Appendix

A. Details of training

For training, we divided the original ImageNet training set into training and validation set and used the original validation set for testing, because the labels of the original test set are not available.

We used the reference scripts published by Torchvision[1] for training. We used the specified recipes for training the models and changed only the parameter regarding augmenting images to get the models trained with different strategies. We decided to keep these basic augmentations to obtain models with similar performance. Resized cropping could be interpreted as translating and scaling the image. However, our results do not indicate a difference between these two methods and the rest, see Sec 3.

The "ResNet50 full aug" model was trained with the following command:

```
torchrun --nproc_per_node=8 train.py \
--model resnet50 \
--batch-size 128 \
--lr 0.5 \
--lr-scheduler cosineannealinglr \
--lr-warmup-epochs 5 \
--lr-warmup-method linear \
--auto-augment ta_wide \
--epochs 600 \
--random-erase 0.1 \
--weight-decay 0.00002 \
--norm-weight-decay 0.0 \
--label-smoothing 0.1 \
--mixup-alpha 0.2 \
--cutmix-alpha 1.0 \
--train-crop-size 176 \
--model-ema \
--val-resize-size 232 \
--ra-sampler \
--ra-reps=4
```

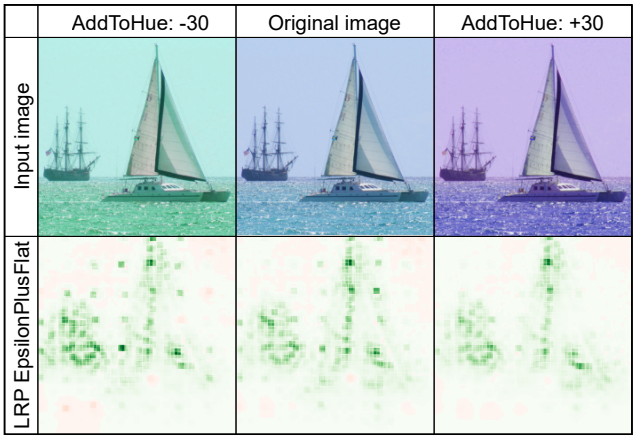
The training of "ResNet50 lim aug" is the same but without `--auto-augment ta_wide`. Table 1 shows the number of parameters, training time and accuracies of all the models.

B. Augmentation methods

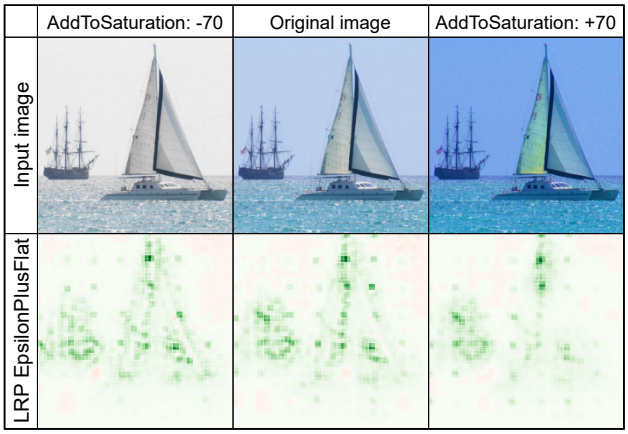
Table 2 shows the intervals of augmentation parameters used in the evaluation of "ResNet50 full aug". Figure 1 shows examples of the figures and explanations for the extreme points of these intervals.

[1] https://github.com/pytorch/vision/tree/main/references/classification

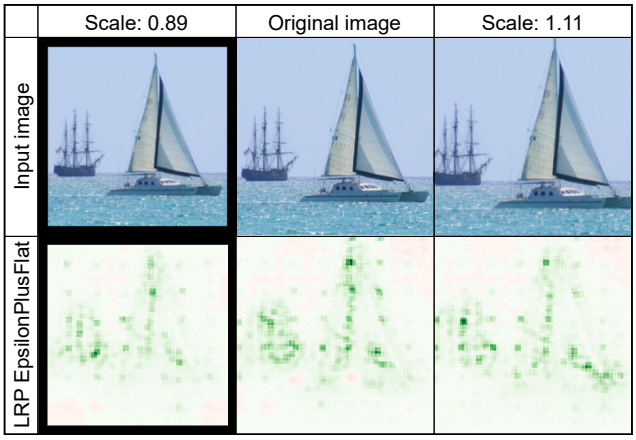
Figure 1. Examples of the augmented images and their explanations.



