [54] (AR), and DeiT [57].

sizes, in order to learn whether our method improves robustness across training techniques and model sizes. All models use 224 resolution images with a patch size of 16×16. We use the implementation and pre-trained weights from [64]. The small, base models are finetuned on a single RTX 2080 Ti GPU, and the large models on a single Tesla V100 GPU. All models are finetuned as described in Sec. 3 for 50 epochs, with a batch size of 8. We use 3 training images from 500 ImageNet classes for our finetuning (overall- 1500 samples), and another 414 images as a validation set. In most experiments, we simply select the first 500 classes from ImageNet-S to be used for training (set A). For results with multiple random seeds and multiple choices of the 500 classes to use for training, see Appendix E. The learning rate of each model is determined using a grid search between the values $5e-7$ and $5e-6$. As a rule of thumb, we select the highest learning rate for which the validation

Table 1: Robustness evaluation for ViT [15], ViT AugReg [54] (AR), and DeiT [58] with our method, and the baseline methods GradMask [50] and Right for the Right Reason (RRR) [44]. "Annotated segmentation" indicates whether we used annotated segmentation [18] or unsupervised localization [63]. "Original" stands for the model without finetuning. The bottom rows indicate the average change caused by our method across all architectures (on some models the baselines could not be run successfully; therefore, we do not compute their average change).

Model	Method	Annotated segmentation	INet val		INet-A		INet-R		Sketch		INet-v2		ObjNet	
			R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
ViT-B	Original	✗	81.5	96.0	16.0	37.0	33.8	48.5	35.4	57.4	71.1	89.9	35.1	56.4
	GradMask	✓	81.8	96.1	17.5	39.8	34.5	49.4	35.8	57.8	**71.4**	**90.5**	36.7	58.2
	RRR	✓	**81.9**	**96.2**	18.9	41.9	34.8	49.7	35.8	57.8	**71.4**	**90.5**	38.1	60.0
	Ours	✓	80.3	95.4	**24.1**	**48.0**	**36.3**	**51.4**	**36.2**	**58.5**	70.0	89.4	**42.2**	**65.1**
	Ours	✗	80.4	95.4	23.0	45.7	35.4	50.0	35.8	58.2	69.8	89.4	40.8	64.0
ViT-L	Original	✗	**82.9**	**96.4**	19.0	41.5	36.6	52.0	40.4	63.4	71.8	90.7	37.4	59.5
	Ours	✓	82.0	96.2	**25.2**	49.6	38.8	54.6	41.2	64.3	71.3	90.6	42.5	65.4
	Ours	✗	82.7	**96.4**	**25.2**	**50.0**	**39.8**	**55.1**	**41.8**	**64.8**	**72.1**	**91.2**	**43.2**	**65.8**
AR-S	Original	✗	81.4	**96.1**	13.0	33.9	31.2	47.1	32.8	54.2	69.9	90.1	34.3	55.8
	GradMask	✓	81.3	**96.1**	16.4	39.2	32.3	48.3	32.5	53.7	70.1	**90.3**	37.6	60.2
	RRR	✓	**81.5**	**96.1**	13.7	35.1	31.6	47.4	32.9	54.2	**70.3**	90.1	35.1	56.7
	Ours	✓	79.8	95.7	18.2	40.6	**33.9**	**50.2**	33.5	55.4	69.6	90.0	38.7	61.1
	Ours	✗	80.3	95.8	**19.1**	**42.2**	33.8	49.7	**33.8**	**55.5**	69.6	90.1	**39.3**	**61.7**
AR-B	Original	✗	84.4	97.2	23.9	49.2	41.0	57.8	43.1	65.7	73.8	92.3	41.4	63.7
	GradMask	✓	84.5	**97.3**	25.1	51.4	41.5	58.1	43.1	65.7	74.0	**92.6**	42.7	64.8
	RRR	✓	**84.6**	**97.3**	26.8	53.0	41.9	58.5	43.2	65.7	**74.3**	**92.6**	43.7	65.9
	Ours	✓	83.1	96.9	**31.3**	57.1	**44.7**	**61.5**	44.6	**67.4**	73.5	92.0	**47.1**	**70.0**
	Ours	✗	83.6	97.1	31.2	**57.2**	44.5	60.9	**44.7**	**67.4**	73.7	92.4	46.5	69.1
AR-L	Original	✗	**85.6**	**97.8**	34.7	61.0	48.8	64.9	51.8	73.6	75.8	93.4	46.5	68.3
	Ours	✓	85.1	97.5	42.1	67.5	**54.0**	**69.1**	**54.2**	**75.8**	75.8	93.4	51.6	73.2
	Ours	✗	85.4	97.6	**42.4**	**68.0**	53.8	69.0	54.1	**75.8**	**76.1**	**93.6**	**52.0**	**73.5**
DeiT-S	Original	✗	78.1	93.7	8.3	23.5	28.2	41.9	28.8	46.7	66.5	86.6	28.3	47.3
	GradMask	✓	77.0	93.6	7.9	24.7	26.6	40.5	26.0	43.5	64.5	85.6	28.2	48.6
	RRR	✓	78.1	94.1	9.0	26.9	26.9	40.6	26.9	44.4	66.0	86.7	29.3	49.9
	Ours	✓	**78.6**	**94.5**	10.1	29.0	29.3	43.6	29.1	47.8	**67.3**	87.3	**31.6**	**53.0**
	Ours	✗	**78.6**	94.4	**11.0**	**30.3**	**29.9**	**44.4**	**29.4**	**48.0**	67.1	**87.4**	**31.6**	52.9
DeiT-B	Original	✗	80.8	94.2	12.9	31.0	30.9	44.2	31.2	48.6	**69.7**	86.8	31.4	48.5
	GradMask	✓	**81.1**	**95.3**	15.1	36.9	31.0	45.5	31.2	49.1	**69.7**	**88.7**	33.5	53.1
	RRR	✓	81.0	95.2	14.8	37.0	30.7	45.1	30.9	48.8	69.5	88.6	33.6	53.3
	Ours	✓	80.5	94.9	17.2	40.0	32.4	47.0	30.9	49.2	69.1	88.3	35.9	56.2
	Ours	✗	80.5	95.0	**18.3**	**40.9**	**32.8**	**47.5**	**31.5**	**49.9**	69.3	88.5	**36.3**	**56.6**
Avg. change	Ours	✓	-0.8	0.0	+5.8	+7.8	+2.7	+3.0	+0.9	+1.3	-0.3	+0.2	+5.0	+6.4
	Ours	✗	-0.5	0.0	+6.1	+8.2	+2.8	+2.9	+1.1	+1.4	-0.1	+0.4	+5.0	+6.3

accuracy does not decrease by more than $2-3\%$ and $\mathcal{L}_{\text{classification}}$ does not increase. We find that our method is not sensitive to small shifts in the learning rate, thus this rule applies well to all models.

As mentioned in Sec. 3, we use segmentation maps to distinguish between the foreground and the background of an input image. Our experiments employ two options of obtaining segmentation maps for ImageNet training images. In the first option, we use human-annotated segmentation maps from the ImageNet-S dataset [18]. The ImageNet-S dataset contains 10 training samples with their segmentation maps for 919 of the ImageNet classes. We employ a considerably smaller subset, as detailed above. The second option does not employ extra supervision in the form of manually labeled images; instead, it employs the Tokencut object localization method presented in [63] to produce foreground segmentation maps.

Results Tab. 1 presents the results of our method and the baseline methods applied to different ViT-based models. As can be seen, for real-world datasets, both adversarial datasets (INet-A) and

Table 2: Robustness evaluation on the synthetic SI-Score dataset [14], which tests changes in object position, rotation, and size using our method and the baseline methods GradMask [50], Right for the Right Reasons (RRR) [44]. The models tested are ViT [15], ViT AugReg [54] (AR), and DeiT [58]. "Annotated segmentation" indicates if we used annotated segmentation [18] or unsupervised localization [63]. "Original" stands for the model without finetuning.

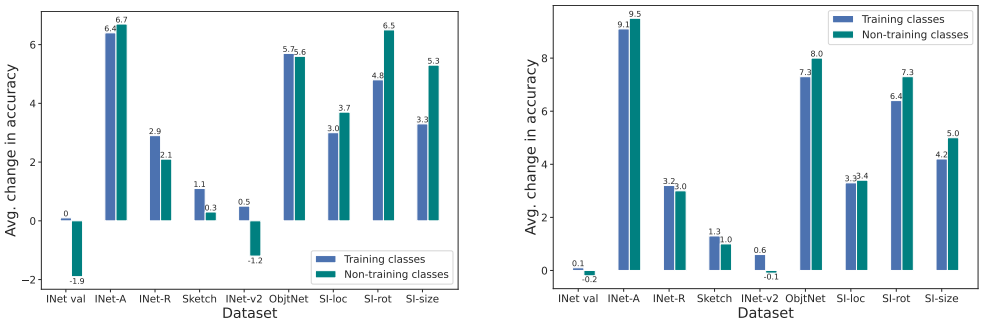
Model	Method	Annotated segmentation	SI-location		SI-rotation		SI-size	
			R@1	R@5	R@1	R@5	R@1	R@5
ViT-B	Original	✗	33.3	52.2	39.1	58.3	55.6	76.2
	GradMask	✓	34.6 (+1.3)	53.9 (+1.7)	40.7 (+1.6)	60.3 (+2.0)	57.0 (+1.4)	77.5 (+1.3)
	RRR	✓	35.6 (+2.3)	55.0 (+2.8)	41.9 (+2.8)	61.8 (+3.5)	58.0 (+2.4)	78.4 (+2.2)
	Ours	✓	**38.6** (+5.3)	**57.8** (+5.6)	**46.2** (+7.1)	**67.0** (+8.7)	**61.0** (+5.4)	**81.4** (+5.2)
	Ours	✗	38.4 (+5.1)	57.0 (+4.8)	44.8 (+5.7)	65.2 (+6.9)	60.2 (+4.6)	80.6 (+4.4)
ViT-L	Original	✗	31.6	50.3	40.7	60.1	54.8	75.6
	Ours	✓	36.3 (+4.7)	56.2 (+5.9)	**45.3** (+4.6)	66.2 (+6.1)	58.6 (+3.8)	80.3 (+4.7)
	Ours	✗	**36.7** (+5.1)	**56.3** (+6.0)	**45.3** (+4.6)	**66.6** (+6.5)	**59.1** (+4.3)	**80.5** (+4.9)
AR-S	Original	✗	32.4	51.7	40.6	59.6	55.4	75.7
	GradMask	✓	34.3 (+1.9)	53.9 (+2.2)	43.3 (+2.7)	63.0 (+3.4)	58.0 (+2.6)	78.3 (+2.6)
	RRR	✓	32.9 (+0.5)	52.3 (+0.6)	41.4 (+0.8)	60.6 (+1.0)	56.0 (+0.6)	76.3 (+0.6)
	Ours	✓	**36.8** (+4.4)	**56.6** (+4.9)	**47.6** (+7.0)	**67.8** (+8.2)	**61.3** (+5.9)	**81.2** (+5.5)
	Ours	✗	36.3 (+3.9)	55.6 (+3.9)	46.6 (+6.0)	66.7 (+7.1)	60.7 (+5.3)	80.4 (+4.7)
AR-B	Original	✗	40.5	60.8	48.1	68.3	60.6	80.4
	GradMask	✓	41.5 (+1.0)	61.8 (+1.0)	49.3 (+1.2)	69.5 (+1.2)	61.4 (+0.8)	81.3 (+0.9)
	RRR	✓	42.4 (+1.9)	62.7 (+1.9)	50.4 (+2.3)	70.7 (+2.4)	62.1 (+1.5)	82.0 (+1.6)
	Ours	✓	43.2 (+2.7)	62.8 (+2.0)	54.0 (+5.9)	74.6 (+6.3)	64.1 (+3.5)	83.9 (+3.5)
	Ours	✗	**44.3** (+3.8)	**64.0** (+3.2)	**54.6** (+6.5)	**74.7** (+6.4)	**64.5** (+3.9)	**84.6** (+4.2)
AR-L	Original	✗	43.8	64.2	52.4	72.5	62.3	82.2
	Ours	✓	**48.3** (+4.5)	**68.5** (+4.3)	57.0 (+4.6)	77.2 (+4.7)	66.4 (+4.1)	**86.0** (+3.8)
	Ours	✗	47.4 (+3.6)	67.4 (+3.2)	**58.0** (+5.6)	**78.1** (+5.6)	**66.5** (+4.2)	85.6 (+3.4)
DeiT-S	Original	✗	30.7	50.4	36.7	54.3	51.6	72.0
	GradMask	✓	32.0 (+1.3)	50.7 (+0.3)	38.9 (+2.2)	56.7 (+2.4)	54.1 (+2.5)	74.0 (+2.0)
	RRR	✓	32.0 (+1.3)	51.0 (+0.6)	38.5 (+1.8)	56.3 (+2.0)	53.9 (+2.3)	73.8 (+1.8)
	Ours	✓	32.3 (+1.6)	**51.5** (+1.1)	40.6 (+3.9)	59.4 (+5.1)	55.8 (+4.2)	**76.3** (+4.3)
	Ours	✗	**32.5** (+1.8)	51.4 (+1.0)	**41.0** (+4.3)	**59.6** (+5.3)	**56.0** (+4.4)	76.1 (+4.1)
DeiT-B	Original	✗	34.5	54.6	39.3	56.3	54.6	73.4
	GradMask	✓	34.1 (-0.4)	54.9 (+0.3)	39.1 (-0.2)	58.3 (+2.0)	55.2 (+0.6)	75.8 (+2.4)
	RRR	✓	34.4 (-0.1)	55.2 (+0.6)	40.4 (+1.1)	58.5 (+2.2)	55.3 (+0.7)	75.8 (+2.4)
	Ours	✓	36.6 (+2.1)	57.0 (+2.4)	42.9 (+3.6)	61.5 (+5.2)	58.0 (+3.4)	78.2 (+4.8)
	Ours	✗	**37.8** (+3.3)	**58.1** (+3.5)	**44.2** (+4.9)	**62.7** (+6.4)	**59.3** (+4.7)	**79.0** (+5.6)

datasets with random and controlled background, rotations, and view points (ObjectNet), our method significantly and consistently improves performance (average of $5.8\%, 5.0\%$ top-1 improvement on INet-A, ObjectNet, respectively). For datasets that contain art, sculptures, sketches etc. (INet-R, INet-Sketch) the increase in accuracy is less steep ($2.7\%, 0.9\%$ averaged top-1 improvement, respectively). This can be intuitively attributed to the fact that art and sketches often feature the object without a background, or with a uniform background. Additionally, as can be seen, although the baseline methods preserve accuracy on the datasets from the original ImageNet distribution (INet val set and INet-v2), they fall behind our method on real-world out-of-distribution datasets (INet-A and ObjectNet), indicating that the baselines are less successful in alleviating overfitting.

