Interpreting Interpretations: Organizing Attribution Methods by Criteria

Zifan Wang, Piotr Mardziel, Anupam Datta, Matt Fredrikson
Carnegie Mellon Univeristy
Moffett Field, CA, 94089
zifanw@andrew.cmu.edu

Abstract

Attribution methods that explains the behaviour of machine learning models, e.g. *Convolutional Neural Networks (CNNs), have developed into many different forms, motivated by desirable distinct, though related, criteria. Following the diversity of attribution methods, evaluation tools are in need to answer: which method is better for what purpose and why? This paper introduces a new way to decompose the evaluation for attribution methods into two criteria: ordering and proportionality. We argue that existing evaluations follow an ordering criteria roughly corresponding to either the logical concept of necessity or sufficiency. The paper further demonstrates a notion of Proportionality for Necessity and Sufficiency, a quantitative evaluation to compare existing attribution methods, as a refinement to the ordering criteria. Evaluating the performance of existing attribution methods on explaining the CNN for image classification, we conclude that some attribution methods are better in the necessity analysis and the others are better in the sufficiency analysis, but no method is always the winner on both sides.*

1. Introduction

[1] While deep learning has become the state of the art in numerous machine learning applications, deep models are especially resistant to human understanding. The gap between power and interpretability is growing especially wide in vision and audio applications where adversarial examples [17] demonstrate that models incorporate semantically meaningless features (road stop signs with human-imperceptible changes being classified as speed limit signs [10]) .

Among approaches aiding the interpretability of opaque models are input attributions which assign to each model input a level of contribution to a particular prediction. When visualized alongside inputs, an attribution gives a human interpreter some notion of what about the input is important to the outcome (see, for example, Figure 1). Being explanations of highly complex systems intended for highly complex humans, attributions have been varied in their approaches and sometimes produce distinct explanations for the same outputs.

Nevertheless, save for the earliest approaches, attribution methods distinguish themselves with one or more desirable criteria and in some instances define quantitative evaluation metrics indicating preference of one attribution method over another. Ablation-based criteria such as *Area Over Perturbation Curve*[24] and similar[7, 21] tested by interventions: an attribution should point out inputs that, when dropped or ablated while keeping all other inputs fixed, induce the greatest change in output. Alternatively, measures such as *Average % Drop* [8] instead determine to what extent important inputs stand on their own by comparing model scores relative to scores on just the important inputs (all other inputs are perturbed/ablated). Finally, scaling criteria such as *completeness* [31], *sensitivity-n* [2], *linear-agreement* [18, 31] calibrate attribution to the change in output as compared to change in input when evaluated on some baseline.

While evaluation criteria endow attributions with some limited semantics, the variations in design goals, evaluation metrics, and the underlying methods resulted in attributions failing at their primary goal: aiding in model interpretation. This work alleviates these problems and makes the following contributions.

- We decompose and organize existing attribution methods' goals along two complementary properties: ordering and proportionality. While ordering requires that an attribution should order input features according to some notion of importance, proportionality stipulates also a quantitative relationship between a model's outputs and the corresponding attributions in that particular ordering.

- We describe how all existing methods are motivated by an attribution ordering corresponding roughly to

[1] Under Submission

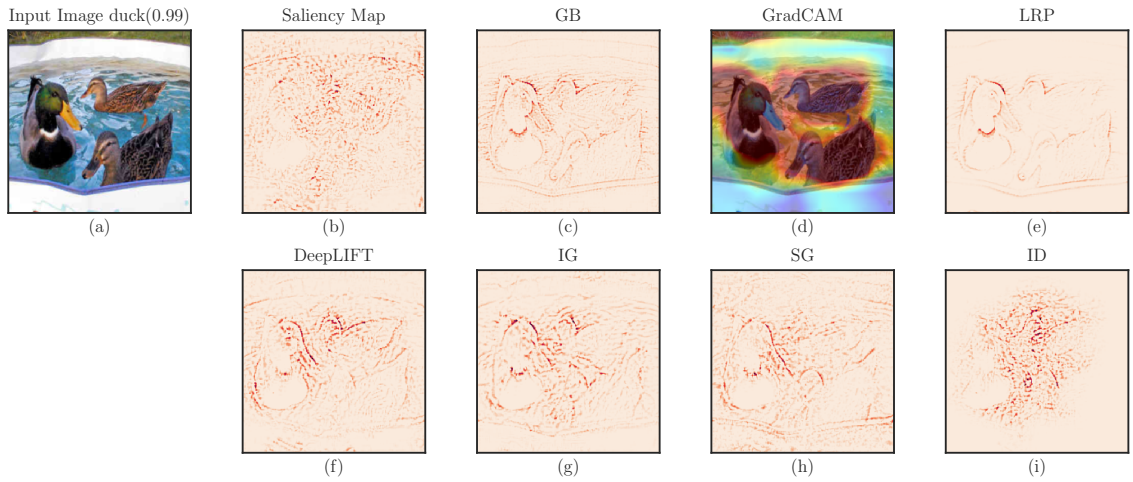

Figure 1. Visualizations of different attribution methods using VGG16 model [28]. (a) is the input image with an output confidence 0.99 for the true label "duck". (b)-(i): different attribution methods discussed in Section 2.1. *grad* $\times$ *input* is applied to (b), (c), (g), (h) and (i) to find attribution scores, while (d), (e) and (f) consider their results as attribution scores already. Use heatmap for (d) as the choice in [25].

the logical notion of necessity which leads to a corresponding sufficiency ordering not yet fully discussed in literature.

