# Supplementary materials for:
# SAM: The Sensitivity of Attribution Methods to Hyperparameters

## S1. Method Description and Implementation Details

We now provide a detailed description of the interpretability methods that we have used in our proposed experiments. As described in Sec. 2, a deep learning model is a function $f$, mapping a coloured image $\boldsymbol{x}$ of spatial dimension $d \times d$ onto a softmax probability of a target class, i.e $f : \mathbb{R}^{d\times d\times 3} \rightarrow \mathbb{R}$. The model $f$ can also be represented as composition of functions i.e $f(\boldsymbol{x}) = softmax(L(x))$, where $L$ represents the logit score. An attribution method $A$, maps the model $f$, an image $\boldsymbol{x}$ and the respective set of hyperparameters $\mathcal{H}$ to an attribution map $\boldsymbol{a} \in [-1, 1]^{d\times d}$ [4], *i.e.* $A(f, \boldsymbol{x}, \mathcal{H}) = \boldsymbol{a}$. The attribution score $a_i \in [-1, 1]$ corresponding to a pixel $x_i$, is an indication of how much a pixel contributes for or against the model prediction, $f(\boldsymbol{x})$, depending on the sign of $a_i$. Most explanation methods, particularly the perturbation-based methods, inadvertently introduce their own hyperparameters, $\mathcal{H}$, but the set $\mathcal{H}$ can be empty for some explanation methods.

Now we describe different gradient and perturbation-based explanation algorithms used in our experiments.

- **Gradient** - Model gradients for a given image and a target class represent how a small change in input pixels values affects the classification score and thus, serves as a common attribution map. Mathematically, Gradient attribution map, $\boldsymbol{a}^{Grad}$, is defined as:

$$\boldsymbol{a}^{Grad} = \frac{\partial L}{\partial \boldsymbol{x}}$$

- **Gradient $\odot$ Input (GI)** - It is the Hadamard product of the input and the model gradients with respect to the input. Mathematically, GI attribution map, $\boldsymbol{a}^{GI}$, is defined as:

$$\boldsymbol{a}^{GI} = \boldsymbol{x} \odot \frac{\partial L}{\partial \boldsymbol{x}}$$

- **Integrated Gradients (IG)** - IG tackles the gradient saturation problem by averaging the gradients over $N_{IG}$ interpolated inputs derived using input and "baseline" image. Here, "baseline image" is the featureless image for which model prediction is neutral. Mathematically, IG attribution map, $\boldsymbol{a}^{IG}$ is defined as:

$$\boldsymbol{a}^{IG} = (\boldsymbol{x} - \bar{\boldsymbol{x}}) \times \int_{\alpha=0}^{1} \frac{\partial f(\bar{\boldsymbol{x}} + \alpha \times (\boldsymbol{x} - \bar{\boldsymbol{x}}))}{\partial \boldsymbol{x}} d\alpha$$

  where $\bar{\boldsymbol{x}}$ is the baseline image. In practice, the integral above is approximated as follows:

$$\boldsymbol{a}^{IG} = \frac{1}{N_T} \sum_{j=1}^{N_T} \left( (\boldsymbol{x} - \bar{\boldsymbol{x}}_{\boldsymbol{j}}) \times \int_{\alpha=0}^{1} \frac{\partial f(\bar{\boldsymbol{x}}_{\boldsymbol{j}} + \alpha \times (\boldsymbol{x} - \bar{\boldsymbol{x}}_{\boldsymbol{j}}))}{\partial \boldsymbol{x}} d\alpha \right)$$

  with $N_T$ being the number of trials.
  In our experiments, we only consider the number of trials, $N_T$, as a hyperparameter and fix the number of interpolated samples $N_{IG}$ to 100. Our PyTorch implementation of IG follows the original implementation by the authors [1].

[4] Following Adebayo et al. [8], we normalized the attribution maps of all explanation methods to the range [-1.0, 1.0] except for SP and MP. The attribution maps for SP and MP, by default, have a fixed range of [-1.0, 1.0] and [0.0, 1.0] respectively. For other explanation methods, the attribution maps were normalized by dividing the heatmaps by the maximum of their absolute values.

- **SmoothGrad (SG)** - To create smooth and potentially robust heatmaps (to input perturbations), SG averages the gradients across a large number of noisy inputs. Mathematically, SG attribution map, $\boldsymbol{a}^{SG}$, is defined as:

$$\boldsymbol{a}^{SG} = \frac{1}{N_{SG}} \sum_{n=1}^{N_{SG}} \frac{\partial L(\boldsymbol{x} + \boldsymbol{\epsilon_n})}{\partial \boldsymbol{x}}$$

where $\epsilon$ are i.i.d samples drawn from a Gaussian distribution of mean $\mu$ and std $\sigma$.
In our experiments, we consider two major hyperparameters of SG, namely the std, $\sigma$ and $N_{SG}$ samples. The mean for the i.i.d. samples were fixed to 0. Our PyTorch implementation of SG follows the original implementation by the authors [5].

- **Sliding Patch (SP)** - SP, or Occlusion as it is simply called, is one of the simplest perturbation-based methods where the authors use a gray patch to slide across the image and the change in probability is treated as an attribution value at the corresponding location. Concretely, given a binary mask, $\boldsymbol{m} \in \{0,1\}^{d \times d}$ (with 1's for the pixels in the patch and 0's otherwise), and a filler image, $\boldsymbol{z}$, a perturbed image $\bar{\boldsymbol{x}} \in \mathbb{R}^{d \times d \times 3}$ is defined as follows:

$$\bar{\boldsymbol{x}} = \boldsymbol{x} \odot (\mathbb{1}_{D \times D} - \boldsymbol{m}) + \boldsymbol{z} \odot \boldsymbol{m} \tag{1}$$

where $\boldsymbol{z}$ is a zero image or gray image[5] before input-pre-processing. Thus, the SP explanation map, $\boldsymbol{a}^{SP}$, at the pixel location $i$ is defined as:

$$a_i^{SP} = f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}^i)$$

where $\bar{\boldsymbol{x}}^i$ is the corresponding perturbed image generated by setting the patch centre at $i$. Due to computational complexity, the square patch (size $p \times p$ where $p \in \mathbb{N}$)) is slid using a stride value of $s$ greater than 1 ($s \in \mathbb{N}$), resulting in an attribution map $\boldsymbol{a}^{SP} \in \mathbb{R}^{d' \times d'}$ where $d' = \lfloor \frac{d-p}{s} + 1 \rfloor$ with $\lfloor . \rfloor$ being the greatest integer. We use bilinear upsampling to scale $\boldsymbol{a}^{SP}$ back to the full image resolution.
In our experiments, we fix the stride $s$ to be 3 and only change the patch side $p$. We implemented SP from scratch using PyTorch based on a MATLAB implementation [4].

- **LIME** - Similar to SP, it is another perturbation-based method which occludes the input image randomly. The input image is first segmented into a set of $S$ non-overlapping superpixels. Then it generates $N_{LIME}$ perturbed samples by graying out a random set of superpixels out of all the $2^S$ possible combinations, *i.e.* it generates a random superpixel mask $\boldsymbol{m'} \in \{0,1\}^S$, to mask out the image as in Eq. 1. For each perturbed sample $\bar{\boldsymbol{x}}^i$, LIME distributes the model prediction $f(\bar{\boldsymbol{x}}^i)$ among the superpixels, inversely weighted by the $L_2$ distance of $\bar{\boldsymbol{x}}^i$ from the original image $\boldsymbol{x}$. Finally, the weights of the superpixels are averaged over $N_{LIME}$ perturbed samples. The final weight $a_k$ for the $k^{\text{th}}$ superpixel is assigned to all the pixels in it, thus, resulting in LIME attribution map $\boldsymbol{a}^{LIME}$.
We use SLIC algorithm [7] for generating the superpixels and consider the number of samples, $N_{LIME}$, number of superpixels $S$ and the random seed as hyperparameters in our experiments. All the other parameters are set to their default value as given in the author's implementation [3].