Following our method, there is a slight decrease in the performance of the models on data from the ImageNet distribution (INet val set and INet-v2), which can be attributed to the fact that we reduce some of the overfitting on the ImageNet data. Fig. 2 presents examples of all three possible prediction cases on the ImageNet validation set: cases where our method preserves the original classification, corrects the original prediction, and changes a correct prediction. As can be seen, while performance sometimes decreases with respect to the assigned label, in most cases the rationale of the class proposed by the modified model is clear, often indicating a more natural labeling option. In addition, in the cases where the prediction remains the same or is corrected, our method produces improved relevance maps. An extended version of Fig. 2 appears in Appendix F.

Table 3: Evaluation of segmentation performance from relevance maps on the ImageNet-segmentation dataset [23] for ViT [15], ViT AugReg [54] (AR), and DeiT [58] before and after finetuning with our method. Metrics and dataset are taken from [10].

Model	ViT-B		ViT-L		AR-S		AR-B		AR-L		DeiT-S		DeiT-B	
	Orig	Ours	Orig	Ours	Orig	Ours	Orig	Ours	Orig	Ours	Orig	Ours	Orig	Ours
Pixel acc.	76.3	**82.1**	73.4	**82.5**	76.7	**83.3**	76.6	**81.2**	65.2	**78.9**	78.7	**80.8**	79.0	**81.3**
mIoU	58.3	**65.8**	54.4	**66.4**	57.7	**67.7**	57.1	**64.6**	43.6	**61.0**	60.7	**64.0**	61.6	**64.7**
mAP	85.3	**87.5**	82.7	**86.9**	84.2	**87.7**	84.4	**86.8**	78.6	**85.4**	85.0	**86.4**	85.7	**86.8**



(a) (b)

Figure 4: Evaluation of the average change produced by our method on the training classes and on the classes that were not in the training set. Changes are averaged across the base models of ViT [15], ViT AugReg [54], and DeiT [57]. (a) top-1 average change, (b) top-5 average change.

Additionally, as shown in Tab. 2 for the SI-Score dataset, which is a synthetic dataset designed for testing resilience for shifts in object locations, sizes, and rotations, there is a very steep improvement in performance across all models, while, once again, the baselines fall behind on all models and sizes.

Evidently, in both Tab. 1,2 our method works just as well and often better when using the unsupervised segmentation maps. This means that our method may be applied without requiring any manual supervision, except for the image label.

Fig. 3 presents example cases in which our method is able to correct the prediction of the original model on images from various robustness datasets. As can be seen, the original models tend to overinterpret the background, and therefore produce false classifications based on it. For example, a lemon is classified as a golf ball due to the grass in the background (third example in the first row), a tank is classified as a tram due to the tram cables at the top of the image (first example in the second row), and so on. An extended version of the figure with additional models appears in Appendix F.

Segmentation Tests Since our motivation is to encourage the relevance to focus less on the background and more on as much of the foreground as possible, we test the resemblance of the resulting relevance maps to the segmentation maps following [10]. As can be seen in Tab. 3, our method significantly and consistently improves segmentation metrics on all models, indicating that our finetuning indeed achieves its goal.

Comparing training classes to the other classes To ensure that the effects of our finetuning generalize to classes that were not included in the training set, we test the increase in robustness resulting from our method separately on the training classes and non-training classes. Fig. 4 presents the average improvement across the base models of ViT [15], ViT AugReg [54](AR), and DeiT [57]. As can be seen, both subsets of classes demonstrate a very similar increase in accuracy for the robustness datasets, with the classes that belong to the training set demonstrating better performance on the datasets from the original ImageNet distribution (INet val, INet-v2), as can be expected due to the fact that they were represented in the training set. The full results of this experiment are presented in Appendix H.

Ablation Study We conduct an ablation study to test the effect of each of our loss terms on the result of the finetuning process, by studying the impact of removing each loss term $\mathcal{L}_{\text{classification}}, \mathcal{L}_{\text{bg}}, \mathcal{L}_{\text{fg}}$.

Our ablation study is conducted on the base models of ViT and DeiT, as they demonstrate different advantages for each loss term. Additionally, we perform an ablation to test the choice of confidence-boosting as the classification loss (Eq. 4), by replacing it with the classic cross-entropy loss with the ground-truth label. For brevity, Tab. 4 presents the top-1 accuracy results for each version of our method; the complementary table for top-5 accuracy can be found in Appendix I. As can be seen, validation accuracy is lost mostly due to our background loss (Eq. 1), as when we remove it, the accuracy remains intact, and yet, as we hypothesized, it is more effective than the foreground loss (Eq. 2) in increasing the robustness, since when removing it, the accuracy on the out-of-distribution datasets drops significantly. Additionally, the DeiT ablation demonstrates that in some cases, the classification loss does not contribute to the increase in robustness (other than for INet-A), nor is it necessary for preserving the original ImageNet validation accuracy. However, for ViT, the classification loss is crucial for avoiding a significant loss of accuracy.

Next, we explore the variants where only one of the relevance loss terms $\mathcal{L}_{\text{bg}}, \mathcal{L}_{\text{fg}}$ is applied without the classification loss $\mathcal{L}_{\text{classification}}$ (labeled "only $\mathcal{L}_{\text{bg}}$", "only $\mathcal{L}_{\text{fg}}$" in Tab. 4). We note that by removing the classification loss, the model is at risk of mode collapse since $\mathcal{L}_{\text{bg}}$ encourages the relevance on the background to be 0. The mode collapse, in this case, would be to zero-out the relevancy map for the entire image. In an analog manner, $\mathcal{L}_{\text{fg}}$ encourages a high relevance on the foreground and can cause a mode collapse where all the image receives uniform relevancy of 1. When applied together (the "w/o $\mathcal{L}_{\text{classification}}$ (Eq. 4)" ablation in Tab. 4) the two losses balance each other. However, when only employing one loss without the other and not adding a regularization term, the fine-tuning would encourage a mode collapse leading to lower accuracy.

Additionally, our ablation study demonstrates the benefit of using confidence boosting over the ground-truth for the classification loss (labeled "w/ ground-truth" in Tab. 4). While the ground-truth variant preserves the original accuracy better (more so for ViT than for DeiT), using confidence boosting often improves robustness more significantly over using the ground-truth labels (see for example INet-A for ViT).

Finally, we explore different options to replace GAE [9] as the explainability algorithm to produce the relevance maps. We consider two variants that employ pure attention weights. The first, labeled "Attention instead of [9]" in Tab. 4, utilizes the pure attention weights of the last attention layer as an explanation. The second, labeled "Rollout instead of [9]" in Tab. 4, employs attention Rollout [1] which combines the attention maps in all layers linearly using matrix multiplication. As can be seen, our method outperforms both these ablations, with the exception of ImageNet and ImageNet-v2. This can be explained by the experiments in [9, 10] which demonstrate that pure attention weights do not constitute a faithful explanation.

5 Discussion and limitations

A surge of works have explored the benefits of using large-scale datasets of unlabeled samples from the internet, and many techniques have been proposed to train vision models in an unsupervised or self-supervised manner [6, 21]. As a result, it is increasingly important to develop methods for boosting model robustness without requiring any labels. We note that our method is compatible with such frameworks. As can be seen from Tab. 1, 2, our method performs well, even when applied with foreground masks obtained in an unsupervised manner, using Tokencut. Additionally, all of our losses can be applied without knowledge of the ground-truth label, since our classification loss $\mathcal{L}_{\text{classification}}$ uses the predicted label, and the relevance maps could be propagated w.r.t. the predicted class.

Recently, it has been discovered that the effect of augmentation-based regularization is extremely uneven across classes [4]. While such a regularization improves performance on average, some classes benefit significantly from it, while other classes suffer from a large drop in accuracy. This variation is also apparent in the effect of our method. Above we have established that the method helps both classes which are included in the training set of the finetuning and classes which are not. However, in both sets there is considerable variability in the effect of individual classes. This makes sense, since some classes present more well-localized objects and some classes require more reliance on the context of the object. See Appendix J for these results.

Table 4: Ablation study for our method on ViT [15] and DeiT [57] base models. The table presents top-1 accuracy. We check the impact of each term of our loss on robustness, as well as the choice of confidence boosting versus cross-entropy with the ground-truth label (labeled as w/ ground-truth).

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
ViT-B	Original	81.5	16.0	33.8	35.4	71.1	35.1	33.3	39.1	55.6
	Ours	80.3	**24.1**	**36.3**	36.2	70.0	**42.2**	38.6	**46.2**	**61.0**
	w/o $\mathcal{L}_{\text{classification}}$ (Eq. 4)	77.8	17.9	34.2	34.8	67.4	37.6	37.2	43.0	58.4
	w/o $\mathcal{L}_{\text{bg}}$ (Eq. 1)	81.5	19.1	35.9	**36.4**	**71.4**	39.7	34.9	42.3	58.4
	w/o $\mathcal{L}_{\text{fg}}$ (Eq. 2)	80.2	**24.1**	34.3	35.2	69.6	41.8	**39.2**	45.6	60.7
	w/ ground-truth	**81.7**	21.5	35.5	35.8	71.2	40.3	37.8	44.5	60.1
	only $\mathcal{L}_{\text{bg}}$	66.0	18.0	23.2	25.8	55.1	32.1	33.3	35.0	50.5
	only $\mathcal{L}_{\text{bg}}$	77.1	8.9	31.1	33.6	65.4	31.4	25.8	33.1	50.3
	Attention instead of [9]	80.9	21.7	35.1	35.3	70.7	40.0	37.4	42.7	58.9
	Rollout instead of [9]	79.3	23.4	35.4	34.4	68.6	40.9	35.6	46.1	60.1
DeiT-B	Original	80.8	12.9	30.9	31.2	69.7	31.4	34.5	39.3	54.6
	Ours	80.5	17.2	32.4	30.9	69.1	35.9	36.6	42.9	58.0
	w/o $\mathcal{L}_{\text{classification}}$ (Eq. 4)	**81.0**	15.5	**33.3**	**32.1**	**69.8**	**36.1**	**39.0**	**44.0**	**58.3**
	w/o $\mathcal{L}_{\text{bg}}$ (Eq. 1)	**81.0**	13.2	30.5	30.2	69.6	33.0	32.7	39.7	54.5
	w/o $\mathcal{L}_{\text{fg}}$ (Eq. 2)	80.3	**17.9**	32.6	31.0	69.0	35.8	37.3	43.0	58.2
	w/ ground-truth	80.7	17.2	31.9	31.3	69.2	35.6	36.7	42.8	57.6
	only $\mathcal{L}_{\text{bg}}$	71.7	13.1	25.3	24.7	59.7	29.2	38.4	37.4	53.4
	only $\mathcal{L}_{\text{fg}}$	79.8	6.9	27.5	28.2	67.7	28.3	27.6	35.3	49.9
	Attention instead of [9]	**81.0**	15.4	31.8	31.3	**69.8**	34.0	35.5	41.4	56.4
	Rollout instead of [9]	80.8	14.4	31.4	30.9	69.6	32.6	34.0	40.5	55.6

6 Conclusions

Can segmentation information help image categorization? Intuitively, the answer has to be positive. However, despite some effort to manually segment images of classification datasets, the obtained improvement, if any, is soon overtaken by better image-level methods.

Here, we propose a generic way to improve classification accuracy that can be applied to virtually any image classifier. Applied to transformers, we present evidence to show that while the accuracy on the original dataset does not improve, there is an increase in accuracy for out-of-distribution test sets. The method optimizes the relevancy maps directly based on intuitive desiderata. It opens a new way for improving accuracy using explainability methods, which are currently seldom used for improving downstream tasks other than seeding weakly supervised segmentation methods.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *stat*, 1050:27, 2020.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[4] Randall Balestriero, Leon Bottou, and Yann LeCun. The effects of regularization and data augmentation are class dependent. *arXiv preprint arXiv:2204.03632*, 2022.

[5] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[7] Brandon Carter, Siddhartha Jain, Jonas W Mueller, and David Gifford. Overinterpretation reveals image classification model pathologies. In *Advances in Neural Information Processing Systems*, 2021.

[8] Brandon Carter, Jonas Mueller, Siddhartha Jain, and David Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576. PMLR, 2019.

[9] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, October 2021.

[10] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, 2020.

[12] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6970–6979, 2017.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[14] Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan I. Moldovan, Sylvan Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16453–16463, 2021.

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[16] Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*, 2018.

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[18] Shanghua Gao, Zhong-Yu Li, Ming-Hsuan Yang, Ming-Ming Cheng, Junwei Han, and Philip Torr. Large-scale unsupervised semantic segmentation. *arXiv preprint arXiv:2106.03149*, 2021.

[19] Itai Gat, Idan Schwartz, and Alex Schwing. Perceptual score: What data modalities does your model perceive? *Advances in Neural Information Processing Systems*, 34, 2021.

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

[21] Priya Goyal, Quentin Duval, Isaac Seessel, Mathilde Caron, Ishan Misra, Levent Sagun, Armand Joulin, and Piotr Bojanowski. Vision models are more robust and fair when pretrained on uncurated images without supervision. *ArXiv*, abs/2202.08360, 2022.

[22] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

[23] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110:328–348, 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, December 2015.

[25] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *ECCV*, 2018.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8340–8349, October 2021.

[27] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, June 2021.

[28] Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge J. Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12879–12888, 2021.