- We show that Saliency Map[6, 27] and GradCAM [25] are the ones we should avoid to use when localizing necessary input features, whereas DeepLIFT [26], Guided Backpropagation [30] and Influence-Directed Explanation [18] have higher probability to be the best method when localizing sufficient input features.

- We show no evaluated method in this paper can be a frequent winner on the necessity and sufficiency at the same time.

2. Background

Attributions are a simple form of model explanations that have found significant application to Convolutional Neural Networks (CNNs) with their ease of visualization alongside model inputs (i.e. images). We summarize the various approaches in Section 2.1 and the criteria with which they are evaluated and/or motivated in Section 2.2.

2.1. Attribution Methods

The concept of Attribution is well-defined in [31] but it excludes any method without an baseline (reference) input. We consider a relaxed version. Consider a classification model $\mathbf{y} = M(\mathbf{x})$ that takes an input vector $\mathbf{x}$ and outputs a score vector $\mathbf{y} = [y_0, \cdots, y_i, \cdots, y_{n-1}]^\top$, where y_i is the score of predicting $\mathbf{x}$ as class i and there are n classes in total. Given a pre-selected class c, an attribution method attempts to explain y_c by computing a score for each feature x_i as its contribution toward y_c. Even though each feature in $\mathbf{x}$ may receive different attribution scores given different choice of attribution methods, features with positive attribution scores are universally explained as important part in $\mathbf{x}$, while the negative scores indicate the presence of these features decline the confidence for predicting y_c.

Previous work has made great progress in developing gradient-based attribution methods to highlight important features in the input image for explaining model's prediction. The primary question to answer is whether should we consider *grad* or *grad* $\times$ *input* as attributions [19, 26, 29, 31]. As Ancona et al. [2] argues *grad* is *local attribution* that only accounts for how tiny change around the input will influence the output of the network but *grad* $\times$ *input* is the *global attribution* that accounts for the marginal effect of a feature towards output. We use *grad* $\times$ *input* as the attribution to be discussed in this paper. We briefly introduce methods to be evaluated in this paper and examples are provided in Fig 1.

Saliency Map [6, 27] uses the gradient of the class of interests with respect to the input to interpret the prediction result of CNNs. **Guided Backpropagation** (GB) [30] modifies the backpropagation of ReLU [13] so that only the positive gradients will be passed into the previous layers. **GradCAM** [25] builds on the Class Activation Map (CAM) [33] targeting CNNs. Although its variations [8, 23] show sharper visualizations, their fundamental concepts remain unchanged. We consider only GradCAM in this paper. **Layer-wise Relevance Propagation** (LRP) [5], **DeepLift** [26] modifies the local gradient and rules of backpropagations. Another method sharing similar motivation in design with DeepLift is **Integrated Gradient** (IG) [31]. IG computes attribution by integrating the gradient over a path from a pre-defined baseline to the input. **SmoothGrad** (SG) [29]

attempts to denoise the result of Saliency Map by adding Gaussian noise to the input and provides visually sharper results. **Influence-Directed Explanation** (ID) [18] identifies neurons in the model's internal presentation with high *distributional influence* toward the output of a target class of interest. Feature importance scores are considered as attributing to a group of internal neurons with high influence. The use of the internal modules of neural networks in explanation is also discussed by Olah et al. [22].

Other methods like Deep Taylor Decompostion [20] related with LRP [21] and Occluding [32] are not evaluated in this paper but will be a proper future work to discuss.

2.2. Evaluation Criteria

Evaluation criteria measure the performance of attribution methods towards some desirable characteristic and are typically employed to justify the use of novel attribution methods. The most common evaluations are based on pixel-level interventions or perturbations. These quantify the correlation between the perturbed pixels' attribution scores and the output change [3, 7, 8, 12, 21, 24, 26]. For perturbations that intend to remove or ablate a pixel (typically by setting it to some baseline or to noise), the desired behavior for an optimal attribution method is to have perturbations on the highly attributed pixels drop the class score more significantly than on the pixels with lower attribution.

Quantification of the behavior described by Samek et al. [24] with *Area Over Perturbation Curve (AOPC)*. It measures the area between two curves: the model's output score against the number of perturbed pixels in the input image and the horizontal line of the score at the model's original output. Another similar measurement is *Area Under Curve (AUC)* of Binder et al. [7], Montavon et al. [21] that measures the area under the perturbation curve instead. AOPC and AUC measurement are equivalent and both are orignally used to endorse LRP. For reasons which will become clear in the next section, we categorize these criteria as supporting necessity order. We argue that evaluating attribution methods only with perturbation curves, *e.g. Area Under Curve (or AUC)*, only discovers the tip of the iceberg and potentially can be problematic. A toy model is shown in Example 1 to elaborate our concerns.