- **MP** - Instead of perturbing the image with a fixed mask, MP learns the minimal continuous mask, $\boldsymbol{m} \in [0,1]^{d \times d}$, which could maximally minimize the model prediction. MP proposes the following optimization problem:

$$\boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} \lambda \left\| \boldsymbol{m} \right\|_1 + f(\bar{\boldsymbol{x}})$$

where the perturbed input, $\bar{\boldsymbol{x}}$, is given by Eq. 1 and the filler image, $\boldsymbol{z}$, is obtained by blurring $\boldsymbol{x}$ with a Gaussian blur of radius $b_R$. In order to avoid the generation of adversarial samples, MP learns a small mask of size $d'' \times d''$ which is upsampled to the original image size, $d \times d$, in every optimization step. To learn a robust and smooth mask, the authors further change the objective function as follows:

$$\boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} \lambda_1 \left\| \boldsymbol{m} \right\|_1 + \lambda_2 TV(\boldsymbol{m}) + E_{\tau \sim \mathcal{U}(0,a)} f(\Phi(\bar{\boldsymbol{x}}, \tau))$$

[5] In the ImageNet dataset, the mean pixel value is $(0.485, 0.456, 0.406)$.

where $TV(\boldsymbol{m})$ is the TV-norm used to obtain a smooth mask. The third term is the expectation over randomly jittered samples. The jitter operator $\Phi(.)$ translates the perturbed sample by $\tau$ pixels in both horizontal and vertical direction, where $\tau$ is uniformly sampled from the range $[0, a]$ with $a \in \mathbb{R}$. In practice, the above equation is implemented by gradient-descent for a number of iterations $N_{iter}$

Notably, MP introduces many hyperparameters and the model explanation map, $\boldsymbol{a}^{MP} = \boldsymbol{m}$, learnt by MP is entangled with these hyperparameters. We perform sensitivity experiments with various setting of iterations $N_{iter}$, Gaussian blur radius $b_R$, and the random seed for mask initialization. Our MP implementation in PyTorch is based on the Caffe implementation given by the authors [6].

## S2. Adversarial training

Madry et al. [33] proposed training robust classifiers using adversarial training. Engstrom et al. [20] adversarially trained a ResNet-50 model using Projected Gradient Descent (PGD) [33] attack with a normalized step size. We followed [20] and trained robust GoogLeNet model, denoted as GoogLeNet-R, for our sensitivity experiments. We used adversarial perturbation in $l_2$-norm for generating adversarial samples during training. Additionally, we used $\epsilon = 3$, a step size of $0.5$ and the number of steps as $7$ for PGD. The model was trained end-to-end for $90$ epochs using a batch-size of $256$ on $4$ Tesla-V100 GPU's. We used SGD optimizer with a learning rate ($lr$) scheduler starting with $lr = 0.1$ and dropping the learning rate by $10$ after every $30$ epochs. The standard accuracy for off-the-shelf GoogLeNet model [39] on 50k ImageNet validation dataset was $68.862\%$. Our adversarially trained GoogLeNet-R achieved an accuracy of $50.938\%$ on the same 50k images.

## S3. Similarity between IG heatmaps for regular classifiers and GI heatmaps for robust classifiers

IG generates a smooth attribution map by averaging gradients over a large collection of interpolated inputs. Intuitively, both IG and GI are computed using the element-wise product of an input and its respective gradient. Hence, similar to Sec. 4.2, we evaluate the similarity between the IG of regular models with the GI of robust models.

**Experiment** For each image, we generated IG explanations for regular models by sweeping across the number of trials $N_T \in \{0, 10, 50, 100\}$. Here, $N_T = 0$ represents vanilla GI. We computed the similarity between each IG heatmap of a regular model (*e.g.* ResNet) and the vanilla GI of their robust counterparts (*e.g.* ResNet-R).

**Results** We observed that, on increasing the $N_T$, the IG becomes increasingly similar to the GI of the robust model (Fig. S1). The same trend holds for the average similarity scores across the $1735$ images for both GoogLeNet and ResNet (Fig. S2). Similar to Sec. 4.2, the observed similarity scores give a false sense of assurance to the end-users about the model robustness.

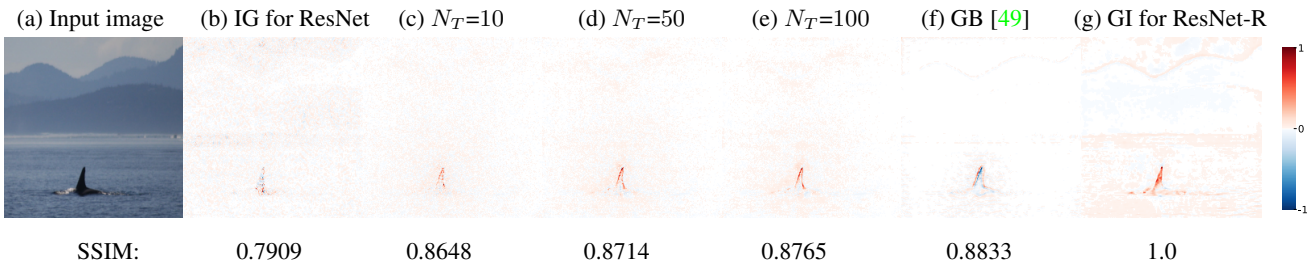
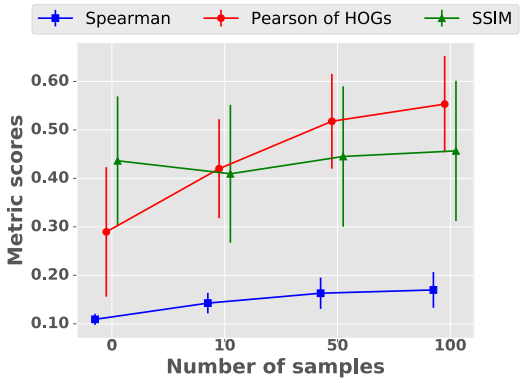
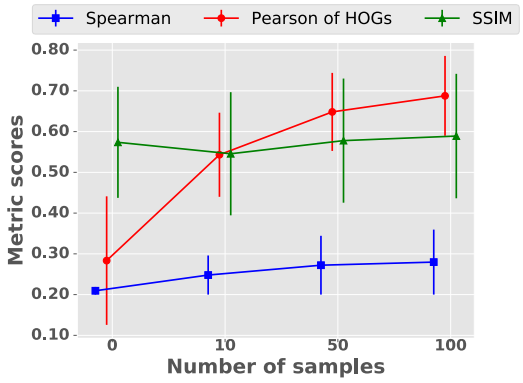



Figure S1: The Integrated Gradient (IG) [51] explanations (c–e) for a prediction of ResNet are turning into the explanation of a different prediction of a different classifier *i.e.* ResNet-R as we increase $N_T$—the hyperparameter that governs the smoothness of IG explanations. Similarly, under GuidedBackprop (GB) [49], the explanation appears substantially closer to that of a different model (f vs. g) compared the original heatmaps (f vs. b). Below each heatmap is the SSIM similarity score between that heatmap and the heatmap in (g).