[29] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

[30] Daniel S. Kermany, Michael H. Goldbaum, Wenjia Cai, Carolina Carvalho Soares Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Yin Lin Ting, Jie Zhu, Christina M. Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet Anh Nguyen Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali R. Tafreshi, M. Anthony Lewis, Huimin Xia, and Kang Zhang. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172:1122–1131.e9, 2018.

[31] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, 2018.

[32] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[34] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[35] Y. Liu, Haoliang Li, Yangyang Guo, Chen Kong, Jing Li, and Shiqi Wang. Rethinking attention-model explainability through faithfulness violation test. *ArXiv*, abs/2201.12114, 2022.

[36] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[37] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

[38] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[39] Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2501–2508, 2020.

[40] A Nguyen, J Yosinski, and J Clune. Deep neural networks are easily fooled: high confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*, 2014.

[41] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: reducing the bias towards global statistics in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9482–9491, 2021.

[42] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[43] Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

[44] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. *ArXiv*, abs/1703.03717, 2017.

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[46] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[47] Rakshith Shetty, Bernt Schiele, and Mario Fritz. Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8226, 2019.

[48] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[49] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer, 2014.

[50] Becks Simpson, Francis Dutil, Yoshua Bengio, and Joseph Paul Cohen. Gradmask: Reduce overfitting by regularizing saliency. *ArXiv*, abs/1904.07478, 2019.

[51] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: learning to overcome contextual bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11070–11078, 2020.

[52] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[53] Suraj Srinivas and François Fleuret. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, pages 4126–4135, 2019.

[54] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *ArXiv*, abs/2106.10270, 2021.

[55] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[56] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011.

[57] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[59] Joseph D Viviano, Becks Simpson, Francis Dutil, Yoshua Bengio, and Joseph Paul Cohen. Saliency is a possible red herring when diagnosing poor generalization. *arXiv preprint arXiv:1910.00199*, 2019.

[60] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, 2019.

[61] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[62] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *ICCV*, 2019.

[63] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, June 2022.

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

[65] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

[66] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[67] Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *ACL*, 2019.

Submission checklist; appendices follow in the next pages.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Will be included in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix D, and in the experiments section of the paper.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Reported in Appendix E in the supplementary material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the Models and Training section
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] All used data and code is licensed for public use, given the appropriate credit (MIT license). The ObjectNet dataset [5] is the only exception to this rule, while it is free to use for research and commercial use, it may never be used in model training (we only use it for evaluation), and individual images from it should be presented with a 1 pixel red border (see Fig. 3, 7).
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] The data assets used are curated in a way that it does not contain sensitive or personal information.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The data assets used are curated in a way that it does not contain sensitive, personal, or offensive information.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Reinterpreting overinterpretation

One way to assess the salient behavior of a model is to study Sufficient Input Subsets (SIS), i.e., the minimal number of pixels necessary for a confident prediction [8]. A gradient signal can be utilized to find the sufficient pixels [7]. Finding an SIS for a class can imply that the classifier has overinterpreted its input since it can make a confident accurate decision using a small, sparse subset of pixels, which does not appear meaningful to humans.

We study the SIS with gradients approach for ViT models, and find that it can be misleading. Specifically, SIS can be regarded as an adversarial method that can lead to high-confidence classification of *any label* from a sparse set of pixels.

Fig. 5 demonstrates the resulting SIS pixels for 4 randomly selected images from the ImageNet validation set, with 5 randomly selected ImageNet labels- Gibbon, Black-widow, Common Iguana, Shovel, and Australian Terrier. As can be seen, we were able to find a subset of pixels for each random image and random label that made the classifier predict the random label with a confidence higher than 90%.

This can indicate that there is no guarantee that the classifier used the unnatural cues to decide for a given class based on the existence of a corresponding SIS, i.e. the subset of pixels considered an SIS is not necessarily indicative of the pixels used for the original decision, and resembles an adversarial attack in the sense that it can make the model predict unexpected outputs from a given input. Thus, we opt to use datasets designed specifically for testing robustness [20, 42, 27, 61, 5].

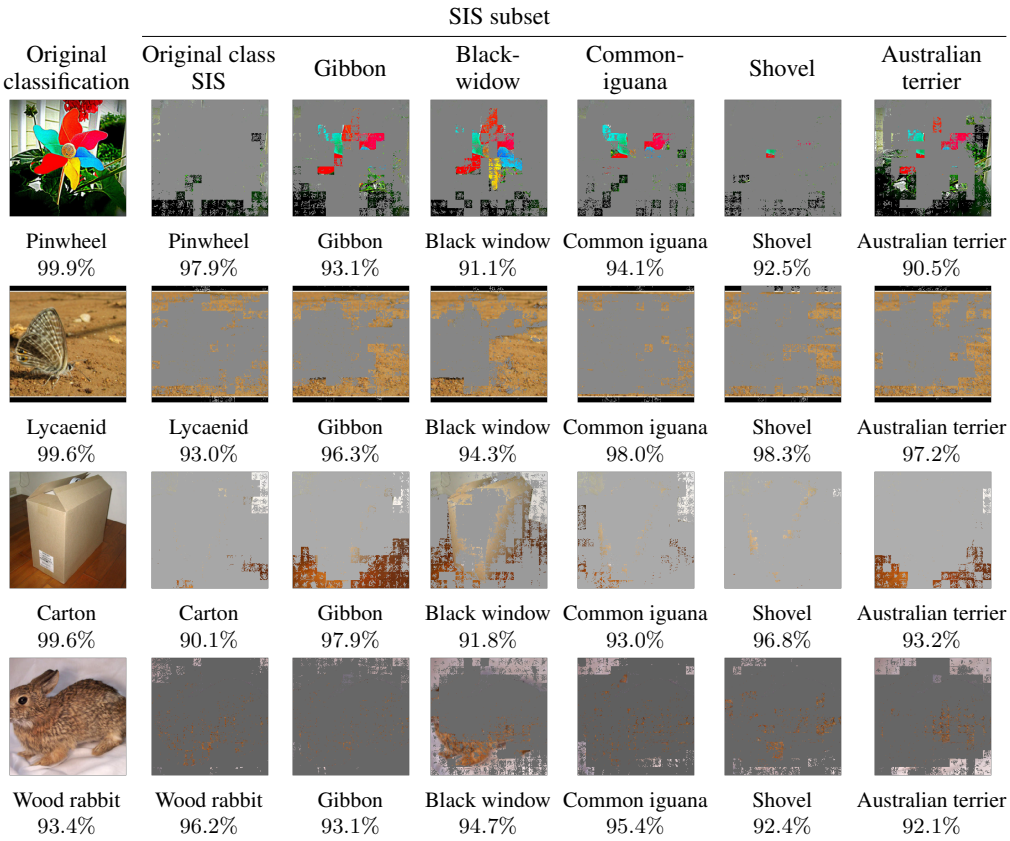


Figure 5: Examples of SIS subsets for 4 random images from the ImageNet validation set with 5 random ImageNet classes. As can be seen, an SIS subset can be found for all images with all classes with confidence higher than 90%, demonstrating that SIS subsets do not necessarily explain the prediction of the model.

B Background: explainability for Vision Transformers

Our method employs the Generic Attention Explainability (GAE) method [9] to create a relevance map for the input image tokens. In essence, GAE produces a relevance map $\bar{\mathbf{A}}$ for each self-attention layer. The final relevance map is the result of aggregating all the self-attention layers' relevance maps into a single map $\mathbf{R}(i)$ for the entire network in a forward pass using matrix multiplication.

We notate by h the number of different attention heads, and by s, d_h the number of tokens in the input sequence and the embedding dimension per head, respectively. Recall the self-attention mechanism employed by ViT:

$$\mathbf{A} = softmax\left(\frac{\mathbf{Q} \cdot \mathbf{K}^T}{\sqrt{d_h}}\right) \tag{6}$$

$$\mathbf{O} = \mathbf{A} \cdot \mathbf{V} \tag{7}$$

where $\mathbf{O} \in \mathbb{R}^{h \times s \times d_h}$ is the output of the attention module, $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{h \times s \times d_h}$ are the query, key, and value for the attention mechanism, namely, different projections of the input to the self-attention module. $\mathbf{A} \in \mathbb{R}^{h \times s \times s}$ is the attention matrix, where row i represents the attention coefficients of each token in the input with respect to the token i. The $softmax$ in Eq. 6 is applied, such that the sum of each row in each attention head of $\mathbf{A}$ is one.

Recall that after processing all the attention blocks, the ViT architecture regards only the classification token ([CLS] token) to produce the output distribution. Therefore, intuitively, the attention weights in the matrix $\mathbf{A}$ that correspond to the [CLS] token reflect the contribution of each token to the [CLS] token. Since the [CLS] token is the only one used for prediction, those contributions are often considered to be relevance values.

The authors in [9] follow this intuition to produce the relevance maps while handling two questions: (i) since, as demonstrated in [60], not all attention heads have an equal contribution to the prediction, how can the heads be averaged in a way that accounts for each head's relevance? (ii) given that a Transformer includes multiple attention layers, how can the different attention layers be aggregated to a single attention matrix?

GAE tracks the updates by the attention of each layer and updates the relevance maps accordingly. In order to account for the varying roles of each head, gradients are used as weights for each attention head, similar to GradCAM [46] where gradients are used as weights for the features maps of the last CNN layer. To aggregate across attention layers, the method uses matrix multiplication and the identity matrix to account for residual connections as in attention Rollout [1].

Importantly, we chose to use GAE over pure-attention or variants of it such as attention Rollout since several papers on Transformer explainability [9, 10, 35] have found pure attention weights to be unfaithful, i.e. pure attention does not loyally reflect the reasoning behind the decision-making process. The aforementioned works have also found GAE to be the most faithful method to interpret predictions by Transformers. Additionally, due to the use of a combination of pure attention weights and attention gradients, GAE is easily derivable, therefore it is well-suited for our purposes of constructing a relevance-based loss.

The aggregated map is initialized using identity matrix: $\mathbf{R}(i) := I_{n_i}$. This is because before applying any attention, each token has no context, i.e., each token is self-contained, and has an influence of 1 on itself. The relevance map of each self-attention layer of the Vision Transformer is calculated as follows:

$$\nabla \mathbf{A} := \frac{\partial \mathcal{M}(i)}{\partial \mathbf{A}}; \qquad \bar{\mathbf{A}} = \mathbb{E}_h((\nabla \mathbf{A} \odot \mathbf{A})^+), \tag{8}$$

where $\mathbf{A}$ is the attention map of the current layer, $\odot$ is the Hadamard product, $\mathcal{M}$ is the Vision Transformer, $\mathcal{M}(i)$ is the logit that corresponds to the class we wish to visualize, and $\mathbb{E}_h$ is the mean across the heads dimension. Put differently, the method integrates the gradients of the desired output logit w.r.t. the attention map, in order to average across the attention heads and produce a single unified attention relevance map. The unified attention map $\bar{\mathbf{A}}$ is considered the relevance map of the attention layer.

Finally, to incorporate each layer's explainability map into the accumulated relevance maps the following propagation rule for self-attention layers is applied during a forward pass on the attention layers:

$$\mathbf{R}(i) \leftarrow \mathbf{R}(i) + \bar{\mathbf{A}} \cdot \mathbf{R}(i), \tag{9}$$

where $\bar{\mathbf{A}}$ is the attention relevance map for the self-attention layers, which is calculated using Eq. 8. To extract the relevance of each image token, the row of $\mathbf{R}(i)$ that corresponds to the $[\texttt{CLS}]$ token is used since the $[\texttt{CLS}]$ token alone determines the classification.

In this work, rather than using the relevance map as a form of explainability for a fixed model, we regularize the network to obtain the desired relevance map.

C Baseline description

Both GradMask and RRR employ two loss functions. First, a classic cross-entropy loss with the ground-truth labels to ensure correct labeling, and second, gradients of the output of the model w.r.t. the input are used as a form of explanation, to limit the relevance scores of irrelevant parts of the input. This notion resembles our background loss (Eq. 1) with gradients as the relevance map.

In the following, we describe the loss terms applied in each method. For both methods, a standard cross-entropy loss with the ground truth class is applied to maintain accuracy, i.e.:

$$\mathcal{L}_{\text{classification}} = \text{CE}\left(\mathcal{M}(i), y_i\right), \tag{10}$$

where i is the input image, and y_i is a one-hot vector, where the ground truth classification of i is assigned the value 1.

GradMask applies a gradient-based loss to ensure that the gradients of the background are close to 0:

$$\mathcal{L}_{\text{bg}} = \left\| \frac{\partial \hat{y}_i}{\partial i} \cdot \bar{\mathbf{S}}(i) \right\|_2, \tag{11}$$

where $\hat{y}_i$ is the predicted output for the ground-truth class, and $\bar{\mathbf{S}}(i)$ is, as before, the reversed segmentation map for the ground-truth class. Eq. 11 resembles our background loss (Eq. 1) with simple gradients w.r.t. the input image instead of the relevance map produced by GAE.

Similarly, Right for the Right Reasons (RRR) applies a loss to restrain the magnitude of explanations outside the relevant information. Their relevance loss is obtained as follows:

$$\mathcal{L}_{\text{bg}} = \left(\frac{\partial \sum_{k=1}^{K} \log(p_{i,k})}{\partial i} \cdot \bar{\mathbf{S}}(i) \right)^2, \tag{12}$$

where $k = 1, ..., K$ are the possible output classes, and $p_{i,k}$ is the probability assigned to the k-th class for image i by the model.

For both GradMask and RRR, the following loss is used $\mathcal{L}_{\text{final}} = \lambda_{\text{bg}} \cdot \mathcal{L}_{\text{bg}} + \lambda_{\text{classification}} \cdot \mathcal{L}_{\text{classification}}$, where $\lambda_{\text{bg}}, \lambda_{\text{classification}}$ are hyperparameters. We note that while using the gradient of the output w.r.t. the input is common practice for interpreting CNNs, these gradients are less stable for transformer-based models. For example, results presented in [35] demonstrate that for transformer-based models the classic Input$\times$Gradient method violates faithfulness.