Example 1. *Consider a model* $M(\mathbf{x}) = max(x_1, x_2)$ *that takes a vector* $\mathbf{x}$ *with three features* $x_1, x_2, x_3 \in \{0, 1\}$. *Given the input to the model is* $x_1 = x_2 = x_3 = 1$, *assume* A_1, A_2, A_3 *are three different methods and output the attribution scores* s_1, s_2, s_3 *shown in Table 1 for each input feature* x_1, x_2, x_3, *respectively.*

We apply zero perturbation to the input which means we set features to 0. The AOPC evaluation for these three attribution methods is shown in Fig 2. Using the conclusion from [24] that higher AOPC scores suggest higher relativity of input features highlighted by an attribution method, Fig 2 shows pixels highlighted by A_3 *are more relative Samek et al. [24] to prediction than* A_1 *and* A_2, *as expected. However,* A_2 *and* A_1 *are considered as showing same level of relativity under the AOPC measurement even though* A_2 *succeeds in discovering* a *is more relevant than* c, *whereas* A_1 *believes* c *is more relevant than both* a, b.

	s_1	s_2	s_3
A_1	$1/6$	$1/3$	$1/2$
A_2	$2/3$	0	$1/3$
A_3	$2/3$	$1/3$	0

Table 1. s_1, s_2, s_3 are attribution scores for x_1, x_2, x_3 computed by A_1, A_2, A_3, respectively.

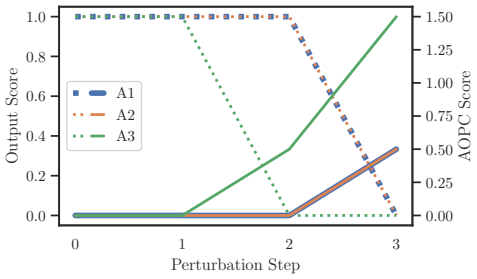

Figure 2. Comparing attribution methods A_1, A_2 and A_3 by applying zero perturbation. Dash lines are the change of model's output at each perturbation step (only one feature is set to 0 at each step). Solid lines are the changes of AOPC scores. A_2 and A_3 are overlapping with each other in this exmaple.

Another set of criteria instead stipulate that positively attributed features should stand on their own independently of non-important features. An example of this criterion is *Average % Drop* of Chattopadhay et al. [8] in support of Grad-Cam++ that measures the change of class score by presenting only pixels highlighted by an attribution (non-important pixels are ablated). We will say this criteria support sufficiency order (definition to follow).

Rethinking the concept of relativity, we believe both necessity and sufficiency can be treated as different types of relativity. In Example 1, neither x_1 nor x_2 is a necessary feature individually because the output will not change if any one of them is absent. However, both x_1 and x_2 are sufficient features, with either of which, the model could produce the same output as before. Besides, A_2 succeeds in placing the order of sufficient feature x_1 in front of the non-sufficient feature x_3 but A_2 fails, while AOPC(or AUC) is unable to discover the success.

Other evaluation criteria exist, like *sensitivity-n* [2] and sanity check [1], will be discussed in Section 5.

3. Methods

To tame the zoo of criteria, we organize and decompose them into two aspects: (1) *ordering* imposes conditions un-

der which an input should be more important than another input in a particular prediction, and (2) *proportionality* further specifies how attribution should be distributed among the inputs. We elaborate on ordering criteria in Section 3.1 with instantiations in Section 3.2 and Section 3.3. We describe proportionality in Section 3.4. We begin with the logical notions of necessity and sufficiency as idealized versions of ablation-based measures described in Section 2.

3.1. Logical Order

The notions of necessity and sufficiency are commonly used characterizations of logical conditions. A necessary condition is one without which some statement does not hold. For example, in the statement $P_1 = A \wedge B$, both A and B are necessary conditions as each independently would invalidate the statement were they be made false. On the other hand, a sufficient condition is one which can independently make a statement true without without other conditions being true. In the statement $P_2 = A \vee B$, both A and B are sufficient but neither are necessary.

In more complex statements, no atomic condition may be necessary nor sufficient though compound conditions may. In the statement $P_3 = (A \wedge B) \vee (C \wedge D)$, none of A, B, C, D are necessary nor sufficient but $(A \wedge B)$ and $(C \wedge D)$ are sufficient. As we are working in the context of input attributions, we relax and order the concept of necessity and sufficiency for atomic conditions (individual input pixels).