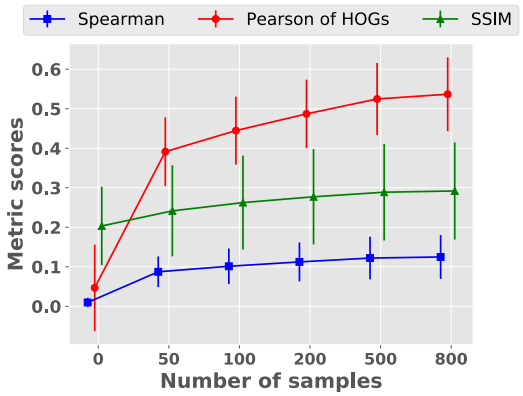



(a) GoogLeNet vs. GoogLeNet-R

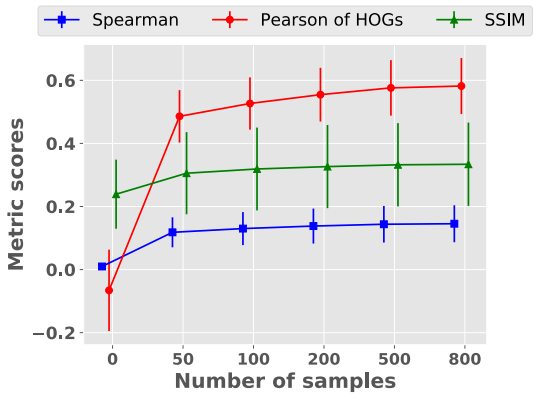



(b) ResNet vs. ResNet-R

Figure S2: Error plots showing the similarity between the Gradient $\odot$ Input [46] of a robust model (GoogLeNet-R or ResNet-R) and the Integrated Gradient [51] of the respective regular model (GoogLeNet or ResNet) across all metrics as we increase $\boldsymbol{N_T}$ — a hyperparameter that governs the smoothness of IG explanations. Here, $N_T = 0$ represents the GI of the regular model. The scores represent the average similarity scores across $1,735$ images.




(a) GoogLeNet vs. GoogLeNet-R




(b) ResNet vs. ResNet-R

Figure S3: Error plots showing the similarity between the gradients of a robust model (GoogLeNet-R or ResNet-R) and the SmoothGrad heatmaps [48] of the respective regular model (GoogLeNet or ResNet) across all metrics as we increase $\boldsymbol{N_{SG}}$ — a hyperparameter that governs the smoothness of SG explanations. Here, $N_{SG} = 0$ represents the gradient of the regular model. The scores are the mean similarity scores taken over $1,735$ images.

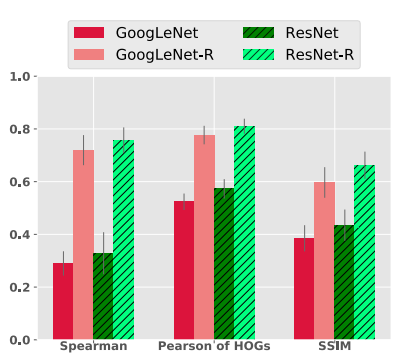
## S4. Additional sensitivity experiments

### S4.1. SmoothGrad sensitivity to the std of Gaussian noise

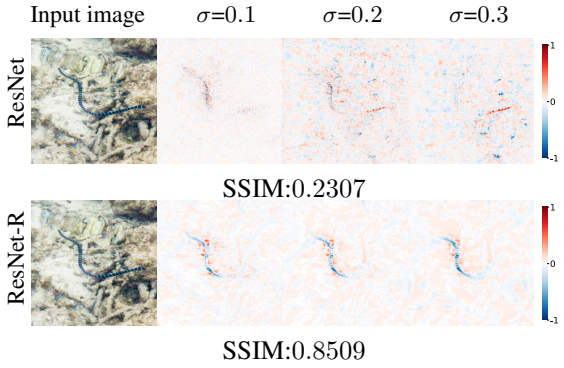
SmoothGrad (SG) generates the attribution map by averaging the gradients from a number of noisy images. The std of Gaussian noise $\sigma$ is a heuristically chosen parameter which, ideally, should not change the resultant attribution map. On the contrary, we found that changing $\sigma$ causes a large variation in the SG attribution maps.

**Experiment** To test the sensitivity to the std of Gaussian noise, we measure the average similarity between a reference heatmap at $\sigma = 0.2$ and each of the heatmaps generated by sweeping across $\sigma \in \{0.1, 0.3\}$ on the same input image. Other than the aforementioned changes, we used all default hyperparameters as in [48].

**Results** We found that the SG attribution maps of regular models are more sensitive as compared to that of robust models (Fig. S4b). Quantitatively, high sensitivity was observed in the average similarity scores across the dataset (Fig. S4a). Notably, the average Spearman correlation score, across the dataset, for GoogLeNet-R is $2.5\times$ than that of GoogLeNet.




(a) Average robustness across the dataset when changing std of Gaussian noise $\sigma$.



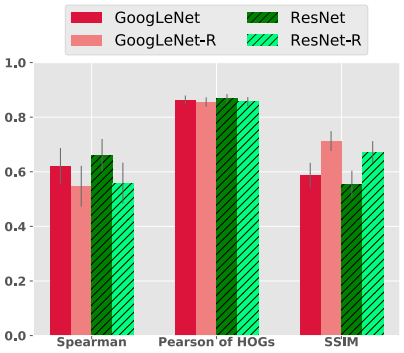

(b) SG heatmaps for ResNet-R are more consistent compared to those for ResNet.

Figure S4: Quantitative (a) and qualitative (b) figures showing the sensitivity of SmoothGrad (SG) [48] attribution maps when the **std of the Gaussian noise ($\sigma$)** changes.
**Left panel:** Compared to regular models (GoogLeNet and ResNet), heatmaps generated for robust models (GoogLeNet-R and ResNet-R) are substantially more consistent to when the Gaussian std hyperparameter changes (a).
**Right panel:** Across the dataset, the reference image caused the largest difference between the SSIM scores of ResNet heatmaps vs. ResNet-R heatmaps (b). As $\sigma$ increases, the attribution maps of ResNet become noisier while ResNet-R heatmaps become smoother (b; row 1).

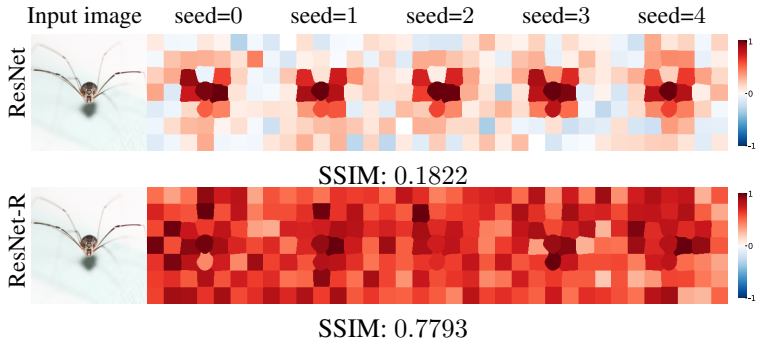
## S4.2. LIME sensitivity to changes in the random seed and number of perturbed samples

The most common hyperparameter setting for LIME is the random seed for sampling different superpixel combinations. We quantify the sensitivity of LIME across different random seeds as one can expect a minimum change in the output attribution map on changing the algorithm seed.
**Experiment** To test the sensitivity to random seed, we measure the average similarity between a reference heatmap at $seed = 0$ and each of the heatmaps generated by sweeping across $seed \in \{1, 2, 3, 4\}$ on the same input image. Notably, the number of intermediate samples for the linear regression fitting in LIME is an important factor for the resultant heatmap. Hence, we also quantify the sensitivity of LIME across the number of perturbed images, *i.e.* $N_{LIME} \in \{500, 1000\}$, to generate two heatmaps and calculate the average similarity metric scores between them.
**Results** We did not observe any significant difference between similarity scores of robust and regular models across both experiments (Fig. S5a, S6a). Note that the robust models were adversarially-trained on pixel-wise noise whereas, LIME operates at the superpixel level. We hypothesize this to be a reason for insignificant differences found between robust vs. regular models when changing the random seed. The previous experiments were performed at the number of superpixels $S = 50$. Additionally, we also repeated the same experiments at 150 superpixels but observed no significant improvement in the robustness of robust models (data not shown).