In our experiments, we found it difficult to grid-search hyperparameters to fit $\mathcal{L}_{\text{bg}}$ and $\mathcal{L}_{\text{classification}}$. Furthermore, we had to tune $\lambda_{\text{bg}}, \lambda_{\text{classification}}$ for each model separately to obtain an improvement for $\mathcal{L}_{\text{bg}}$. Our method, on the other hand, uses the same hyperparameter choice (see Sec. 3), which makes it far more stable to use, thus allowing us to run experiments on large models as well. We refer the reader to Appendix D for the full description of hyperparameters used in our experiments.

D Hyperparameters

In Tab. 5 we present the hyperparameter selection for all our experiments. Our method is stable and uses the same selection in all cases, other than the learning rate. The learning rates range from $6e-7$ to $3e-6$, allowing for a quick and easy grid search.

Tab. 6, 7 represent the hyperparameters of RRR, GradMask, respectively. For RRR, we had to tune the parameters per model, and the method is sensitive to the specific selection. For GradMask, we had to carefully tune the learning rate for each model. We found that the results of GradMask are sensitive to the specific learning rate selection, and to minor changes in the learning rate. We found it difficult to get the background loss to converge in both cases.

Table 5: Hyperparameter selection for our method for all models- ViT [15], ViT AugReg [54], and DeiT [57]. All hyperparameters are fixed except for the learning rate.

Model	$\lambda_{\text{classification}}$	$\lambda_{\text{relevance}}$	λ_{bg}	λ_{fg}	Learning rate
ViT-B	0.2	0.8	2	0.3	$3e-6$
ViT-L	0.2	0.8	2	0.3	$9e-7$
AR-S	0.2	0.8	2	0.3	$2e-6$
AR-B	0.2	0.8	2	0.3	$6e-7$
AR-L	0.2	0.8	2	0.3	$9e-7$
DeiT-S	0.2	0.8	2	0.3	$1e-6$
DeiT-B	0.2	0.8	2	0.3	$8e-7$

Table 6: Hyperparameter selection for the Right for the Right Reasons [44] method for all models- ViT [15], ViT AugReg [54], and DeiT [57]. Hyperapramters vary according to the model.

Model	$\lambda_{\text{classification}}$	λ_{bg}	Learning rate
ViT-B	$2e-6$	$1e-10$	$2e-6$
AR-S	$2e-8$	$1e-8$	$1e-5$
AR-B	$2e-7$	$1e-8$	$5e-7$
DeiT-S	$2e-6$	$1e-10$	$1e-5$
DeiT-B	$2e-6$	$1e-10$	$5e-6$

Table 7: Hyperparameter selection for the GradMask [50] method for all models- ViT [15], ViT AugReg [54], and DeiT [57].

Model	$\lambda_{\text{classification}}$	λ_{bg}	Learning rate
ViT-B	$3e-9$	50	0.0002
AR-S	$3e-9$	50	0.0005
AR-B	$3e-9$	50	$1e-5$
DeiT-S	$3e-9$	50	0.005
DeiT-B	$3e-9$	50	0.001

E Random seed selection of the training classes

Tab. 8, and Tab. 9 present the results for the main experiment for multiple seeds, extending Tab. 1 and Tab. 2 of the main text. Each seed changes the 500 random classes used for finetuning with exactly the same hyperparameters.

As can be seen, the Standard Deviation is not large, especially in comparison to the performance gap. In all shifted-distribution datasets, the original result, before our intervention, is outside the standard error range for the multiple-seed experiments.

Table 8: Results using 3 random seeds. The table presents the average and standard deviation of the top-1 accuracy per dataset.

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
ViT-B	Original	**81.5**	16.0	33.8	35.4	**71.1**	35.1	33.3	39.1	55.6
	Ours	80.3±0.1	**23.6±0.7**	**36.1±0.3**	**36.3±0.1**	70.1±0.3	**41.7±0.6**	**38.6±0.2**	**46.1±0.4**	**60.9±0.4**
ViT-L	Original	**82.9**	19.0	36.6	40.4	**71.8**	37.4	31.6	40.7	54.8
	Ours	82.0±0.1	**25.0±0.3**	**38.6±0.4**	**41.1±0.1**	71.2±0.2	**42.6±0.2**	**36.6±0.4**	**45.3±0.4**	**58.6±0.5**
AR-S	Original	**81.4**	13.0	31.2	32.8	**69.9**	34.3	32.4	40.6	55.4
	Ours	79.8±0.2	**18.3±0.8**	**34.1±0.2**	**33.6±0.1**	69.4±0.2	**39.1±0.4**	**36.4±0.9**	**47.6±0.9**	**61.0±0.8**
AR-B	Original	**84.4**	23.9	41.0	43.1	**73.8**	41.4	40.5	48.1	60.6
	Ours	83.0±0.1	**30.9±0.3**	**45.3±0.7**	**44.9±0.4**	73.3±0.2	**47.0±0.6**	**43.8±0.7**	**54.7±0.6**	**64.6±0.5**
AR-L	Original	**85.6**	34.7	48.8	51.8	**75.8**	46.5	43.8	52.4	62.3
	Ours	84.8±0.3	**42.2±0.4**	**53.9±0.3**	**54.0±0.3**	75.7±0.1	**51.7±0.2**	**48.6±0.6**	**57.8±0.8**	**66.6±0.3**
DeiT-S	Original	78.1	8.3	28.2	28.8	66.5	28.3	30.7	36.7	51.6
	Ours	**78.7±0.1**	**10.4±0.3**	**29.3±0.1**	**29.0±0.2**	**67.3±0.3**	**31.5±0.1**	**32.2±0.2**	**40.6±0.4**	**55.6±0.4**
DeiT-B	Original	**80.8**	12.9	30.9	**31.2**	**69.7**	31.4	34.5	39.3	54.6
	Ours	80.6±0.2	**17.2±0.2**	**32.7±0.3**	**31.2±0.3**	69.3±0.2	**35.9±0.2**	**37.0±0.4**	**43.3±0.4**	**58.3±0.3**

Table 9: Results using 3 random seeds. The table presents the average and standard deviation of the top-5 accuracy per dataset.

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
ViT-B	Original	**96.0**	37.0	48.5	57.4	**89.9**	56.4	52.2	58.3	76.2
	Ours	95.4±0.0	**47.1±0.8**	**51.3±0.2**	**58.7±0.3**	89.4±0.1	**64.7±0.5**	**57.5±0.3**	**66.6±0.5**	**81.2±0.3**
ViT-L	Original	**96.4**	41.5	52.0	63.4	**90.7**	59.5	50.3	60.1	75.6
	Ours	96.2±0.0	**49.2±0.6**	**54.6±0.3**	**64.3±0.1**	90.6±0.3	**65.5±0.2**	**56.4±0.3**	**66.3±0.5**	**80.3±0.3**
AR-S	Original	**96.1**	33.9	47.1	54.2	**90.1**	55.8	51.7	59.6	75.7
	Ours	95.6±0.1	**40.9±1.1**	**50.2±0.4**	**55.3±0.1**	89.9±0.1	**61.8±0.6**	**56.1±0.8**	**67.6±0.8**	**81.2±0.5**
AR-B	Original	**97.2**	49.2	57.8	65.7	**92.3**	63.7	60.8	68.3	80.4
	Ours	96.8±0.1	**56.8±0.3**	**62.0±0.7**	**67.9±0.6**	91.9±0.1	**69.9±0.5**	**63.5±0.8**	**75.1±0.6**	**84.4±0.5**
AR-L	Original	**97.8**	61.0	64.9	73.6	**93.4**	68.3	64.2	72.5	82.2
	Ours	97.4±0.1	**67.3±0.6**	**69.2±0.4**	**75.7±0.2**	93.3±0.1	**73.5±0.4**	**68.8±0.7**	**78.1±0.8**	**86.3±0.3**
DeiT-S	Original	93.7	23.5	41.9	46.7	86.6	47.3	50.4	54.3	72.0
	Ours	**94.5±0.1**	**29.0±0.5**	**43.6±0.0**	**47.6±0.4**	**87.4±0.1**	**53.0±0.1**	**51.3±0.2**	**59.3±0.5**	**76.1±0.3**
DeiT-B	Original	94.2	31.0	44.2	48.6	86.8	48.5	54.6	56.3	73.4
	Ours	**95.0±0.1**	**40.1±0.2**	**47.4±0.4**	**49.7±0.4**	**88.4±0.1**	**56.3±0.2**	**57.4±0.4**	**61.9±0.5**	**78.4±0.3**

F Additional qualitative results

In Fig. 6, 7 we present additional qualitative examples to supplement Fig. 2, 3, respectively. As can be seen from Fig. 6, in accordance with the results in the main paper, while performance sometimes decreases with respect to the assigned label, in most cases the rationale of the class proposed by the modified model is clear, often indicating a more natural labeling option. In addition, in the cases where the prediction remains the same or is corrected, our method produces improved relevance maps.

Fig. 7 provides more examples of cases where the salient behavior of the models prevents them from producing correct predictions. For example, a vulture is classified as a kite due to the trees in the background (first example in the fourth line), a porcupine is classified as a sea lion due to the presence of the ocean (third example in the second line), and a teddy bear is classified as a ping-pong ball due to sparse consideration of the foreground that is focused on its paw (second example in the second line). By correcting the relevance maps, our method assists the models to achieve an accurate prediction.

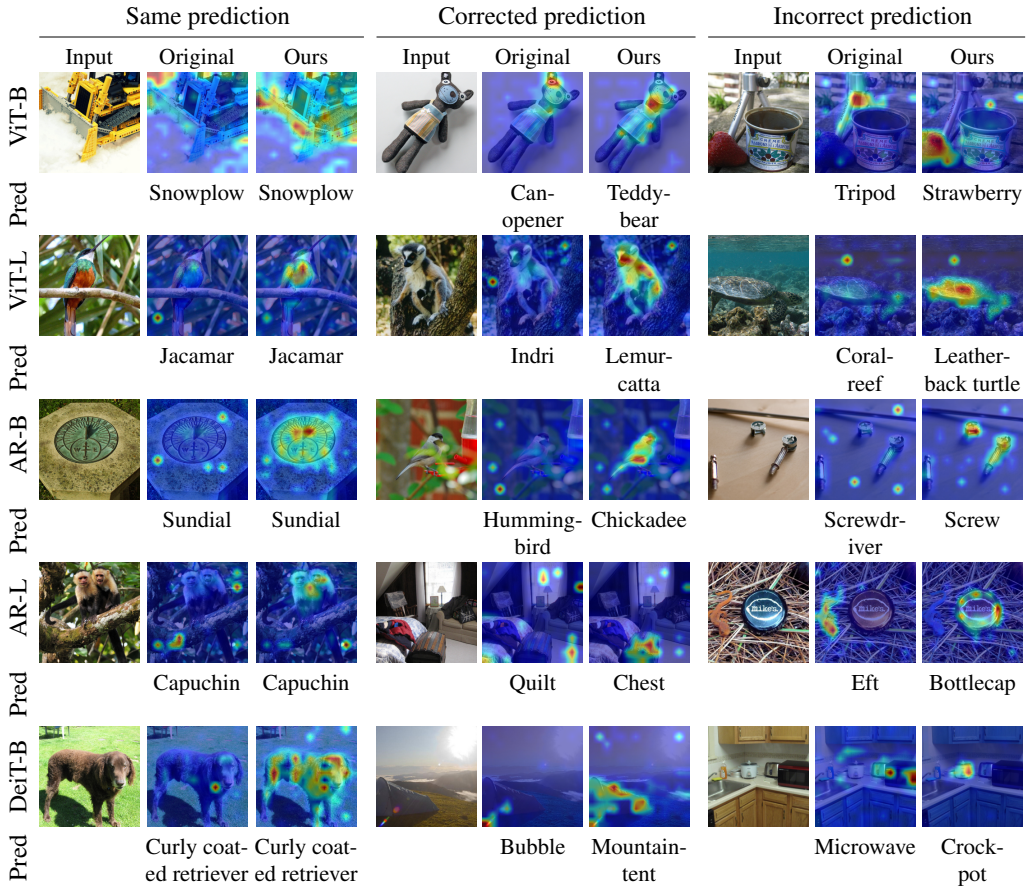


Figure 6: Examples from the ImageNet validation set of cases in which our method does not change the prediction, corrects the prediction, and ruins the prediction. Even in cases where our method changes a correct prediction, there is often a rationale behind the modified prediction.