Definition 1 (Logical Necessity Ordering)**.** *Given a statement P over some set of atomic conditions, and two orderings a and b, both ordered sets of the conditions, we say a has better necessity ordering for P than b if:*

$$\min_i \left(\{a_k\}_{k \geq i} \not\models P \right) \leq \min_i \left(\{b_k\}_{k \geq i} \not\models P \right) \quad (1)$$

Definition 2 (Logical Sufficiency Ordering)**.** *Likewise, a has better sufficiency ordering for P than b if:*

$$\min_i \left(\{a_k\}_{k \leq i} \models P \right) \leq \min_i \left(\{b_k\}_{k \leq i} \models P \right) \quad (2)$$

A better necessity ordering is one that invalidates a statement P by removing the shorter prefix of the ordered conditions while a better sufficiency ordering is the one that can validate a statement using the shorter prefix.

3.2. Necessity Ordering (N-Ord)

Unlike logical statements, numeric models do not have an exact notion of a condition (feature) being present or not. Instead, inputs at some baseline value or noise are viewed as having a feature removed from an input. Though this is an imperfect analogy, the approach is taken by every one of the measures described in Section 2 that make use of perturbation in their motivation. Additionally, with numeric outputs, the nuances in output obtain magnitude and we can longer describe an attribution by a single index like the minimal index of Definitions 1 and 2. Instead we consider an ideal ordering as one which drops the numeric output of the model the most with the least number of inputs ablated.

Notation

Given an attribution method A, it computes a set of attribution scores $s_1, s_2, ..., s_n$ for each pixel $x_1, x_2, ..., x_n$ in the input image $\mathbf{x}$. We permute the pixels into a new ordering $\pi_A(\mathbf{x}) = [x'_1, x'_2, ..., x'_n]$ so that $s'_1 \geq s'_2 \geq ... \geq s'_n$. We take the subset $\pi^+_A(\mathbf{x})$ of $\pi_A(\mathbf{x})$ so that $\pi^+_A(\mathbf{x})$ has the same ordering as $\pi_A(\mathbf{x})$ but only contains pixels with positive attribution scores. Let $R_i(\mathbf{x}, \pi)$ be the output of the model with input $\mathbf{x}$ where pixels $x'_1, x'_2, ..., x'_i \in \pi$ are perturbed from the input by setting $x'_1 = x'_2 = ... = x'_i = b$, where b is a baseline value for the image (typically $b = 0$). Denote $R_0(\mathbf{x})$ as the original output of the model given $\mathbf{x}$ without any perturbation. Denote $\mathbf{x}_b$ as the the baseline input image where all the pixels are filled with the baseline value b.

We refer the AUC measurement [7, 21] as a means to measure the Necessity Ordering (N-Ord). Denote $N_o(\mathbf{x}, A)$ as N-Ord score given a input image $\mathbf{x}$ and an attribution method A. Rewrite AUC using the notation in Section 3.2:

$$N_o(\mathbf{x}, A) = \frac{1}{M+1} \sum_{m=0}^{M} R_0^m(\mathbf{x}, A) \quad (3)$$

where $R_0^m(\mathbf{x}, A) = max\{R_m(\mathbf{x}, \pi^+_A(\mathbf{x})) - R_0(\mathbf{x}_b), 0\}$ and M is the total number of pixels in $\pi^+_A(X)$. We include max to clip scores below the baseline output. According to Definition 1, we have the following proposition.

Proposition 1. *An attribution method A_1 shows a (strictly) better Ordering Necessity than another method A_2 given an input image $\mathbf{x}$ if $N_o(\mathbf{x}, A_1) < N_o(\mathbf{x}, A_2)$*

As discussed in Section 2.2, N-Ord only captures whether more necessary pixels, are receiving higher attribution scores. We argue that attribution methods should also be differentiated by the ability of highlighting sufficient features. To evaluate whether more sufficient pixels are receiving higher attribution scores, we propose Sufficiency Ordering as a complementary measurement.

3.3. Sufficiency Ordering (S-Ord)

Use the notation in Section 3.2 and let $R'_i(\mathbf{x}_b, \pi)$ be the model's output with $\mathbf{x}_b$ where $x'_1, x'_2, \cdots, x'_i \in \pi$ are added to the baseline image $\mathbf{x}_b$. Denote $S_o(\mathbf{x}, A)$ as S-Ord score given a input image $\mathbf{x}$ and an attribution method A.

$$S_o(\mathbf{x}, A) = \frac{1}{M+1} \sum_{m=0}^{M} R_0^{m'}(\mathbf{x}, A) \quad (4)$$

where $R_0^{m'}(\mathbf{x}, A) = min\{R'_m(\mathbf{x}_b, A), R'_M(\mathbf{x}_b, \pi_A^+(\mathbf{x}))\} - R_0(\mathbf{x})$, M is the number of pixels in $\pi_A^+(\mathbf{x})$. We include min to clip scores above the original output. $S_o(\mathbf{x}, A)$ is inspired by *Average % Drop* [8] even though *Average % Drop* only measures the final score of adding all pixels back to a baseline image because it is used for localization analysis. According to Definition 2, we have the following proposition.

Proposition 2. *An attribution method A_1 shows (strictly) better Ordering Sufficiency than another method A_2 given an input image X if $S_o(\mathbf{x}, A_1) > S_o(\mathbf{x}, A_2)$.*

N-Ord and S-Ord together provides a more comprehensive evaluation for an attribution method. In Section 3.4, we are going to discuss the disadvantages of only using N-Ord or S-Ord and propose Proportionality as a refinement to the ordering analysis.