Strikingly, the Pearson correlation value for HOG features are high in both experiments (Fig. S5a & Fig. S6a). An explanation for that is because the SLIC superpixel segmentation step of LIME imposes a strong structural bias in LIME attribution maps.

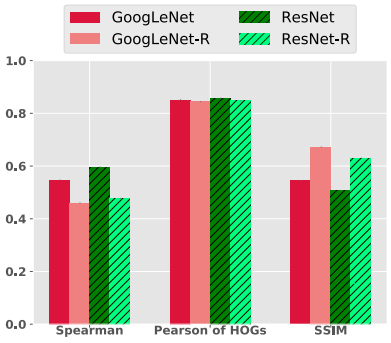



(a) Average similarity scores across the dataset when changing random seed.

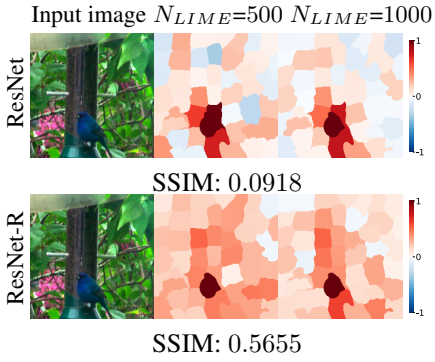



(b) LIME heatmaps for ResNet-R are more consistent (under SSIM similarity score) compared to those for ResNet.

Figure S5: Quantitative (a) and qualitative (b) figures showing the sensitivity of LIME attribution maps when **the random seed** of LIME (which governs the random selection of LIME superpixel masks) changes.
**Left panel:** For both regular and robust models, LIME attribution maps are similarly sensitive to the random seed (similarity scores well below 1.0). The high Pearson of HOGs scores are hypothesized to be because the SLIC superpixel segmentation imposes a consistent visual structure bias across LIME attribution maps (before and after the random seed changes). Under SSIM, LIME heatmaps of robust models are more consistent than those of regular models.
**Right panel:** Across the dataset, the reference image causes the largest difference between the SSIM scores of ResNet heatmaps and those of ResNet-R heatmaps (b; top row vs. bottom row).




(a) Average robustness across the dataset when changing $N_{LIME}$.




(b) LIME heatmaps for ResNet-R are more consistent compared to those for ResNet.

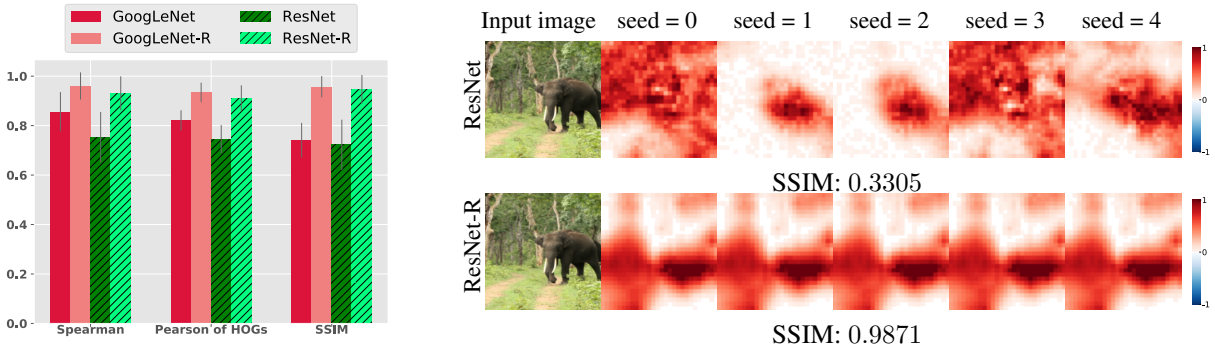
Figure S6: Quantitative (a) and qualitative (b) figures showing the sensitivity of LIME attribution maps when **the number of perturbed samples $N_{LIME}$** changes.
**Left panel**: Both robust and regular models are similarly sensitive to the $N_{LIME}$ under Pearson correlation of HOGs while the heatmaps for robust models are more consistent under SSIM (a).
**Right panel**: Across the dataset, the reference input image causes the largest difference between the SSIM scores of ResNet heatmaps vs. the SIM scores of ResNet-R heatmaps (b; top row vs. bottom row).

### S4.3. Meaningful Perturbation sensitivity to changes in the random seed

For MP mask optimization, Fong et al. [22] used a circular mask initialization that suppresses the score of the target class by 99% when compared to that of using a completely blurred image. We argue that this circular mask acts as a strong bias towards ImageNet images (*i.e.* they may not work for other datasets) since ImageNet mostly contains object-centric images. Hence, we evaluate the sensitivity of MP attribution maps by initializing masks with different random seeds (corresponding to different mask initializations).
**Experiment** Similar to Sec. S4.2, we calculate the average pairwise similarity between a reference heatmap using $seed = 0$ and each of the heatmaps generated by sweeping across $seed \in \{1, 2, 3, 4\}$ on the same input image. All the other

hyperparameters are the same as in [22].

**Results** We found that robust models are less sensitive to random initialization of masks (Fig. S7b). The average similarity scores for robust models are consistently higher than their regular counterparts (Fig. S7a).

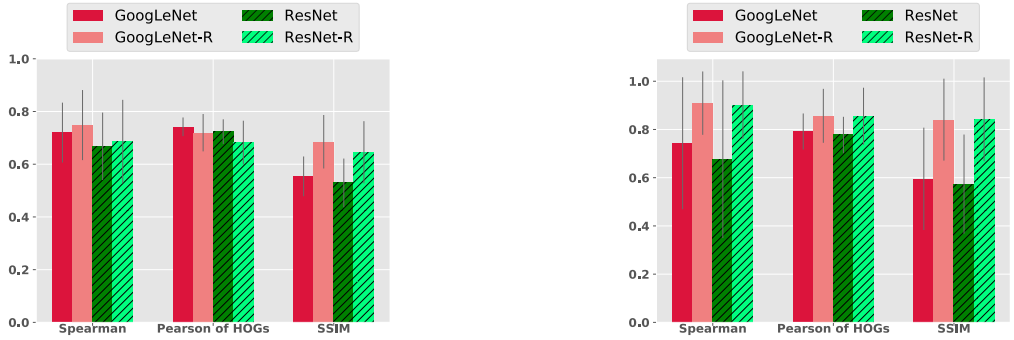



(a) Average robustness across the dataset when changing random seed.

(b) MP heatmaps for ResNet-R are more consistent compared to those for ResNet.

Figure S7: Robust classifiers cause heatmaps to be more consistent (*i.e.* higher SSIM scores) when **the random seed** changes, both quantitatively (a) and qualitatively (b).

**Right panel:** Across the dataset, the reference image causes the largest difference between the SSIM scores of ResNet heatmaps vs. ResNet-R heatmaps (b; top row vs. bottom row).




(a) Average similarity of heatmaps across the dataset under three metrics when the blur radius $b_R$ changes.

(b) Average similarity of heatmaps across the dataset under three metrics when the number of iterations $N_{iter}$ changes.