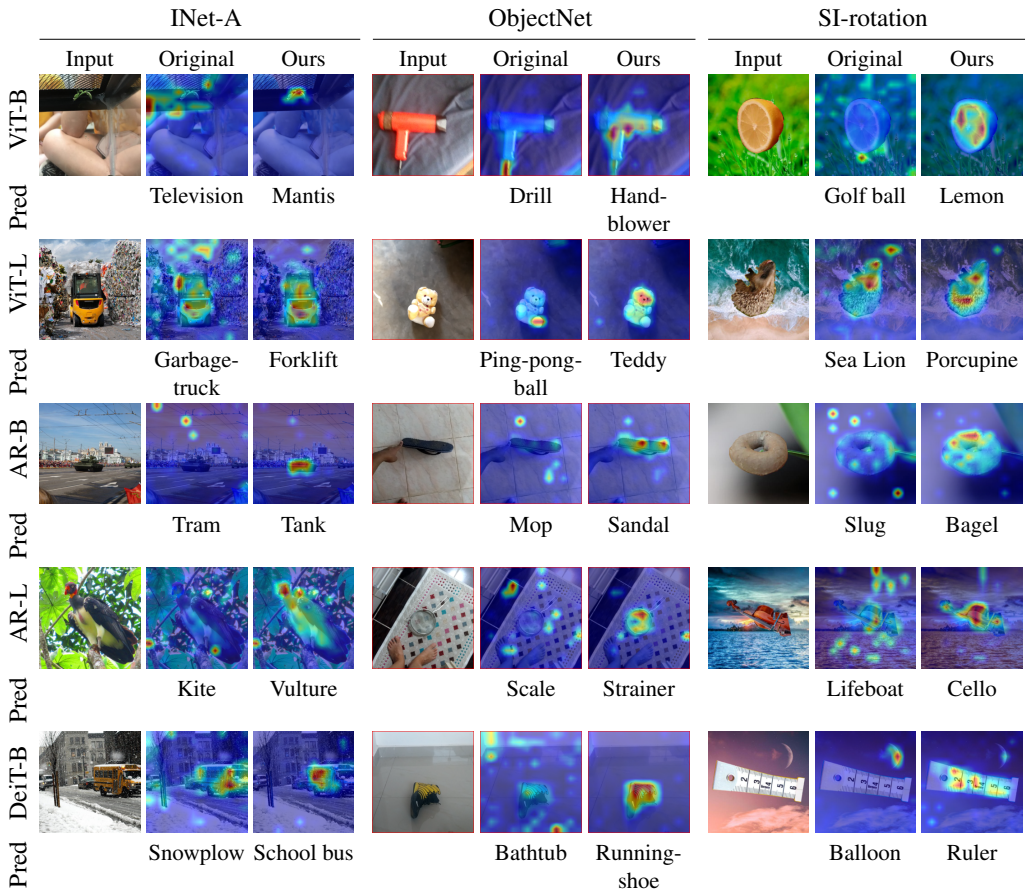
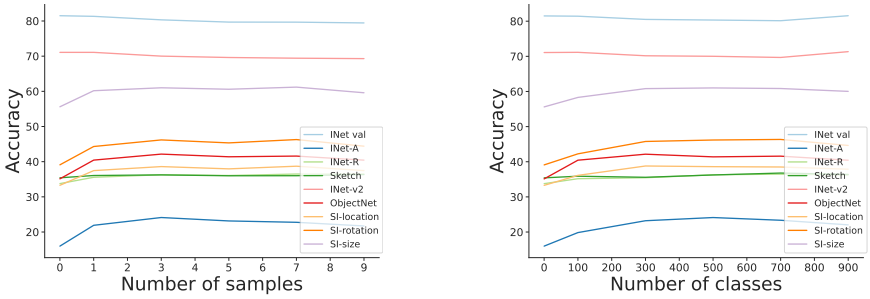


Figure 7: Examples of cases where our method corrects wrong predictions, alongside the original and modified (after finetuning) explainability maps (please zoom in for a better view). The “Pred" row specifies the predictions before and after our finetuning. The examples demonstrate cases where the original classifier relies on partial or irrelevant data, while our method rectifies the classification to be based on the object. The examples are presented for the large and base models of ViT [15], ViT AugReg [54] (AR), and the base model of DeiT [57].

G Sensitivity tests

Fig. 8, presents sensitivity tests evaluating our robustness results on increasing number of samples per class (panel a), and increasing number of classes (panel b). Evidently, three samples for half the classes suffice to achieve the maximal improvement in robustness, while minimally harming the performance on ImageNet-based datasets (ImageNet val, ImagNet-v2). Using only two samples or considerably fewer classes does not harm performance much.



(a) (b)

Figure 8: Evaluation of our method's sensitivity to (a) the number of samples used per class in the training set (we use $n = 3$), and (b) number of training classes (we use $c = 500$), on ViT-B [15]. As can be seen, all the presented combinations of hyperparameters result in a significant increase in robustness, and a relatively modest decrease in INet val, INet-v2 accuracy.

H Comparing training classes to the other classes

Tab. 10, 11 contain the full results of our experiment comparing the effect of our method on the classes in the training set and the classes outside of it. As can be seen, both the training and non-training classes benefit very similarly from applying our method when evaluated in terms of robustness on both real-world and synthetic datasets.

Table 10: Robustness evaluation on the classes that were included in the training set of our finetuning and classes that were not. The results are presented for the base models of ViT [15], ViT AugReg [54], and DeiT [57]. The last row indicates the average change for the training and non-training classes across the models for each dataset.

Model	Train classes		INet val		INet-A		INet-R		Sketch		INet-v2		ObjNet	
			R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
ViT-B	✓	Original	**82.0**	**96.0**	14.8	35.2	32.9	47.4	36.8	58.4	71.3	89.8	36.0	55.8
		Ours	**82.0**	95.7	**22.9**	**46.1**	**35.5**	**50.5**	**38.1**	**59.7**	**71.4**	**89.9**	**43.5**	**64.6**
	✗	Original	**81.0**	**95.9**	17.2	38.7	34.7	49.6	34.0	56.4	**70.9**	**90.1**	33.9	57.0
		Ours	78.6	95.1	**25.3**	**49.8**	**37.1**	**52.3**	**34.4**	**57.4**	68.6	88.9	**40.3**	**65.7**
AR-B	✓	Original	**83.8**	**97.2**	25.3	51.6	39.8	56.9	44.7	66.5	74.4	**92.3**	39.8	64.9
		Ours	81.2	96.7	**32.2**	**59.1**	**44.6**	**60.9**	**47.2**	**68.7**	**74.9**	92.2	**45.2**	**70.9**
	✗	Original	**85.0**	**97.3**	22.6	46.7	42.3	58.7	41.5	64.9	**73.3**	**92.4**	42.6	62.7
		Ours	**85.0**	97.2	**30.3**	**54.9**	**44.8**	**62.2**	**42.0**	**66.1**	72.1	91.9	**48.4**	**69.2**
DeiT-B	✓	Original	**81.7**	94.4	11.7	29.4	29.6	42.9	**32.0**	49.3	**70.3**	86.5	32.7	48.8
		Ours	**81.7**	**95.2**	**15.2**	**37.5**	**30.9**	**45.5**	31.5	**49.6**	69.4	**88.4**	**36.6**	**55.3**
	✗	Original	**79.9**	94.0	14.1	32.5	32.4	45.7	**30.4**	48.0	**69.0**	87.0	29.8	48.1
		Ours	79.3	**94.7**	**19.1**	**42.5**	**33.9**	**48.5**	30.3	**48.8**	68.9	**88.3**	**34.8**	**57.3**
Avg. change	✓		0	+0.1	+6.4	+9.1	+2.9	+3.2	+1.1	+1.3	+0.5	+0.6	+5.7	+7.3
	✗		-1.9	-0.2	+6.7	+9.5	+2.1	+3.0	+0.3	+1.0	-1.2	-0.1	+5.6	+8.0

Table 11: Robustness evaluation on the classes that were included in the training set of our finetuning and classes that were not for the synthetic datasets. The results are presented for the base models of ViT [15], ViT AugReg [54] (AR), and DeiT [57]. The last row indicates the average change for the training and non-training classes across the models for each dataset.

Model	Train classes		SI-loc.		SI-rot.		SI-size	
			R@1	R@5	R@1	R@5	R@1	R@5
ViT-B	✓	Original	34.2	52.5	40.5	60.3	56.5	76.1
		Ours	**38.6**	**58.0**	**46.3**	**68.1**	**60.4**	**80.4**
	✗	Original	32.0	51.8	37.1	55.6	54.4	76.3
		Ours	**38.6**	**57.5**	**46.0**	**65.6**	**61.9**	**82.9**
AR-B	✓	Original	41.2	60.7	50.0	69.2	61.6	79.9
		Ours	**43.8**	**62.7**	**55.5**	**75.3**	**64.7**	**83.6**
	✗	Original	39.7	60.8	45.5	67.0	59.1	81.0
		Ours	**42.5**	**62.9**	**51.9**	**73.7**	**63.2**	**84.4**
DeiT-B	✓	Original	35.2	54.2	40.3	56.3	56.1	72.3
		Ours	**37.1**	**56.6**	**43.5**	**61.6**	**59.0**	**77.0**
	✗	Original	33.6	55.1	38.0	56.1	52.5	74.8
		Ours	**35.8**	**57.5**	**42.1**	**61.4**	**56.7**	**79.7**
Avg. change	✓		+3.0	+3.3	+4.8	+6.4	+3.3	+4.2
	✗		+3.7	+3.4	+6.5	+7.3	+5.3	+5.0

I Ablation study

Tab. 12 presents the top-5 accuracy results of our ablation study to complement Tab. 4. As can be seen, the top-5 results are consistent with the top-1 results from Tab. 4.

Table 12: Ablation study for our method on the ViT [15] and DeiT [57] base models. The table presents top-5 accuracy. We check the impact of each term of our loss on the robustness of the model, as well as the choice of confidence boosting versus cross-entropy with the ground-truth label (labeled as w/ ground-truth).

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
ViT-B	Original	96.0	37.0	48.5	57.4	89.9	56.4	52.2	58.3	76.2
	Ours	95.4	**48.0**	**51.4**	**58.5**	89.4	**65.1**	57.8	**67.0**	**81.4**
	w/o $\mathcal{L}_{\text{classification}}$ (Eq. 4)	94.3	38.9	50.3	57.9	87.4	61.6	56.2	63.8	78.9
	w/o $\mathcal{L}_{\text{bg}}$ (Eq. 1)	**96.1**	41.8	50.9	**58.5**	**90.5**	62.1	54.1	62.7	78.9
	w/o $\mathcal{L}_{\text{fg}}$ (Eq. 2)	95.3	47.2	48.7	57.0	89.2	64.4	**58.3**	66.2	81.2
	w/ ground-truth	**96.1**	45.2	50.4	57.9	90.4	63.2	57.1	64.8	80.3
	only $\mathcal{L}_{\text{bg}}$	85.1	35.9	34.9	45.2	75.9	53.2	49.4	52.6	70.9
	only $\mathcal{L}_{\text{fg}}$	94.6	24.8	47.7	56.6	87.3	53.0	44.0	52.8	71.7
	Attention instead of [9]	95.7	44.8	50.0	57.3	89.9	62.5	56.1	62.5	79.0
	Rollout instead of [9]	94.9	47.1	50.0	55.8	88.7	63.8	53.9	66.6	80.6
DeiT-B	Original	94.2	31.0	44.2	48.6	86.8	48.5	54.6	56.3	73.4
	Ours	94.9	40.0	47.0	49.2	88.3	56.2	57.0	61.5	78.2
	w/o $\mathcal{L}_{\text{classification}}$ (Eq. 4)	**95.2**	37.4	**48.4**	**51.4**	**88.9**	**56.8**	**60.1**	**63.1**	**79.5**
	w/o $\mathcal{L}_{\text{bg}}$ (Eq. 1)	**95.2**	34.4	44.7	47.9	88.3	52.4	53.2	57.5	74.7
	w/o $\mathcal{L}_{\text{fg}}$ (Eq. 2)	94.9	**40.1**	47.2	49.3	88.1	56.4	57.6	61.4	78.5
	w/ ground-truth	95.0	39.6	46.5	49.6	88.2	55.9	57.3	61.2	78.0
	only $\mathcal{L}_{\text{bg}}$	85.9	26.3	35.4	39.7	75.6	44.1	55.0	51.2	70.2
	only $\mathcal{L}_{\text{fg}}$	94.8	22.7	42.2	46.2	87.4	47.0	46.9	53.2	71.1
	Attention instead of [9]	**95.2**	37.5	46.2	49.4	88.7	53.5	56.5	59.7	76.7
	Rollout instead of [9]	**95.2**	35.7	45.7	48.6	88.4	52.5	54.7	58.4	76.0

J Per class robustness analysis

There is considerable variability in the effect of our method between the different classes. This is similar to the varying effect of regularization reported by Balestriero et al. [4] and it is reasonable to assume that it is a common phenomenon among regularization techniques.

Fig. 9-15 depict the classes that benefited the most by our finetune process on ViT-B and those that were harmed the most. For datasets with $1,000$ classes, we present the 50 classes with the most beneficial or harmful change for readability. Other datasets are presented with all their classes.

Inspecting the classes with the largest amount of change, we can observe a few trends. In some cases, classes with relatively small objects considerably benefit, and classes with objects that either reside in the background or benefit from the context are harmed, as can be expected. In some cases, classes with a small number of test samples, regardless of the type of object, lead to a higher absolute change, due to statistical reasons.

In order to study the limitations of our method, we inspect the classes where it fails the most for INet-A and INet-V2. These two test sets represent different amounts of domain shift from the original ImageNet (INet-V2 is very close to ImageNet, while INet-A is out-of-distribution). As shown in the paper, our method increases robustness to domain shifts.

Fig. 16 depicts the two classes, out of each of the two test sets (INet-A and INet-V2) that were harmed the most: pool table and breastplate for INet-A and breastplate and miniature poodle for INet-v2. From each class, we show the effect of our method on the first three samples in which a correct classification before finetuning turned into a wrong classification (so we avoid cherry-picking).

The pool table samples tend to show other objects in the foreground. In the top-left sample, the object is identified as a bucket (it is a hat, which is the 2nd top prediction). In the other two samples for pool tables and INet-A, the foreground object is correctly identified. In all three cases, the heatmap explains the result. In the breastplate case, there are only 11 test samples, two of which were wrongly classified by our method as other, reasonable classes.

For INet-v2, following our method, some breastplates are identified as cuirass (oxford dictionary: a piece of armor consisting of breastplate and backplate fastened together), which is a similar class. In the 2nd most affected class, miniature poodles are identified as toy poodles, or as a Lhasa Apso, which are similar dog breeds.