3.4. Proportionality

N-Ord and S-Ord do not incorporate the attribution scores beyond producing an ordering. This can be an issue toward an accurate description of feature necessity or sufficiency. For example, consider a toy model $M(x_1, x_2) = 2x_1 + x_2$ and let the inputs variables be $x_1 = x_2 = 1$. Any attribution methods that assign higher score for x_1 than x_2 produces the identical ordering $\pi(x_1, x_2) = [x_1, x_2]$, even one could overestimate the degree of necessity (or sufficiency) of x_1 by assigning it with much higher attribution scores. With *linear agreement* [19], scores for x_1 and x_2 are more reasonable if their ratio is close to 2:1. Explaining a decision made by a more complex model only using ordering of attributions may overestimate or underestimate the necessity (or sufficiency) of an input feature. Therefore, We propose Proportionality as a refinement to quantify the necessity and sufficiency in complementary to the ordering measurement.

Definition 3 (Proportionality-k for Necessity)**.** *Consider two positive number n_1, n_2 and an attribution method A. Use notations in Section 3.2 and let $\hat{\pi}_A^+(\mathbf{x})$ be a reversed ordering of $\pi_A^+(\mathbf{x})$. Proportionality-k for Necessity is measured by*

$$N_p^k(\mathbf{x}, A) = |R_{n_1}(\mathbf{x}, \pi_A^+(\mathbf{x})) - R_{n_2}(\mathbf{x}, \hat{\pi}_A^+(\mathbf{x}))| \quad (5)$$

under the condition $\sum_i^{n_1} s_i = \sum_j^{n_2} s_j = kS(A, \mathbf{x}), s_i \in \pi_A^+(\mathbf{x}), s_j \in \hat{\pi}_A^+(\mathbf{x}), k \in [0, 1]$. $R_i(\mathbf{x}, \pi)$ uses the same definition in (3), and $S(\mathbf{x}, A)$ is the sum of total positive attribution scores.

Explanation of Definition 3 the motivation behind Proportionality-k for Necessity is that: given a group of pixels ordered with their attribution scores, there are different ways of distributing scores to each feature while the ordering remains unchanged. An optimal assignment is preferred that features receive attribution scores proportional to the output change if they are modified accordingly. In other words, given any two subsets of pixels π_1 and π_2. with total attribution scores sum to S_1 and S_2, are perturbed, the change of output scores $R(\mathbf{x}, \pi_1)$ and $R(\mathbf{x}, \pi_2)$ should satisfy $R(\mathbf{x}, \pi_1)/R(\mathbf{x}, \pi_2) = S_1/S_2$. This property is demanded because the same share of attribution scores should account for the same necessity or sufficiency. If we restrict the condition to $S_1 = S_2$, the difference between $R(\mathbf{x}, \pi_1)$ and $R(\mathbf{x}, \pi_2)$ becomes an indirect measurement of the proportionality. For the measurement of Necessity, we further restrict that π_1 is perturbed from the pixel with the highest attribution score first and π_2 is perturbed from the one with lowest attribution score first, in accordance with the setup in N-Ord. Therefore, a smaller difference $N_p^k(\mathbf{x}, A)$ shows better Proportionality-k for Necessity

Proposition 3. *An attribution method A_1 shows better Proportionality-k for Necessity than method A_2 if $N_p^k(\mathbf{x}, A_1) < N_p^k(\mathbf{x}, A_2)$*

A similar requirement for attribution method is *completeness* discussed by [31] and its generalization *sensitivity-n* discussed by [2]. *completeness* requires the sum of total attribution scores to be equal to the change of output compared to a baseline input, and *sensitivity-n* requires any subset of n pixels whose summation of attribution scores should be equal to the change of output compared to the baseline if pixels in that subset are removed. When n is the total number of pixels in the input image, *sensitivity-n* reduces to *completeness*. The relationships between *sensitivity-n* and Proportionality-k for Necessity are discussed as follows:

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.*

The proof for Proposition 8 and can be found in Appendix 1. We further contrast our method with *sensitivity-n* in Section 5. Integrating *proportionality* with all possible shares of attribution scores, we define the Total Proportionality for Necessity (TPN):

Definition 4 (Total Proportionality for Necessity)**.** *Given an attribution method A and an input image $\mathbf{x}$, The Total Proportionality for Necessity is measured by*

$$N_p(\mathbf{x}, A) = (1 + e^{\alpha r}) \int_0^1 N_p^k(\mathbf{x}, A) dk \quad (6)$$

where $r = \max(R_M(\mathbf{x}, \pi_A^+(\mathbf{x})) - B, 0)/R_0(\mathbf{x})$, the ratio of the model's output after perturbing all pixels with positive attribution scores over its original output. Details of

notations are dicussed in Section 3.2. α is a positive hyperparameter.