Figure S8: Error bar plots showing the similarity of **Meaningful-Perturbation (MP)** attribution maps when a hyperparameter—here **Gaussian blur radius $b_R$** (a), and **the number of iterations $N_{iter}$** (b)—changes. These figures represent the quantitative results for the experiments in Sec. 4.3.2.

**Left panel**: Changing $b_R$ caused the heatmaps for regular classifiers (GoogLeNet and ResNet) to vary more, under Spearman rank correlation and SSIM, than those for robust classifiers (see Figs. 4a & S16 for qualitative results).

**Right panel**: Heatmaps generated for regular models (dark red & dark green) are consistently more variable than those generated for robust models (light red & light green) across all metrics (b).

| Algorithm | Models | SSIM | Localization Error | Insertion | Deletion |
|---|---|---|---|---|---|
| **SG** | **GoogLeNet** | 0.6422±0.3197 | 0.2744±0.1382 | 0.1627±0.0386 | 0.2091±0.0453 |
| | **GoogLeNet-R** | 0.9648±0.0051 | 0.2798±0.0539 | 0.2146±0.0085 | 0.2433±0.0090 |
| | **ResNet** | 0.7854±0.0238 | 0.2632±0.1140 | 0.2012±0.0388 | 0.2342±0.0447 |
| | **ResNet-R** | 0.9780±0.0034 | 0.2566±0.0611 | 0.2745±0.0089 | 0.3054±0.0095 |
| **SP-S** | **GoogLeNet** | 0.9221±0.0321 | 0.3524±0.0926 | 0.5056±0.0208 | 0.1616±0.0132 |
| | **GoogLeNet-R** | 0.9894±0.0069 | 0.3468±0.0424 | 0.4281±0.0082 | 0.1260±0.0039 |
| | **ResNet** | 0.9633±0.0188 | 0.4649±0.1182 | 0.5959±0.0226 | 0.2581±0.0173 |
| | **ResNet-R** | 0.9891±0.0073 | 0.3666±0.0660 | 0.4699±0.0075 | 0.1459±0.0041 |
| **SP-L** | **GoogLeNet** | 0.6210±0.1021 | 0.3390±0.2194 | 0.4078±0.1354 | 0.1456±0.0595 |
| | **GoogLeNet-R** | 0.6540±0.1361 | 0.3344±0.1729 | 0.4130±0.0817 | 0.1265±0.0434 |
| | **ResNet** | 0.8239±0.0718 | 0.4158±0.2827 | 0.4846±0.1493 | 0.2344±0.0885 |
| | **ResNet-R** | 0.6867±0.1276 | 0.3493±0.2066 | 0.4485±0.0861 | 0.1481±0.0528 |
| **LIME** | **GoogLeNet** | 0.5862±0.0467 | 0.3260±0.1458 | 0.5844±0.0458 | 0.1352±0.0227 |
| | **GoogLeNet-R** | 0.7125±0.0363 | 0.3331±0.1030 | 0.3832±0.0432 | 0.1340±0.0220 |
| | **ResNet** | 0.5552±0.0491 | 0.2951±0.1565 | 0.7224±0.0421 | 0.1800±0.0281 |
| | **ResNet-R** | 0.6722±0.0401 | 0.3301±0.1361 | 0.4549±0.0424 | 0.1437±0.0223 |
| **MP** | **GoogLeNet** | 0.7412±0.0697 | 0.2386±0.1458 | 0.5345±0.0402 | 0.1275±0.0278 |
| | **GoogLeNet-R** | 0.9572±0.0432 | 0.2875±0.0725 | 0.4001±0.0176 | 0.1222±0.0086 |
| | **ResNet** | 0.7221±0.1019 | 0.2651±0.1892 | 0.6184±0.0556 | 0.2064±0.0524 |
| | **ResNet-R** | 0.9476±0.0572 | 0.2928±0.0941 | 0.4328±0.0226 | 0.1407±0.0121 |

Table S1: The results in this table are the numeric format of Fig. 5. Compared to regular models, robust classifiers (GoogLeNet-R and ResNet-R) are more robust in the attribution space (*i.e*. higher SSIM scores) and also more robust in the downstream accuracy space (*i.e*. smaller stds across three different accuracy metrics: Localization error, Deletion, and Insertion).

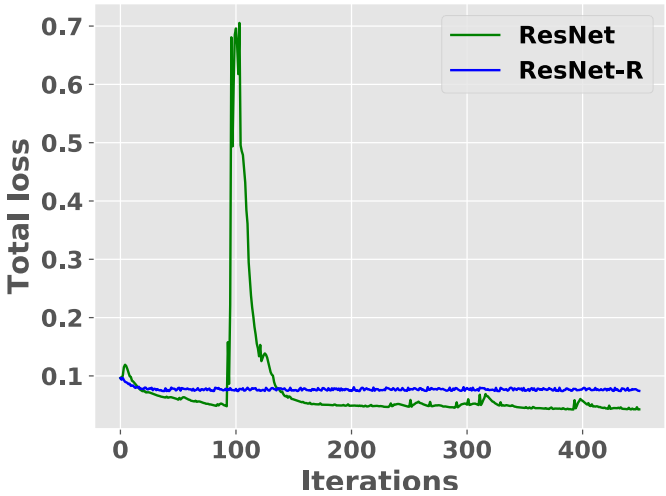
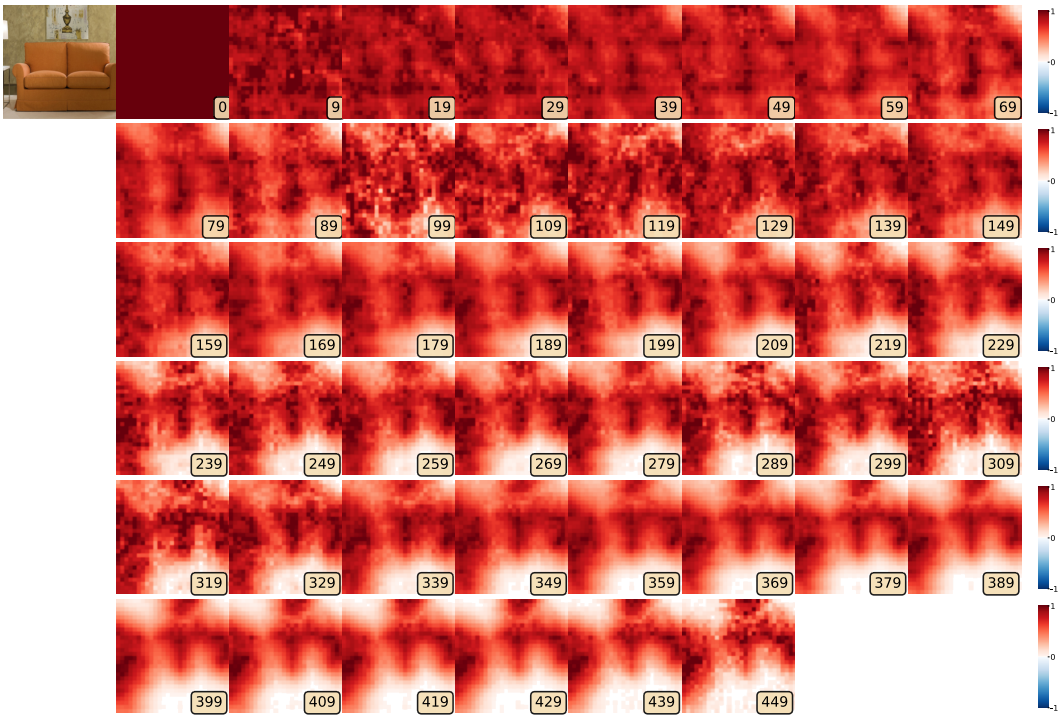
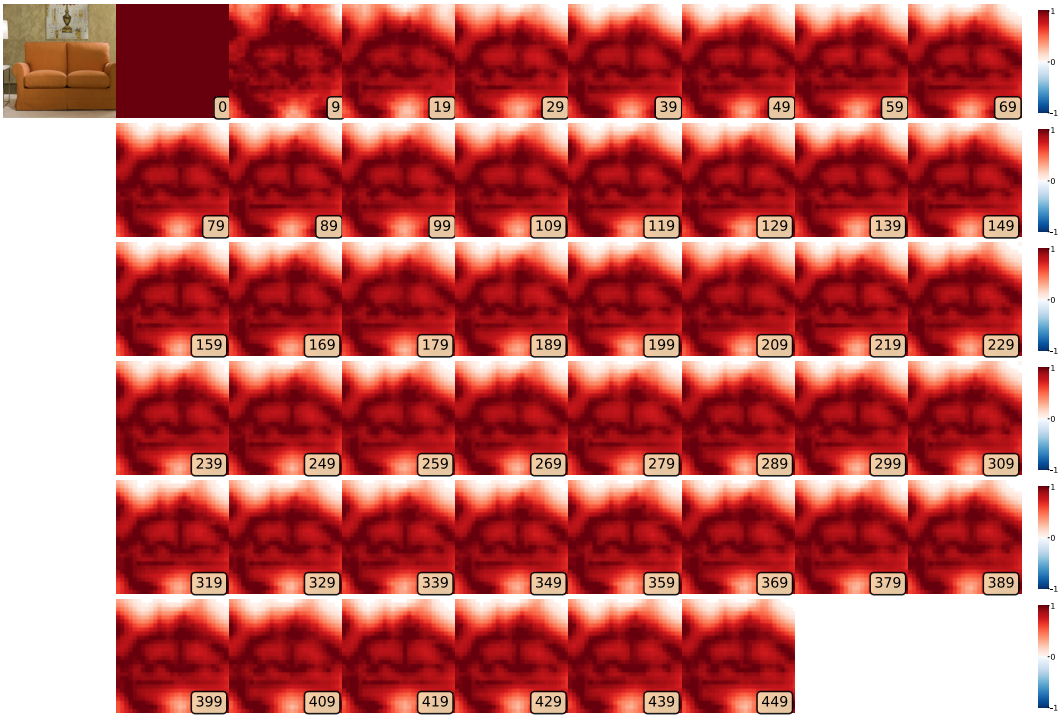



