Appendix

Appendix I: Proofs

Nonlinearity ?] concludes that SM, IG, LRP,DeepLIFT are equivalent for linear models and their proof also applies to SG. We first introduce the following proposition:

Proposition 1. *All attribution methods mentioned in Sec* **??** *except GradCAM and Guided Backpropagation are equivalent if the model behaves linearly.*

Proof. As the Proposition 4 and Conclusion 6 in **?**] prove that Saliency Map, Integrated Gradient, DeepLIFT and LRP are equivalent for a linear model, we just need to prove SmoothGrad is equivalent to Saliency Map if the model is linear.

If a model behaves linearly, we can express the output score y_c for class c as a linear combination such that $y_c = \mathbf{w}_c^\top \mathbf{x} + b_c$. Then the SmoothGrad $S(\mathbf{x})_c$ is

$$\begin{aligned} S(\mathbf{x})_c &= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}\lambda)} \frac{\partial[\mathbf{w}_c^\top(\mathbf{x}+\boldsymbol{\epsilon})+b_c]}{\partial \mathbf{x}} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}\lambda)} \mathbf{w}_c \\ &= \mathbf{w}_c = \frac{\partial y_c}{\partial \mathbf{x}} \quad \text{(Saliency Map)} \end{aligned} \tag{1}$$

□

Proof to Proposition 4 *If an attribution method A satisfies both sensitivity-n_1 and sensitivity-n_2, then $N_p^k(\mathbf{x}, A) = 0$ under the condition if $\sum_i^{n_1} s_i = \sum_j^{n_2} s_j = kS(\mathbf{x}, A), s_i \in \pi_A^+(\mathbf{x}), s_j \in \hat{\pi}_A^+(\mathbf{x}), k \in [0, 1]$, but not vice versa.*

Proof. If A satisfies sensitivity-n_1, for any given ordered subset π, we have

$$\sum^{n_1} s_i = R(\mathbf{x}, \pi)$$

Same thing happens to n_2 if A satisfies sensitivity-n_2. Under the condition if $\sum_i^{n_1} s_i = \sum_j^{n_2} s_j = kS(\mathbf{x}, A), s_i \in \pi_A^+(\mathbf{x}), s_j \in \hat{\pi}_A^+(\mathbf{x}), k \in [0, 1]$,

$$\begin{aligned} N_p^k(\mathbf{x}, A) &= |R(\mathbf{x}, \pi_A(\mathbf{x}) - R(\mathbf{x}, \pi_A^+(\mathbf{x})| \\ &= |\sum_i^{n_1} s_i - \sum_j^{n_2} s_j| \\ &= |kS(\mathbf{x}, A) - kS(\mathbf{x}, A)| = 0 \end{aligned} \tag{2}$$

□

Appendix II: Implementation Details

Models

We evaluate N_Ord, S_Ord for all attribution methods mentioned in Section **??**. We evaluate on 9600 images from ImageNet [**?**] with pre-trained on VGG16[**?**].

Attribution Methods

Saliency Map
As discussed in Sec **??**, we use `grad` × `input` to represent the Saliency Map, instead of the vanilla gradient.

Integrated Gradient
We use the black image as the baseline for all images and we use the 50 samples on the linear path from the baseline to the input.

Smooth Gradient
As discussed in Sec **??**, we use `smooth_grad` × `input` to represent the Smooth Gradient. We pick a noise level of 20 % as it appears to be the best parameter in its original paper [**?**]. We randomly sample 50 points from the Gaussian distribution for the aggregation.

DeepLIFT
We use the black image as the baseline for all images and we use the RevealCancel rule for DeepLIFT [1]

LRP
We use the implementation of LRP-$\alpha 2\beta 1$ with generalization tricks mentioned by **?**] who argues this rule is better for image explanations.

Guided Backpropagation
To implement Guided Backpropagation, we modify the ReLU activation in the network to filter out the negative gradient in tensorflow.

```
@ops.RegisterGradient("GuidedBackProp")
   def _GuidedBackProp(op, grad):
      dtype = op.inputs[0].dtype
      return grad * tf.cast(grad > 0.,
          dtype) * \
            tf.cast(op.inputs[0] > 0.,
                dtype)
```

GradCAM
We use the last convolutional layer to compute the GradCAM for all images.