Explanation for Definition 4 $N_p(\mathbf{x}, A)$ is the area between two perturbation curves one starting from the pixels with highest attribution scores and the other with a reversed ordering. The difference from Necessity Ordering is that $N_p(\mathbf{x}, A)$ is measured against the share of attribution scores (the value of k) instead of the share of pixels in the $N_o(\mathbf{x}, A)$. α is a hyperparameter to adjust the penalty r when an attribution method highlights all non-necessary features instead of the necessary ones (*e.g.* assign similar high scores to the background when classifying ducks and 0s for duck-related features). Perturbations on non-necessary features may not change the output at all and we penalize an method for this. Generalizing Proposition 3, we argue:

Proposition 5. *An attribution method A_1 shows better Total Proportionality for Necessity than method A_2 if $N_p(\mathbf{x}, A_1) < N_p(\mathbf{x}, A_2)$*

Under the similar construction, we have the following definition of Proportionality-k for Sufficiency and Total Proportionality for Sufficiency (TPS):

Definition 5 (Proportionality-k for Sufficiency)**.** *Consider two positive number n_1, n_2 and an attribution method A. Use notations in Section 3.2 and let $\hat{\pi}_A^+(\mathbf{x})$ be a reversed ordering of $\pi_A^+(\mathbf{x})$. Proportionality-k for Sufficiency is measured by*

$$S_p^k(\mathbf{x}, A) = |R'_{n_1}(\mathbf{x}_b, \pi_A^+\mathbf{x})) - R'_{n_2}(\mathbf{x}_b, \hat{\pi}_A^+(\mathbf{x}))| \quad (7)$$

under the condition $\sum_i^{n_1} s_i = \sum_j^{n_2} s_j = kS(\mathbf{x}, A), s_i \in \pi_A^+(\mathbf{x}), s_j \in \hat{\pi}_A^+(\mathbf{x}), k \in [0, 1]$. $R'_i(\mathbf{x}, \pi)$ reuses the definition in (4); $S(\mathbf{x}, A)$ is the sum of total positive attribution scores.

We want the difference $S_p^k(\mathbf{x}, A)$ as small as possible since the same share of attribution scores should reflect same sufficiency. Therefore, we have the following proposition:

Proposition 6. *An attribution method A_1 shows better Proportionality-k for Sufficiency than method A_2 if $S_p^k(\mathbf{x}, A_1) < S_p^k(\mathbf{x}, A_2)$*

Definition 6 (Total Proportionality for Sufficiency)**.** *Given an attribution method A and an input image $\mathbf{x}$, The Total Proportionality for Sufficiency is measured by*

$$S_p(\mathbf{x}, A) = (1 + e^{\beta(1-r')}) \int_0^1 S_p^k(\mathbf{x}, A) dk \quad (8)$$

where $r' = \min\{R'_M(\mathbf{x}, \pi_A^+(\mathbf{x})), R_0(\mathbf{x})\}/R_0(\mathbf{x})$, the ratio of the model's output after adding all pixels with positive attribution scores to a baseline input over its original output. Refer to Section 3.2 and 3.3 for details about the notation. β is a positive hyperparameter.

Similarly, $S_p(\mathbf{x}, A)$ is the area between curves of model's output change by adding pixels to a baseline input with the highest attribution scores first or by the lowest first. β is a hyperparameter to adjust the penalty $(1 - R')$ when an attribution method highlights non-sufficient features. Adding those pixels will not increase the output significantly. Finally, we have

Proposition 7. *An attribution method A_1 shows better Total Proportionality for Sufficiency than another method A_2 if $S_p(\mathbf{x}, A_1) < S_p(\mathbf{x}, A_2)$*

In summary, we differentiate and describe the Necessity Ordering (N-Ord) and Sufficiency Ordering (S-Ord) from previous work and propose Total Proportionality for Necessity (TPN) and Total Proportionality for Sufficiency (TPS) as refined evaluation criteria for necessity and sufficiency. We then apply our measurement to explain the prediction results from an image classificiation task in the rest of the paper.

4. Evaluation

We evaluate our metrics directly on CNNs. A linear model may be a reasonable choice to begin with but as Ancona et al. [2] concludes that SM, IG, LRP,DeepLIFT are equivalent for linear models. Their proof also applies to SG, BP and ID. GradCAM, on the other hand, is not defined for models without convolutional layers. Linear models are therefore not expected to distinguish most attribution methods

4.1. Datasets and Models

We evaluate the necessity and sufficiency for all attribution methods mentioned in Section 2 on VGG16[28] with pretrained weights from ImageNet and fine-tuned on Caltech-256 [14]. Without data augmentation and preprocessing, the model achieves an accuracy of 64.7% on the test dataset containing 5000 images.

4.2. Implementation

We pick a zero baseline in IG and DeepLIFT (RevealCancel) for all images. The last convolutional layer in VGG16 is selected for GradCAM as Selvaraju et al. [25] suggested. For LRP, a variety of rules are available. We use the implementation of LRP-$\alpha 2\beta 1$ with generalization tricks mentioned by Montavon et al. [21] who argues this rule is better for image explanations. For ID, we employ *instance distribution of interests* [19] (simple gradient) to visualize the top 1000 neurons in `block4_conv3` for each class. We also use the same convolutional block in as Leino et al.