Figure S9: The total-loss plots ($L_1 + TV + softmax$) when running MP optimization algorithm (using a ResNet and ResNet-R classifier) on the reference studio couch image in Fig. 4b. The loss curve for ResNet-R converges quickly after 10 steps while MP loss curve often fluctuates (here, peaked at around step 100).

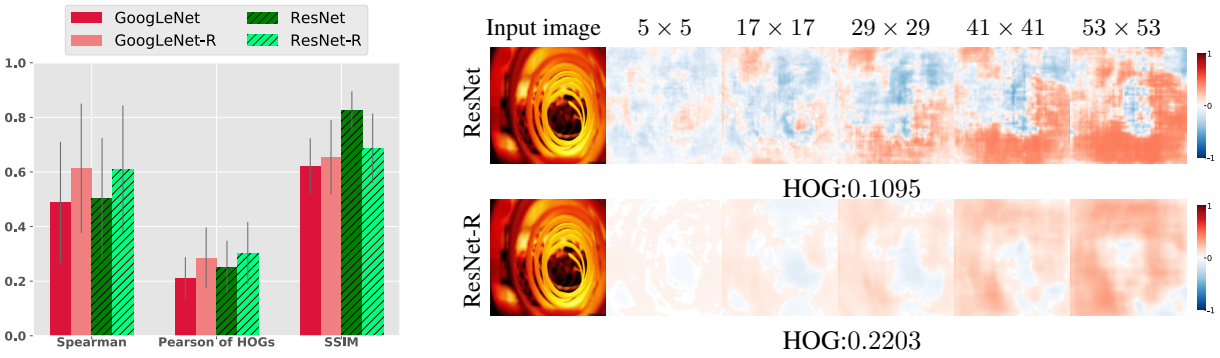



(a) ResNet




(b) ResNet-R

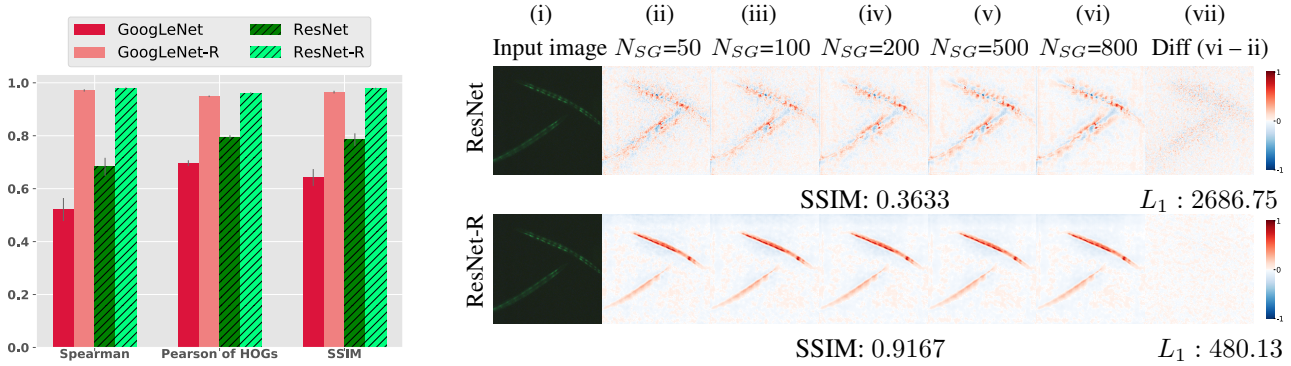
Figure S10: Evolution of attribution maps generated from a 450-step MP optimization run for a studio couch image using ResNet (a) and ResNet-R (b) models. This figure is an extension of Fig. 4b. The attribution maps for ResNet-R model (b) converges to the optimum mask in just $\sim$ 10 iterations whereas the mask in the ResNet model are very inconsistent and keep fluctuating among differernt iterations. For instance, the ResNet (a) masks becomes noisy iteration 289, 309, 319, and 449 despite being stable at 209, 379 and 409 iterations. These qualitatively heatmaps are consistent with the quantitative loss-over-iteration plots (Fig. S9) where the ResNet loss curve oscillates while the ResNet-R curve converges early.




(a) Average similarity of heatmaps across the entire dataset when the patch size changes.

(b) SP heatmaps for ResNet & ResNet-R change entirely for different patch sizes.

Figure S11: Sliding Patch (SP) attribution maps vary largely—both quantitatively (a) and qualitatively (b)—when **the patch size** changes. The stride was 3 for all cases.
**Right panel:** As the patch size increases, we observe the attribution values are higher (*i.e.* higher-intensity heatmaps), for both ResNet and ResNet-R (b).
**Left panel:** On average, across the dataset, we observe low similarity, under all three metrics, across the generated heatmaps (for both ResNet and ResNet-R) when the patch size changes (a). See Fig. S15 for more examples of this behavior.




(a) Average robustness across the dataset

(b) SG heatmaps for ResNet-R are more consistent compared to those for ResNet.

Figure S12: On average, across the dataset, SmoothGrad explanations for robust classifiers are almost perfectly consistent upon varying the sample size $N_{SG} \in \{50, 100, 200, 500, 800\}$ *i.e.* GoogLeNet-R and ResNet-R similarity scores are near 1.0 (a). However, the same heatmaps for regular classifiers are substantially more sensitive (a). We show here the input image (i) that yields the largest difference (among the dataset) between the SSIM score for ResNet-R heatmaps (0.9167) and that for ResNet heatmaps (0.3633). While SG heatmaps may appear qualitatively consistent, the pixel-wise variations (*e.g.* see column vii—the results of subtracting ii from vi) may cause issues for applications that require pixel-level precision.