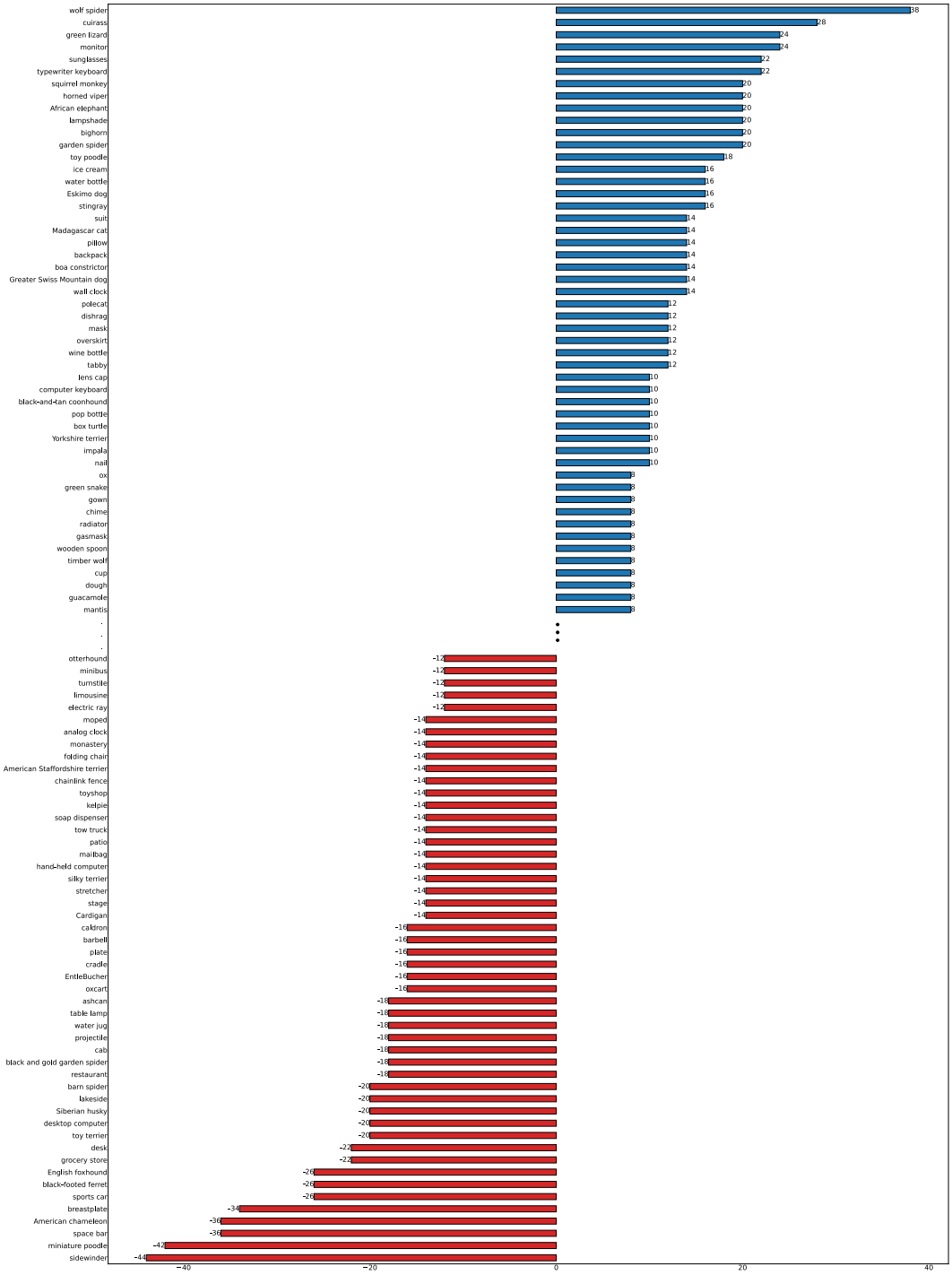


Figure 9: The 50 classes which most benefited and the 50 classes that were most harmed by the finetune procedure for ImageNet.

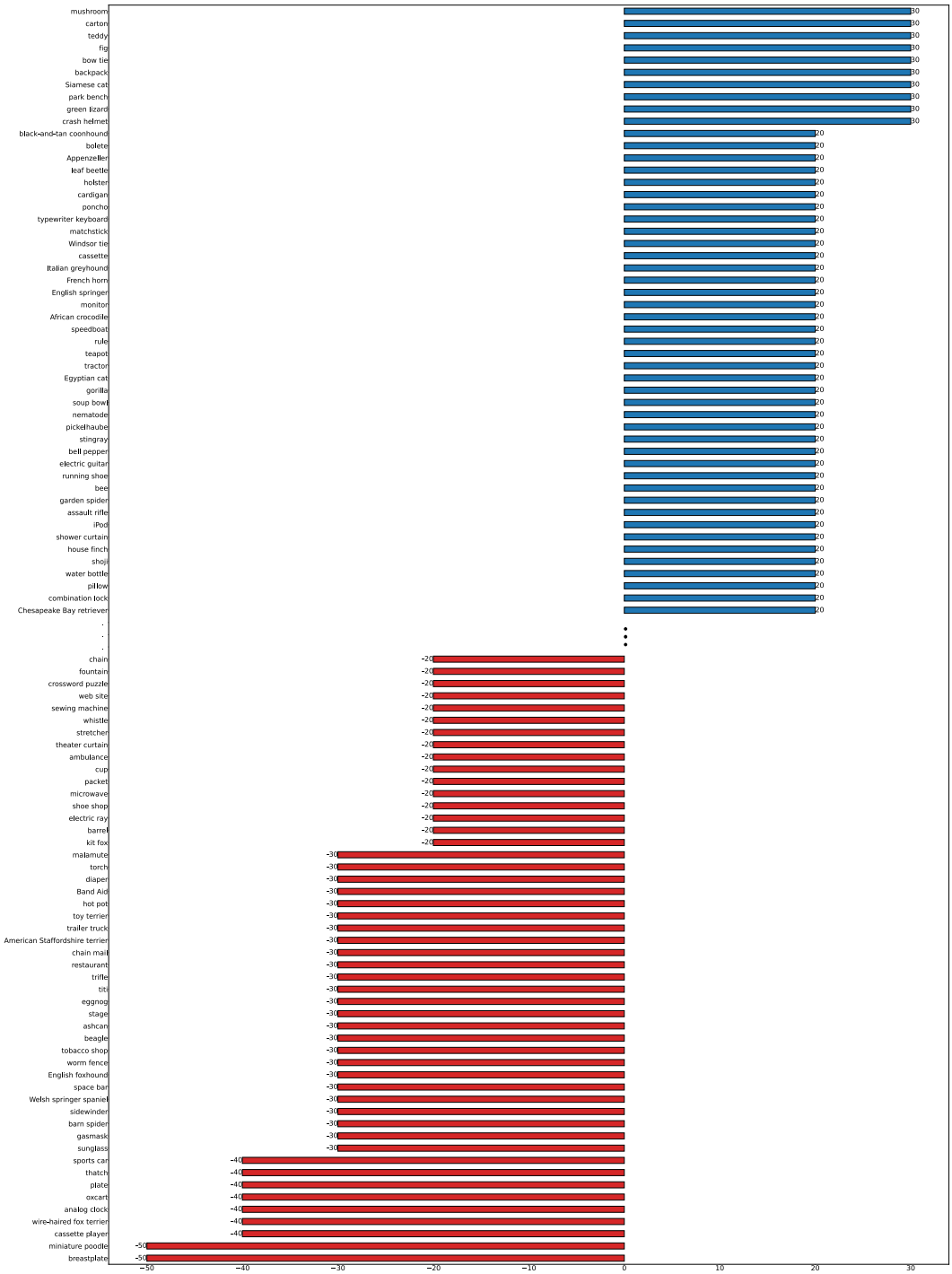


Figure 10: The 50 classes which most benefited and the 50 classes that were most harmed by the finetune procedure for ImageNet-v2.

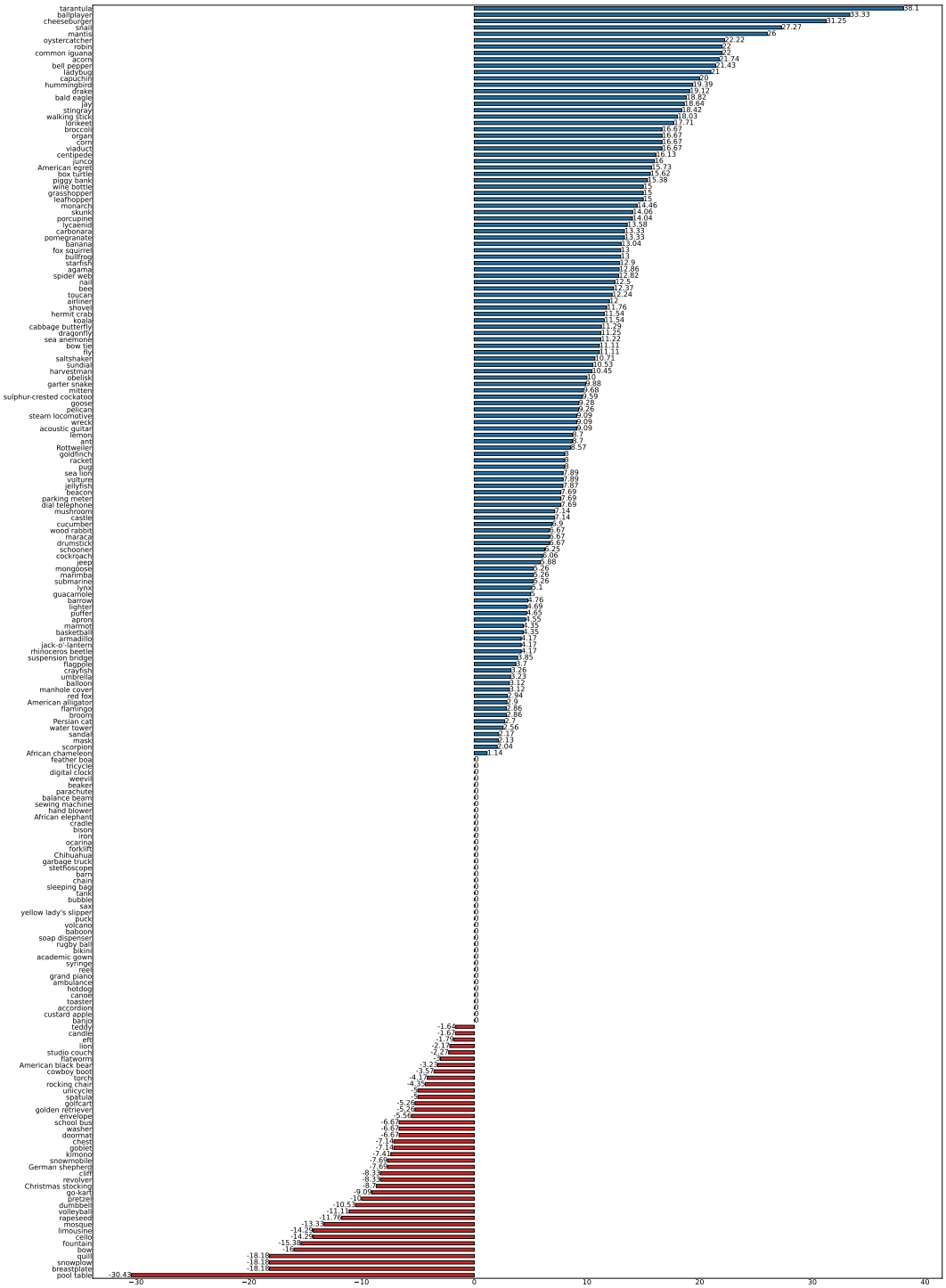


Figure 11: The effect of the finetune procedure on each class for ImageNet-A (there are some classes with zero effect).

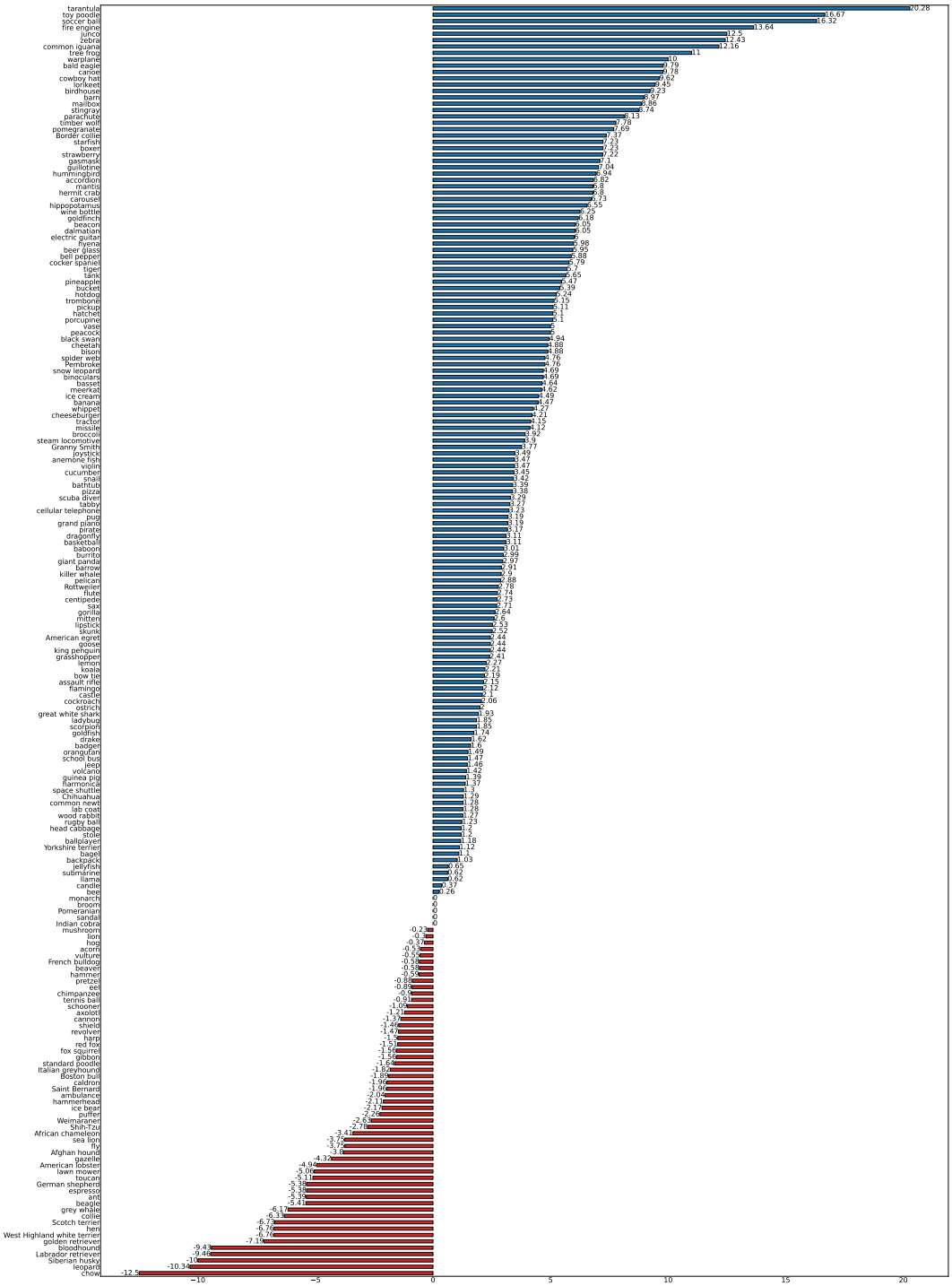


Figure 12: The effect of the finetune procedure on each class for ImageNet-R (there are some classes with zero effect).