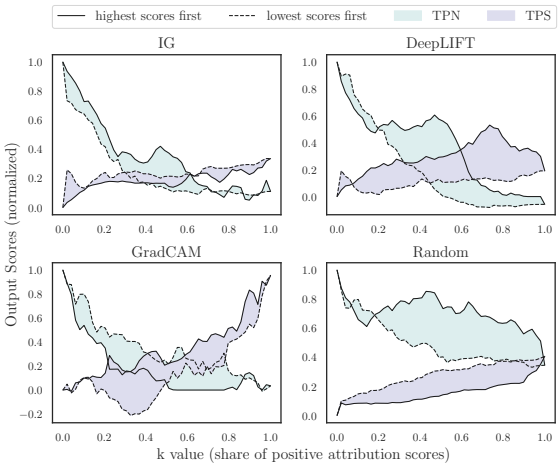

Figure 3. Total Proportionality for Necessity (TPN, area in green) and Total Proportionality for Sufficiency (TPS, area in purple) using the input image from Fig 1. A baseline method of randomly assigning scores to pixels is used for comparison. The y-axis for each sub-figure is the output scores of the label "duck" and the x-axis for each sub-figure is the k value (share of positive attribution scores). Solid lines perturb the pixels with highest attribution scores first and dash lines perturb in a reversed order. For TPN, smaller area with lower score at $k = 1.0$ indicates better Necessity. For TPS, smaller area with higher score at $k = 1.0$ indicates better Sufficiency. Only four methods are shown here and more details for others are provided in Appendix 2(a)

.

[19] did but a deeper slice expecting for deeper feature representations (the author uses `conv4_1`). We believe a better slice can be argued but we find slice searching remains an open question. For the penalty terms in TPN and TPS, we use $\alpha = 1.0$ and $\beta = 3.0$. In the experiment, we observe even random perturbation can cause serious drop in the output, so there is no need for big penalty in TPN. However for TPS, even adding all pixels with positive attribution scores, the model can produce a very low scores compared to the original ones, for some instances. Therefore, we slightly increase β to encourage the attribution method that can locate sufficient features.

4.3. Evaluate with one instance

We evaluate Necessity Ordering, Sufficiency Ordering, TPN and TPS reusing the example from Fig 1 before applying to the whole dataset. Results and comparisons are provided in table 2 and the computations for TPN and TPS before applying the penalty terms are demonstrated in Fig 3. Insights are discussed as below.

Under necessity analysis, N-Ord shows DeepLIFT finds finds the best ordering of pixels for necessity. A deeper look with TPS shows that IG is best at assigning pixels with scores proportional to the score drop of the output when necessary pixels are perturbed in Fig 1.

Under sufficiency analysis, S-Ord shows GradCAM finds the best ordering of pixels for sufficiency. A deeper look with TPS also show GradCAM is best at assigning scores proportional to the score increase of the model when sufficient pixels are provided in Fig 1.

	1st	2nd
N-Ord	DeepLIFT(.10)	GradCAM (.13)
S-Ord	GrdCAM (.45)	GB(.43)
TPN	IG (.23)	GB(.27)
TPS	GradCAM (.43)	IG(.44)

Table 2. Comparisons of attribution methods with Fig 1. Only the top 2 winners with their scores are shown. We apply penalty $\alpha = 1.0$ to TPN and $\beta = 3.0$ to TPS (see Section 4 parameter discussion). For S-Ord, the higher the better. For the rest metrics, the lower the better. More detailed results can be found in Appendix 2(b)

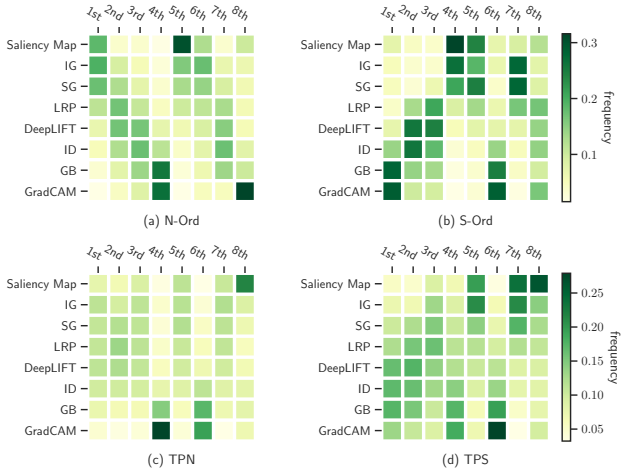

Figure 4. Heatmaps of the frequency for an attribution method receives a particular ranking compared with all methods (*e.g.* the grid (SG, 2nd) in (a) indicates the frequency of SG being ranked as the second good method for N-Ord). A darker color indicates that an method secures that ranking across more instances (high frequency). SG and IG have a higher frequency to show the best N-Ord while GB and GradCAM have a higher frequency to show the best S-Ord. TPN and TPS are calculated with penalty terms $\alpha = 1.0, \beta = 3.0$ (see Section 4 parameter discussion).