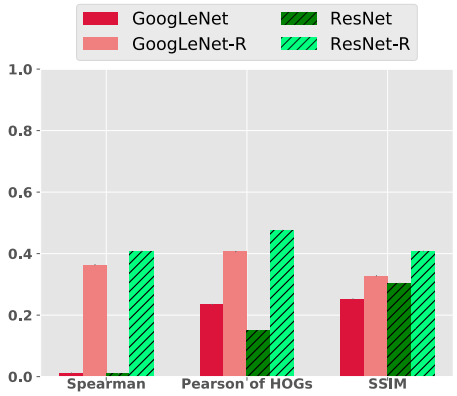



Figure S13: Compared to the gradients of regular classifiers (darker red and green), the gradients of robust classifiers (lighter red and green) are consistently more invariant before and after the addition of noise to the input image under all three similarity metrics (higher is better).


Figure S14: This figure is an extension to Fig. 2. Qualitative trend showing the increase in similarity between the attribution maps from SmoothGrad (SG) of ResNet (c—g) and vanila gradient (Grad) of ResNet-R (i) as **the number of samples $N_{SG}$ increases**. Below each heatmap is the SSIM similarity score between that heatmap and the heatmap in column (i). As the sample size $N_{SG}$ increases, SG attribution maps of ResNet become increasingly more similar, under SSIM, to the gradient heatmaps of ResNet-R, a completely different network. Additionally, by comparing column (h) and (i), one might conclude that ResNet and ResNet-R behave similarly (because the heatmaps are similar both qualitatively and quantitatively under SSIM). However, these are two completely distinct networks with different training regimes and their differences can be seen by comparing column (b) and (i). In sum, de-noising heatmaps, *e.g.* using SG or GB, may cause misinterpretation.

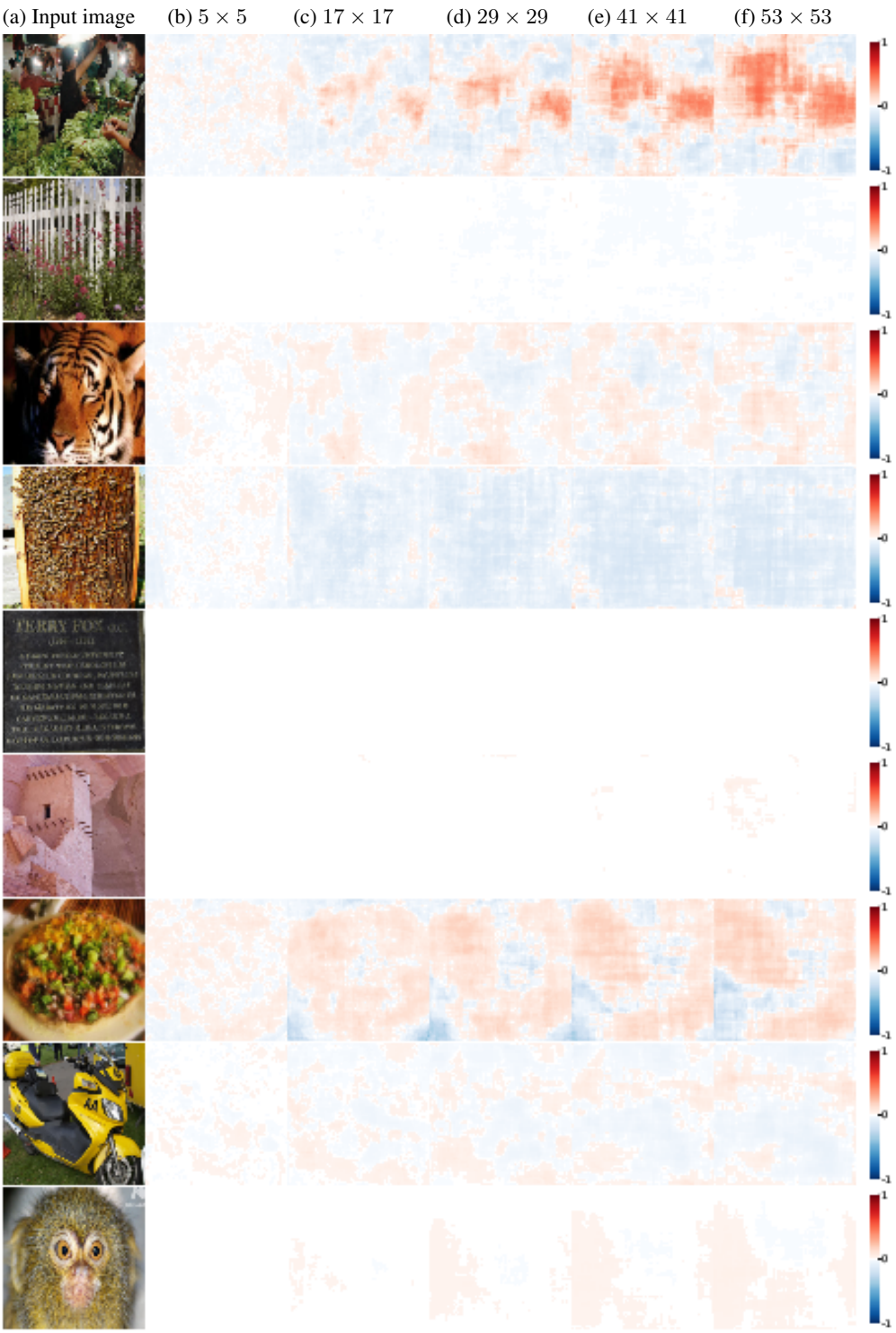
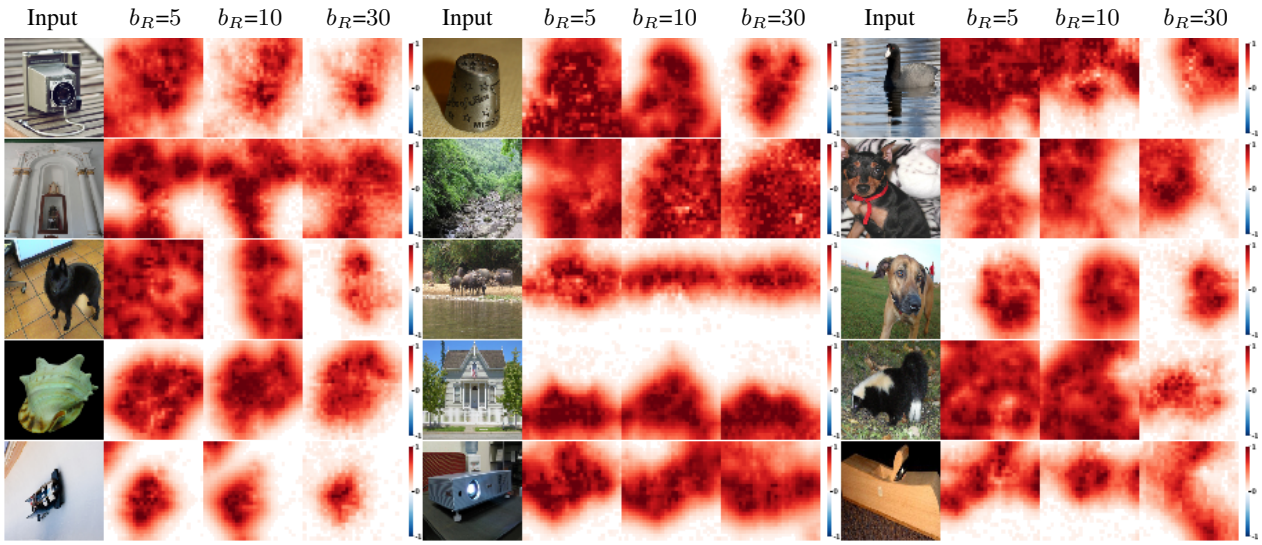
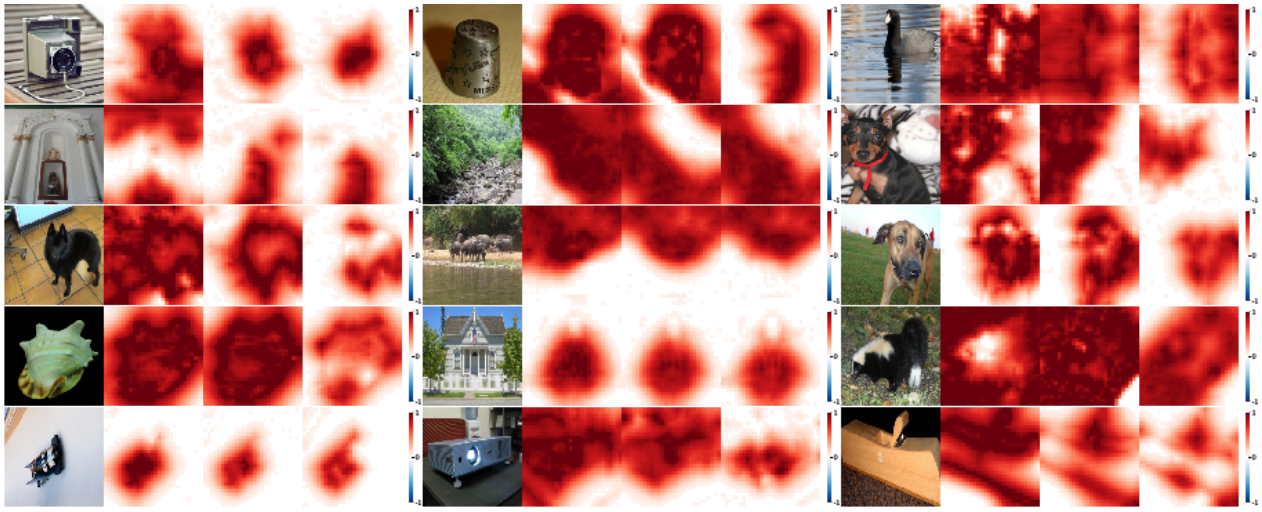