(a) Hue



(b) Saturation



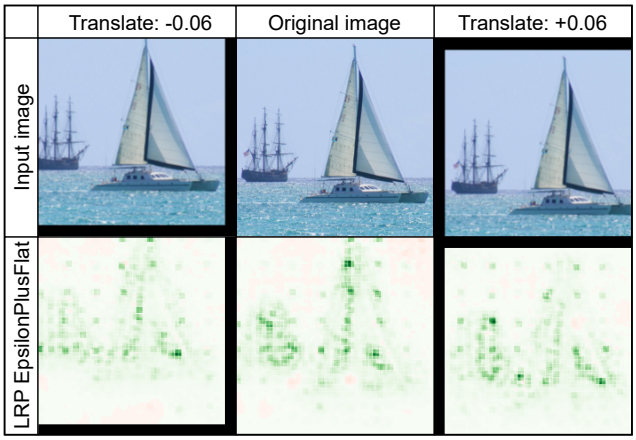
(c) Scale

Model	N parameters	Time (h)	Top-1	Reported Top-1	Top-5	Reported Top-5
ResNet50 full aug	25.6M	39	80.28	80.86	95.15	95.43
ResNet50 lim aug	-	32	79.89	-	94.97	-
EfficientNetV2 small full aug	21M	102	80.99	84.23	95.16	96.88
EfficientNetV2 small lim aug	-	101	80.89	-	95.22	-
VGG16 bn full aug	-	16	73.43	-	91.39	-
VGG16 bn lim aug	138.4M	16	73.49	73.36	91.61	91.52

Table 1. Number of parameters, training time and accuracies of all the models. The models were evaluated on ImageNet validation set. Numbers of parameters and reported accuracies were copied from https://pytorch.org/vision/stable/models.html#table-of-all-available-classification-weights. Models marked as "lim aug" were trained only with random resized cropping, horizontal flipping and, in case of EfficientNet V2 S and ResNet50, with random erasing [2]. Models marked as "full aug" were, on top of that, trained with data augmentation Trivial Augmentation [1]. All models were trained on 8 GPUs.

Name of ImgAug function	Interval
AddToBrightness	$[-95, 95]$
AddToHue	$[-30, 30]$
AddToSaturation	$[-70, 70]$
Rotate	$[-18, 18]$
Scale	$[0.89, 1.11]$
Translate	$[-0.06, 0.06]$

Table 2. Augmentation methods and intervals of magnitudes of these augmentations determined by the drop in probability by 10% for "ResNet50 full aug". The first three methods are invariant, the last three are equivariant.

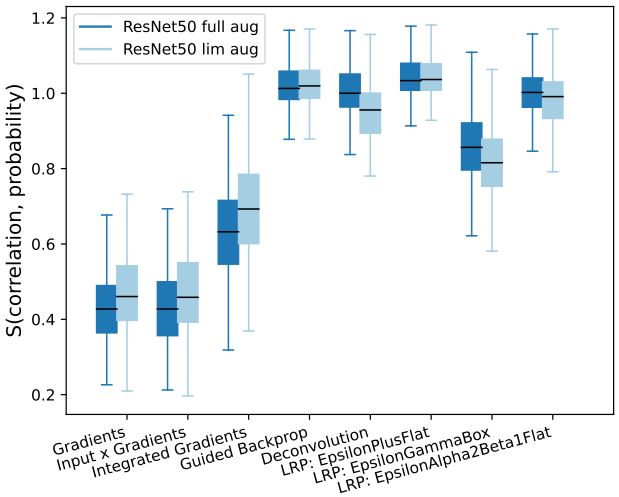


(d) Translate

C. Comparison of models trained with full and limited augmentations

Figure 2 shows the comparisons in correlation and Fig. 3 top-1000 intersection for "ResNet50 full aug" and "ResNet50 lim aug".