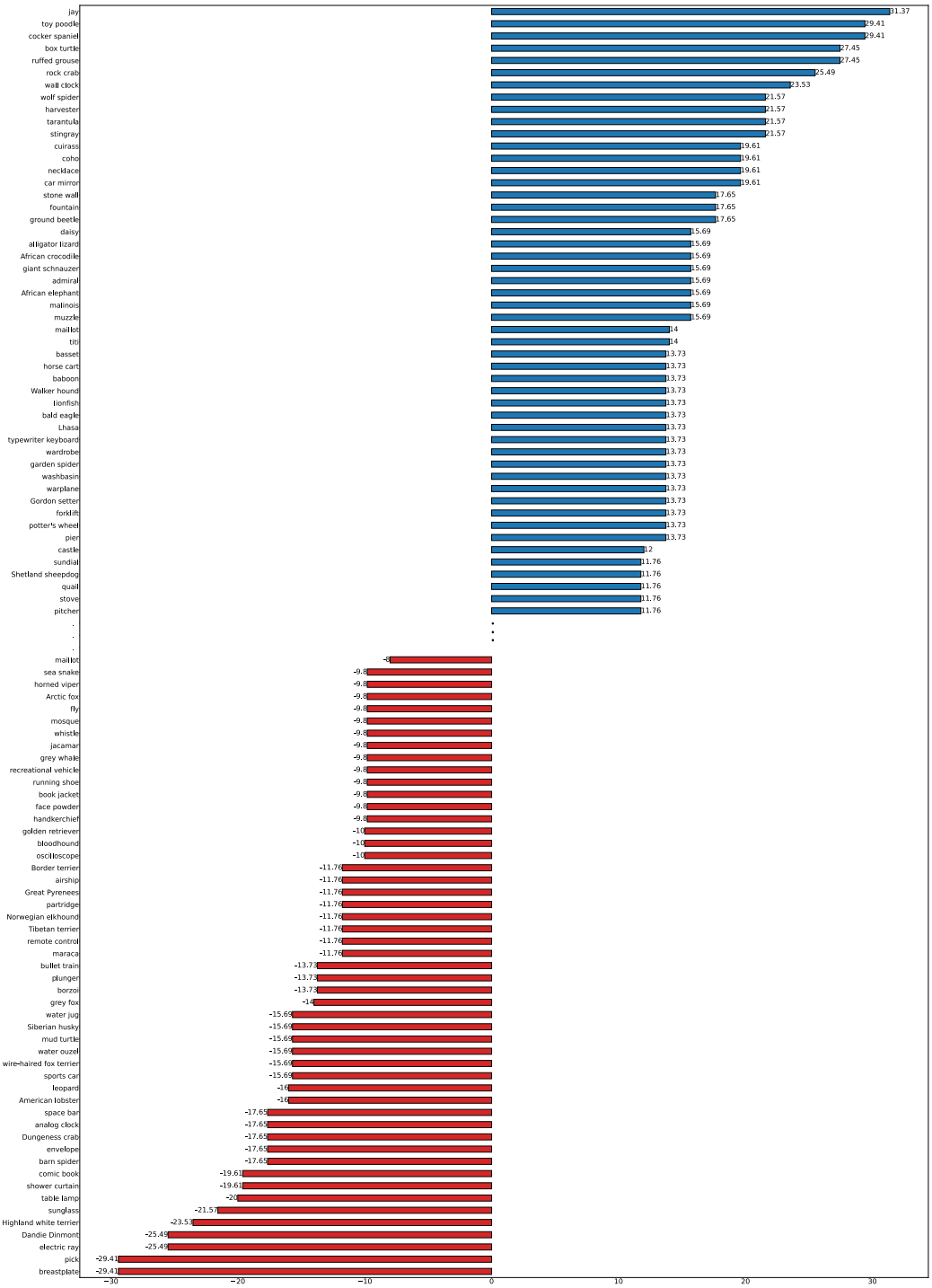


Figure 13: The 50 classes which most benefited and the 50 classes that were most harmed by the finetune procedure for ImageNet-sketch.

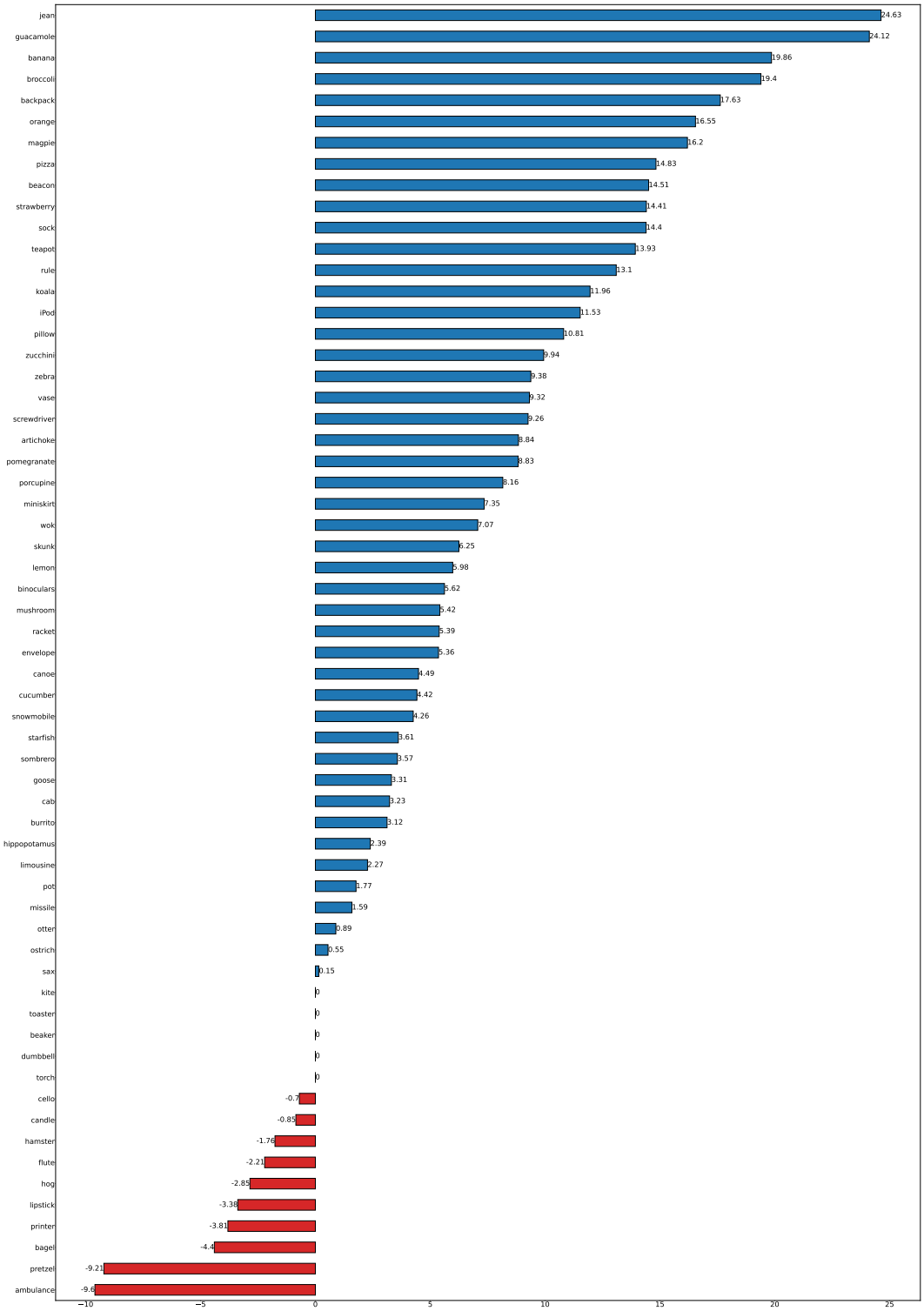


Figure 14: The effect of the finetune procedure on each class for SI-rotation (there are some classes with zero effect).

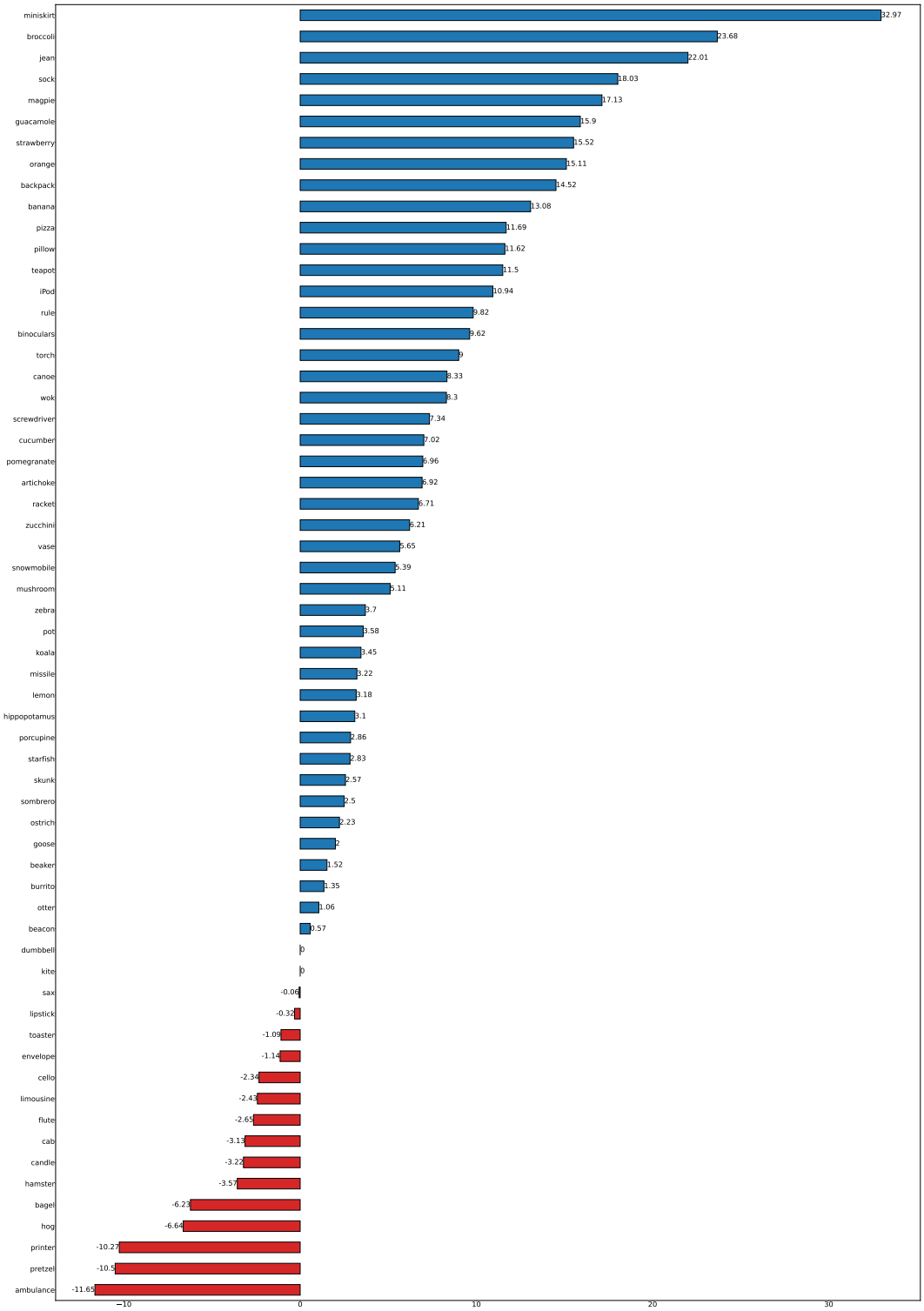


Figure 15: The effect of the finetune procedure on each class for SI-size (there are some classes with zero effect).

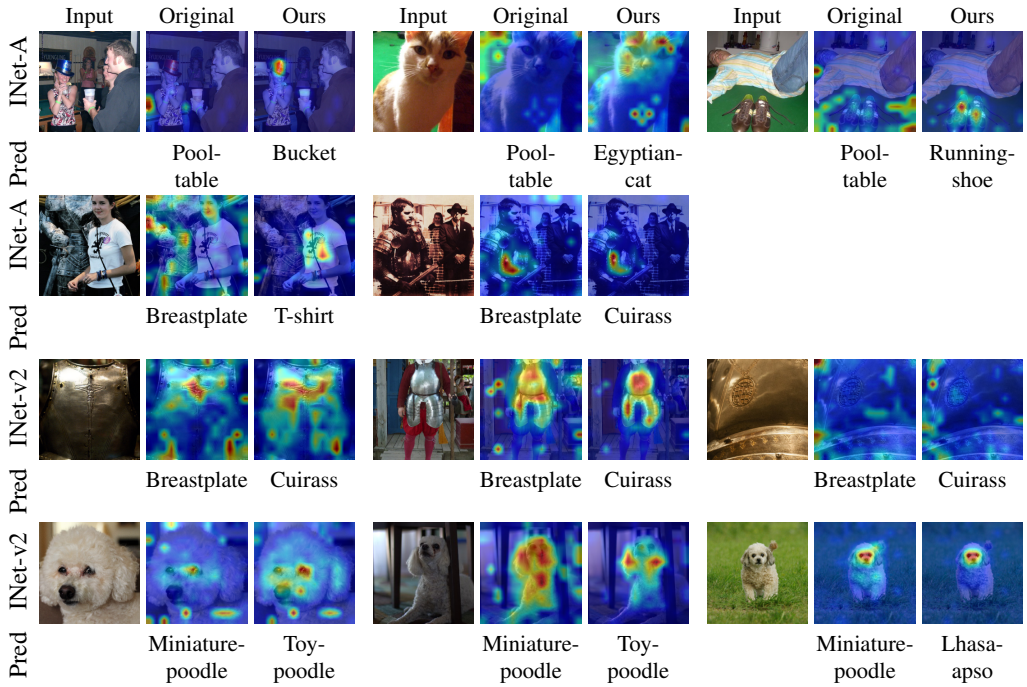


Figure 16: Examples of the 2 classes most harmed by our method for a dataset that is out of distribution (INet-A), and a dataset that has a similar distribution to ImageNet (INet-v2). For each dataset, we show the first three examples where our method modified a correct prediction and made it wrong (by showing the first three samples, we demonstrate that these samples are typical and not cherry-picked). As can be seen, in most cases the heatmaps are improved by our method, and the wrong prediction has a rationale (see text for examples). For INet-A there are only 2 mistakes in the Breastplate class (out of a total of 11 examples in the dataset), therefore the corresponding row (second row) only contains 2 examples.