Figure S15: Sliding-Patch (SP) attribution maps are very sensitive to different **patch sizes** (Sec. 4.4.1). Across the dataset, the reference images had the lowest Pearson correlation of HOG features among the ResNet heatmaps. For some images with huge objects (*e.g.* the image of a white fence in row 2), we do not observe any significant probability drop even for a patch size of $53 \times 53$ (f) and hence the attribution values are almost zero. This observation underlines an important challenge of choosing the right patch size when using SP.

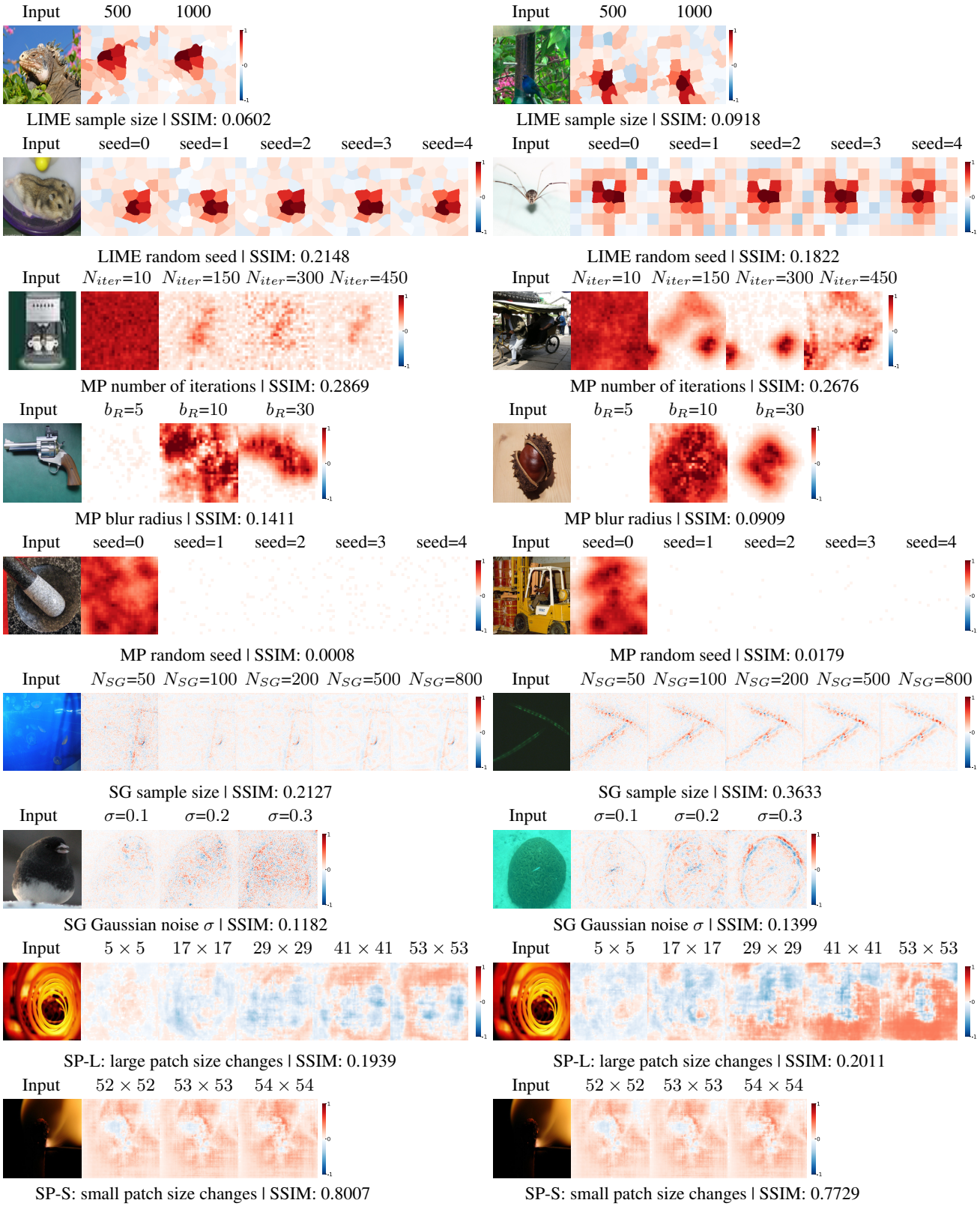



(a) ResNet



(b) ResNet-R

Figure S16: Attribution maps of ResNet (a) become more scattered as we increase the **Gaussian blur radius $b_R$** (from left to right) in the MP sensitivity experiment (Sec. 4.3.2). In contrast, for ResNet-R, the attribution maps become smoother as the blur radius increases. The reference images here were randomly chosen.




(a) GoogleNet

(b) ResNet

Figure S17: Examples where the explanations are the most inconsistent, under SSIM similarity, when a hyperparameter changes. Across the entire dataset, the reference images caused highest sensitivity (*i.e.* lowest SSIM scores) for different attribution methods and their respective hyperparameter settings for both GoogLeNet (a) and ResNet (b).