Figure 2. Comparison of ResNet50 trained with full ("full aug") and limited ("lim aug") data augmentation for each explainability method. We plot S(corelation, probability) for different perturbations. Boxes show the quartiles and medians, and whiskers extend to the most extreme, non-outlier data points.)

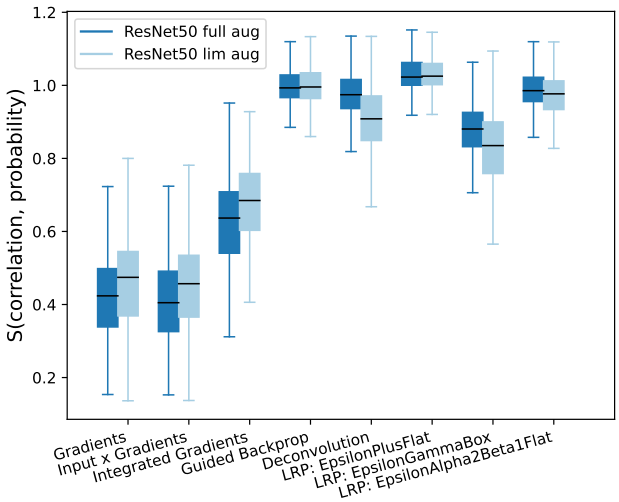


(a) AddToHue, $[-30, 30]$

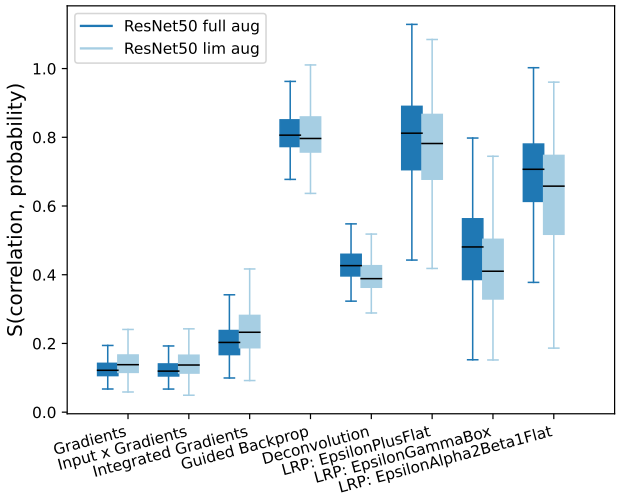
References

[1] Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. page 754–762. IEEE Computer Society, Oct 2021. 2

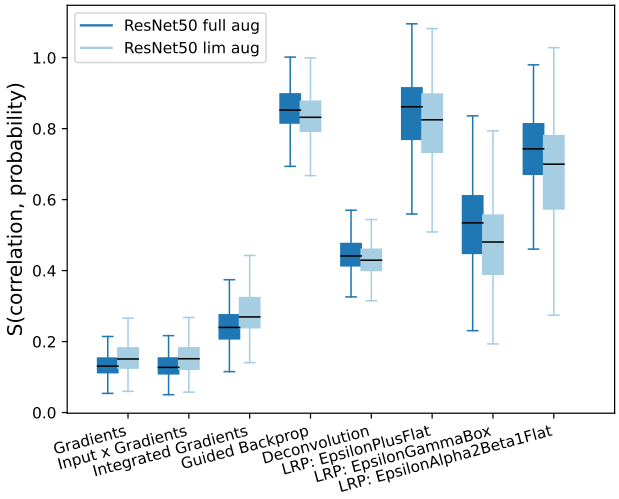
[2] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 2