Appendix III

More examples of evaluating each images with TPN and TPS are shown in Fig 1

[1]We use the release code on https://github.com/kundajelab/deeplift

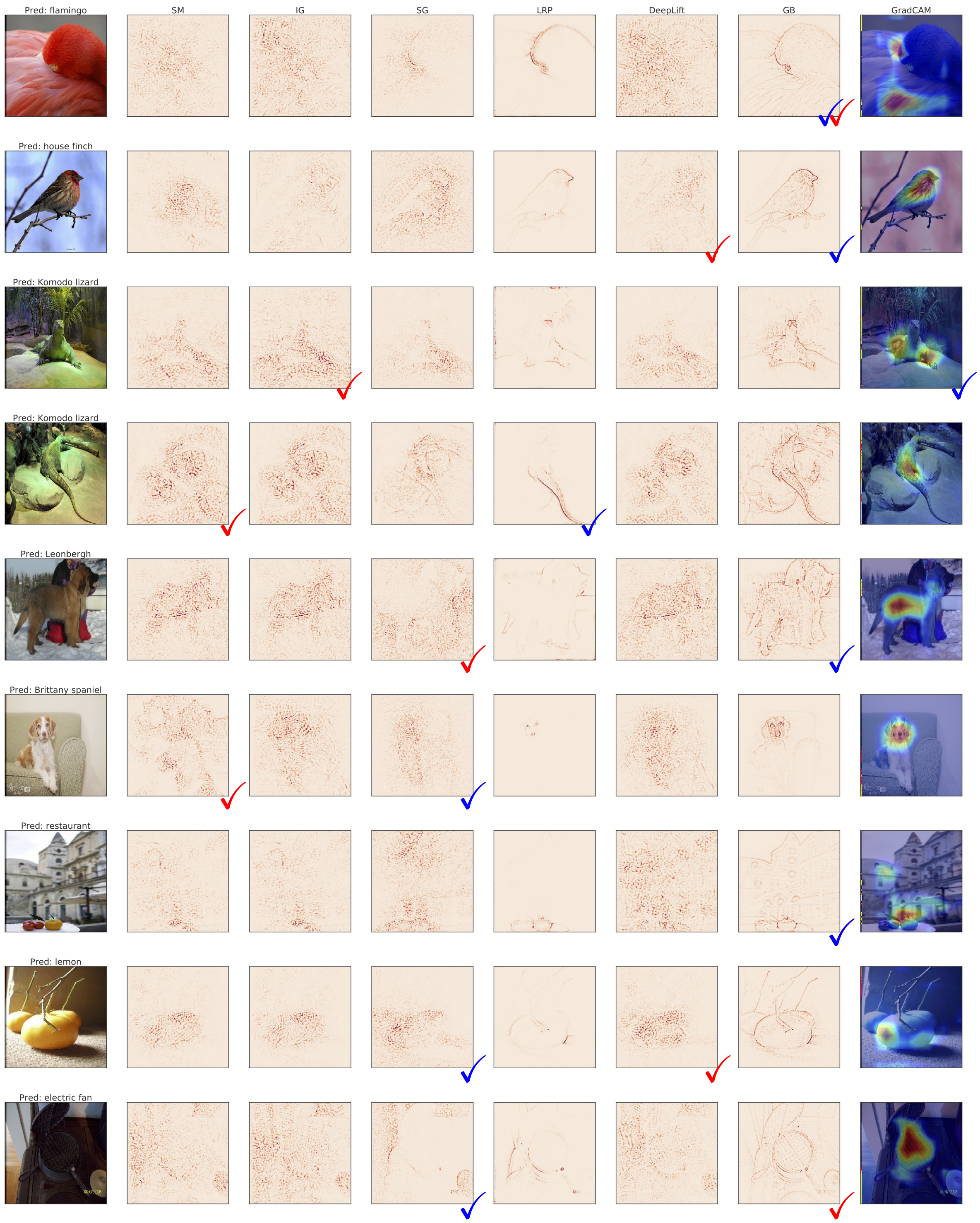

Figure 1. More visualizations of different attribution methods. Red checks mark the winner of Total Proportionality for Necessity and blue checks mark the winner of Total Proportionality for Sufficiency