K Comparison to self-supervised Vision Transformers

Our unsupervised fine-tuning is based on the DINO self-supervised Vision Transformer [6] (see Sec. 4). DINO's attention maps have demonstrated the ability to perform semantic segmentation without being trained with that objective. This property partially corresponds to the goal of our fine-tuning process (see Sec. 3).

In this section, we perform a comparison between the robustness granted by our method and the improved robustness by DINO. We compare two versions of DINO- the first, a self-supervised DINO with a linear head trained on ImageNet, and the second, a DINO network that was fine-tuned on ImageNet. For the linear probing version, the DINO backbone was frozen and a linear head was trained over the features by the frozen network, while for the fine-tuned version, the entire network was fine-tuned on ImageNet in a supervised manner.

Since DINO's code and fine-tuning are based on DeiT [57], we present a comparison between the original DeiT model (trained on ImageNet), our fine-tuned version of DeiT, and DINO. Additionally, we apply our method to the DINO version that was fine-tuned on ImageNet, since this version fine-tuned the entire network based on the supervised classification task, which may cause changes to DINO's salient behavior. In the latter case, we apply the same hyperparameters as we did for DeiT, i.e. a learning rate of $8e-7$ (all other hyperparameters are identical for all models), for a fair comparison.

We note that throughout this section, we use unsupervised fine-tuning results of our method. This way, our method is applied in a way that is congruent with self-supervised learning.

Table 13: Comparison to self-supervised robustness performance by DINO [6] (Top-1 accuracy). The results are compared to DeiT [57] since DINO's code and fine-tuning are built on it. DINO linear stands for DINO with linear probing, and DINO fine-tuned stands for DINO with fine-tuning on ImageNet. For DINO fine-tuned we apply our method since the entire network was fine-tuned on a supervised classification task. All DINO models were taken from the official implementation.

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
DeiT-B	Original	80.8	12.9	30.9	31.2	69.7	31.4	34.5	39.3	54.6
	Ours	80.5	18.3	32.8	31.5	69.3	36.3	**37.8**	**44.2**	**59.3**
DINO-B	linear	78.0	7.5	25.9	25.4	66.7	26.4	34.8	26.2	51.3
DINO-B	fine-tuned	**82.6**	16.4	34.9	**35.4**	**72.0**	34.9	35.4	40.0	55.3
	Ours	82.2	**19.1**	**35.9**	**35.4**	71.3	**37.4**	36.7	43.7	58.1

Table 14: Comparison to self-supervised robustness performance by DINO [6] (Top-5 accuracy). The results are compared to DeiT [57] since DINO's code and fine-tuning are built on it. DINO linear stands for DINO with linear probing, and DINO fine-tuned stands for DINO with fine-tuning on ImageNet. For DINO fine-tuned we apply our method since the entire network was fine-tuned on a supervised classification task. All DINO models were taken from the official implementation.

Model	Method	INet val	INet-A	INet-R	Sketch	INet-v2	ObjNet	SI-loc.	SI-rot.	SI-size
DeiT-B	Original	94.2	31.0	44.2	48.6	86.8	48.5	54.6	56.3	73.4
	Ours	95.0	40.9	47.5	49.9	88.5	56.6	**58.1**	62.7	**79.0**
DINO-B	linear	93.8	23.5	38.1	42.4	86.7	47.0	53.9	46.2	73.4
DINO-B	fine-tuned	**96.0**	38.8	50.1	54.6	**89.7**	54.7	56.6	59.4	76.2
	Ours	95.7	**42.6**	**51.7**	**54.7**	**89.7**	**58.0**	57.7	**63.1**	78.7

Tab. 13, 14 present the top-1, top-5 results of our experiment, respectively. As can be seen, the linear probing version of DINO performs worst, below the original DeiT-B model. However, the fine-tuned version of DINO significantly improves DeiT accuracy and robustness. It should be noted that even when comparing DINO's fine-tuned version with our version of DeiT, our model outperforms DINO in 5 out of the 7 robustness datasets that are not from the ImageNet distribution (INet-A, ObjNet, SI-loc., SI-rot., SI-size), while the two others (INet-R, INet-Sketch) are datasets

that contain sketches, cartoons, and art, for which our method is less effective (due to the absence of background information). Additionally, applied to the fine-tuned version of DINO, our method improves robustness across all non-ImageNet datasets. Evidently, the fine-tuning of DINO had a negative effect on robustness for non-ImageNet datasets, which our method corrects.

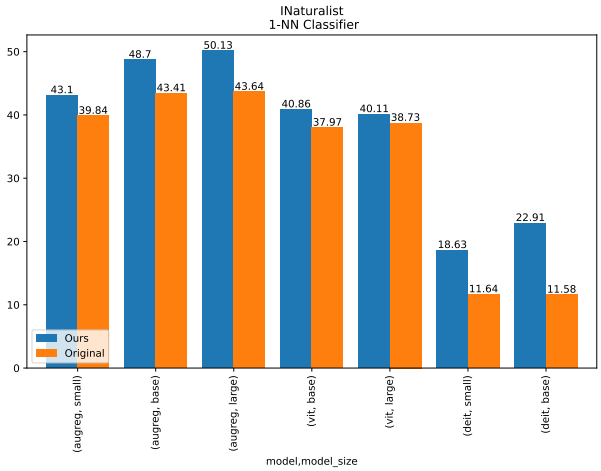
These results further establish our method's significance. Even when a self-supervised robust network is used, the supervised fine-tuning process could cause changes for the worse in the salient behavior of the method, which are rectified by our method.

L Evaluation on non-ImageNet classes

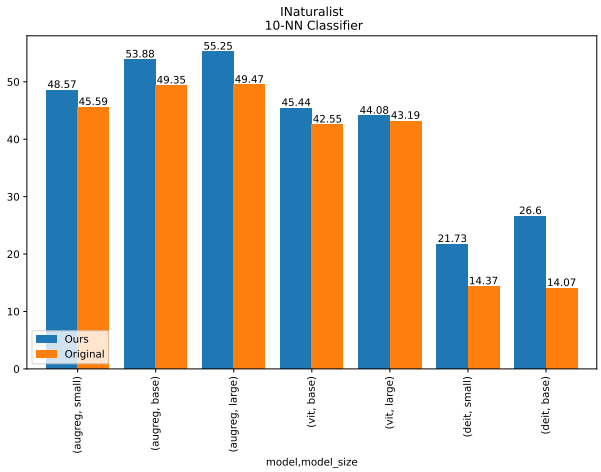
In addition to the evaluation on ImageNet distribution shifts, we test the improvement of robustness on datasets that contain classes that do not appear on ImageNet.

We employ a k-NN test where we remove the classification head from the Vision Transformer, and use the [CLS] token as a global representation of the input. We consider three datasets for this experiment. First, we consider the iNat2021 mini dataset [28] which is a natural visual classification dataset, with $10k$ different classes. Secondly, we employ a medical dataset containing X-ray images of lungs to detect Pneumonia (binary classification) [30]. Finally, we consider the CIFAR-100 [32] dataset, which is an object categorization dataset containing 20 classes.

Fig. 17, 18, 19 present results for these three datasets. We employ both $k = 1$ and $k = 10$ neighbors, in order to demonstrate the effectiveness of our method in multiple settings ($k = 10$ is often better across all Transformer models). As can be seen, our method improves the k-NN accuracy across different settings ($k = 1, k = 10$) and Transformers, for all datasets, with the only exception being ViT AugReg small with $k = 1$ on the X-ray Pneumonia detection dataset. For challenging datasets where the models achieve under 50% such as the iNat dataset, which contains $10k$ classes, our method improves over the original models by a very significant margin ($+5.37\%$ for $k = 1$, $+5.28\%$ for $k = 10$, averaged across all 7 models).

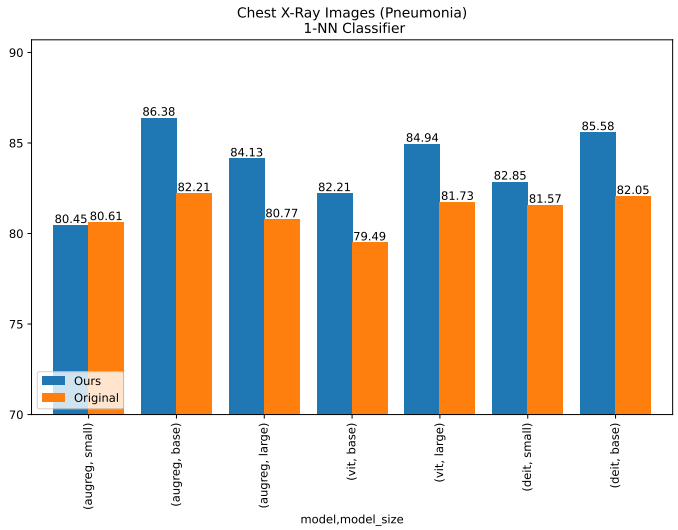


(a)

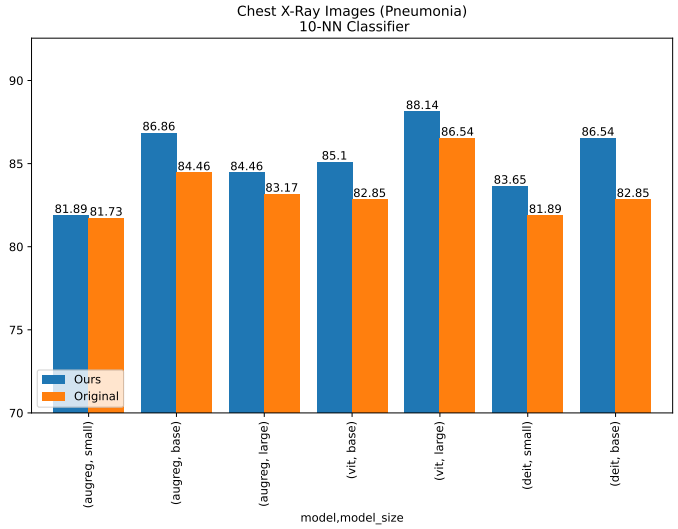


(b)

Figure 17: k-NN evaluation of performance on the iNat2021 mini dataset [28] before and after applying our method. (a) $k = 1$, (b) $k = 10$. Our method improves all models significantly.

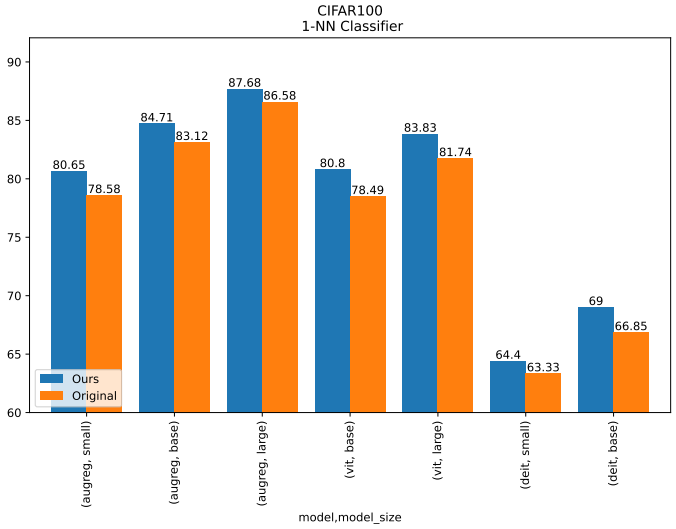


(a)

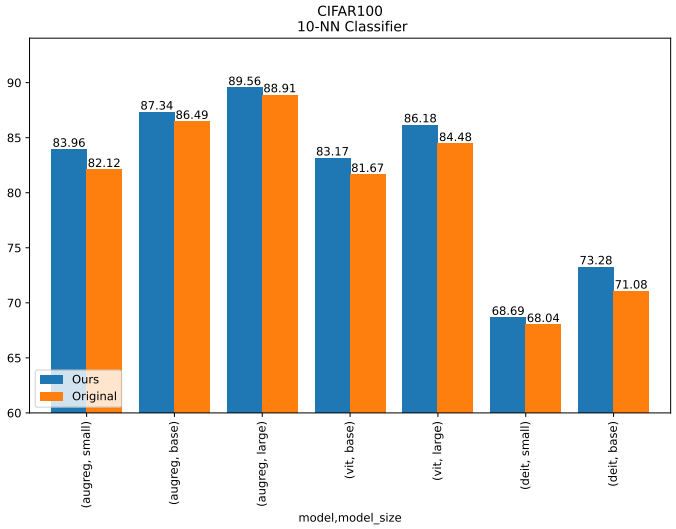


(b)

Figure 18: k-NN evaluation of performance on the X-ray Pneumonia dataset [30] before and after applying our method. (a) $k = 1$, (b) $k = 10$. Our method improves all models, other than the small variant of AugReg with $k = 1$. Note that the range of the y-axis does not start at zero.



(a)



(b)

Figure 19: k-NN evaluation of performance on the CIFAR-100 dataset [32] before and after applying our method. (a) $k = 1$, (b) $k = 10$. Our method improves all models across all settings. Note that the range of the y-axis does not start at zero.