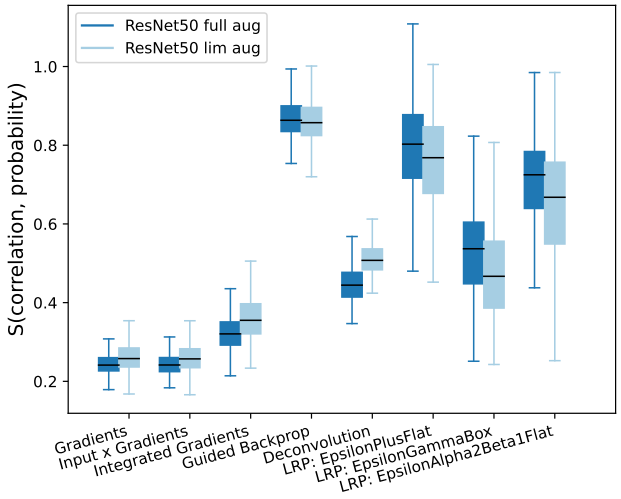
(b) AddToSaturation, $[-70, 70]$



(c) Rotate, $[-18, 18]$

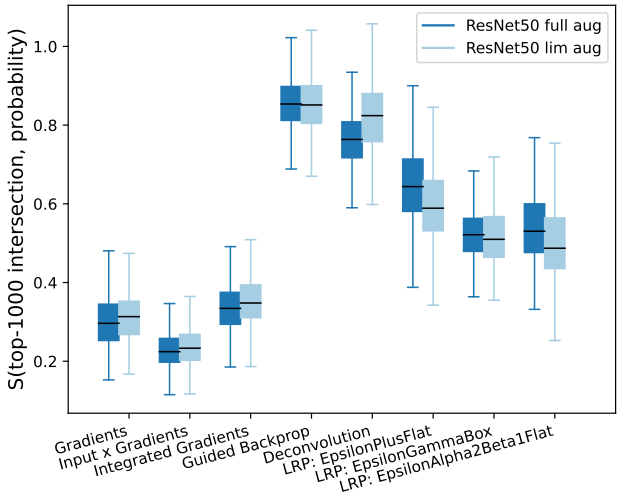


(d) Scale, $[0.89, 1.11]$

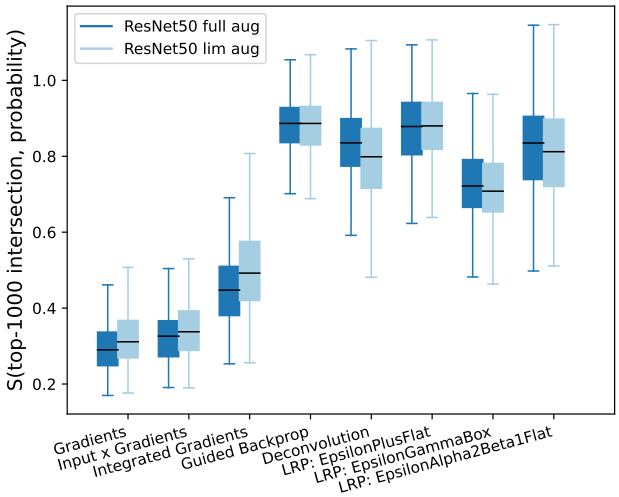


(e) Translate, $[-0.06, 0.06]$

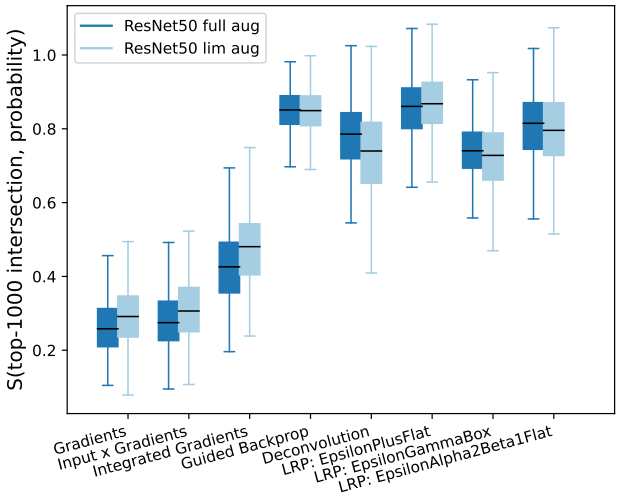
Figure 3. Comparison of ResNet50 trained with full ("full aug") and limiter ("lim aug") data augmentation for each explainability method. We plot S(top-1000, probability) for different perturbations. Boxes show the quartiles and medians, and whiskers extend to the most extreme, non-outlier data points.)



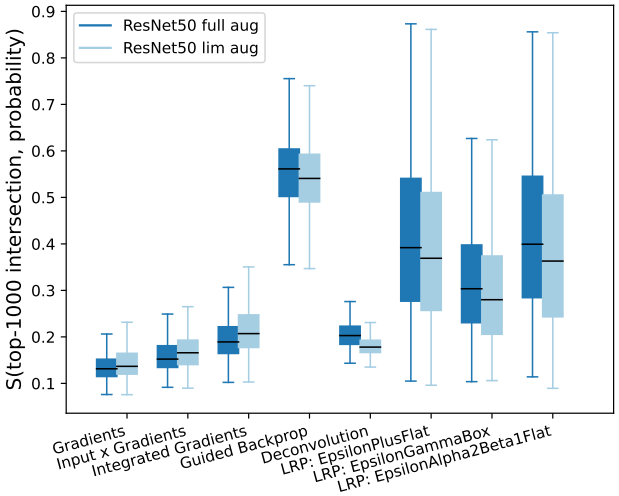
(a) AddToBrightness, $[-95, 95]$



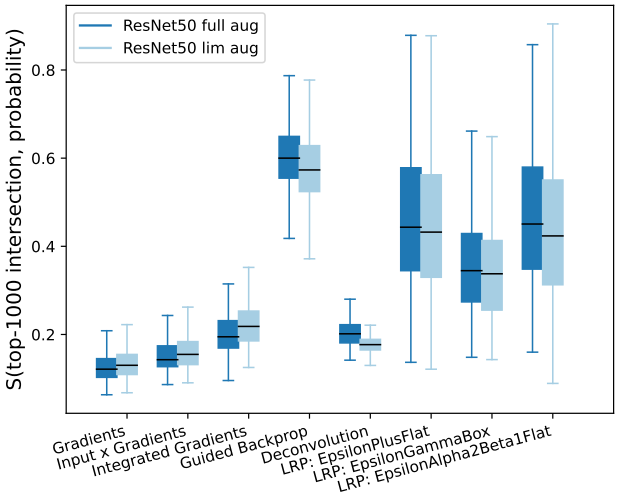
(b) AddToHue, $[-30, 30]$



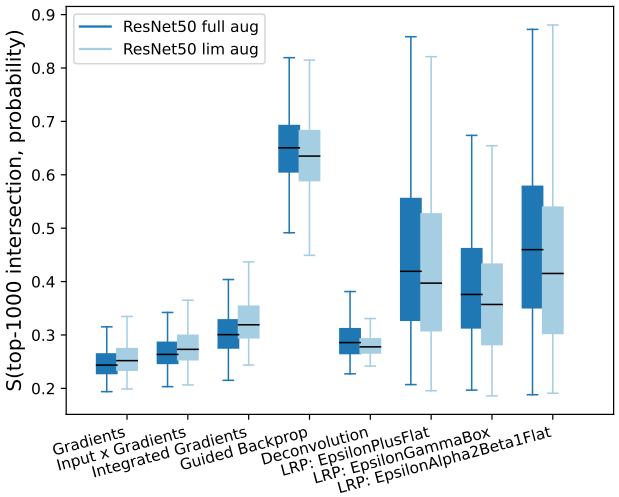
(c) AddToSaturation, $[-70, 70]$



(d) Rotate, $[-18, 18]$



(e) Scale, $[0.89, 1.11]$



(f) Translate, $[-0.06, 0.06]$