4.4. Evaluate with Caltech256

We evaluate N-Ord, S-Ord, TPN and TPS on the test dataset of Caltech256. For each instance, we compare the the N-Ord and S-Ord scores with other methods, ranking all the attribution methods from the 1st to 8th, where 1st means it has lowest N-Ord or highest S-Ord compared to other methods, respectively. As the result provided in Fig 4, we compute the frequency of an attribution method being placed at a particular rank position. A grid with darker

color suggests a higher frequency of corresponding ranking. Based only on ordering analysis from Fig 4 (a) (b), we show that IG and SG are more often to be the best in Necessity Ordering, while GB and GradCAM are more often to be the best in Sufficiency Ordering. In combination with the proportionality analysis, we find the winner of necessity varies but generally Saliency Map, GB and GradCAM are worse methods that should be avoided because they are often the poor ones. For sufficiency, DeepLIFT, ID and GB are more often the best methods. However, regardless of ordering or proportionality analysis, no method is an obvious frequent winner on both sides.

5. Related Work

We consider our work as a subset of **sensitivity** evaluation that how well we can trust an attribution method with its quantification of the feature importance in the input. A close concept is *quantitative input influence* by Datta et al. [9] (even though the author does not target on deep neural networks). *sensitivity (a)(b)* [31] provides the basis of discussion and *sensitivity-n* [2] imposes more strict requirements. The main correlation of *proportionality* with *sensitivity-n* is discussed in Section 3.4. We discuss the main difference of these two concepts here. *proportionality* approaches the sensitivity from a view that, regardless of the number of pixels, same share of attribution scores should account for same change to the output, while *sensitivity-n* requires removing n pixels should change the output by the amount of total attribution scores of that n pixels. *sensitivity-n* only provides *True/False* to an attribution methods, but *proportionality* provides numerical results for comparing different methods under the necessity and sufficiency. Beyond sensitivity, the **continuity** of attribution methods is an important and desired property, which requires an attribution method can to output similar results for similar input and prediction pairs. Ghorbani et al. [11] discussed this property , Montavon et al. [21] and Kindermans et al. [16] provide failure cases and well-designed attacks that causes unreasonable attribution results. Besides, [1] evaluates the correlation between attribution scores with the model's parameters. Outside the discussion around CNNs, [4] provides evaluations on applying attribution methods to language tasks with LSTM [15].

6. Discussion

In this section, we discuss how this paper helps to correct some potential misunderstanding from the interpretation of attribution methods. We argue meaningful interpretations should depend on the purpose of interpretation and the criteria in evaluation.

6.1. Interpretations based on purpose

In this paper, we provide provide two realizable purposes of interpretation: identifying more necessary or more sufficient features. The purposes can be realized by using attribution methods winning the necessity or sufficiency test, respectively. Failing to be a better attribution method in showing necessity does not mean an method can not be showing insight on the sufficiency side, *e.g.* GB is doing poorly in N-Ord test but fairly good in S-Ord test.

6.2. Interpretations based on criteria

We provide ordering and proportionality as two criteria in this paper. Interpretation based on ordering criteria is safe to argue the features with higher attribution scores in the input image is more necessary or sufficient than others, but is not safe to include any quantitative analysis, *e.g.* feature x_1 is as twice necessary as x_2 or x_1 is equivalently necessary as the presence of x_2 and x_3 together because they have similar share of attribution scores. Interpretations based on proportionality, however, provides more confidence in making quantitative argument with the winner attribution method. For example, GradCAM changes from one of the most frequent winners in S-Ord to the most frequent 6th in TPS. Revisiting the algorithm behind GradCAM, it computes scores based on the activation of the selected layer weighted sum by the gradients. We believe GradCAM is outstanding in localizing which part is more sufficient in the image but we find it hard to justify the activation of an internal layer is proportional to the necessity or sufficiency of an input feature.

7. Conclusion

In this paper, we summarize existing evaluation metrics for attribution methods and categorize them into two logical concepts, necessity and sufficiency. We then demonstrate realizable criteria to quantify necessity and sufficiency with the ordering analysis and its refinement, proportionality analysis. We evaluate existing attribution methods against our criteria and list the best methods for each criteria. We discover that certain attribution methods excel in necessity or sufficiency, but none is a frequent winner for both.

Acknowledgement

This work was developed with the support of NSF grant CNS-1704845 as well as by DARPA and the Air Force Research Laboratory under agreement number FA8750-15-2-0277. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation thereon. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official

views or policies of DARPA, the Air Force Research Laboratory, the National Science Foundation, or the U.S. Government.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU and Titan V GPU used for this research.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps, 2018.

[2] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks, 2017.

[3] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. A unified view of gradient-based attribution methods for deep neural networks. 11 2017).

[4] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations, 2019.

[5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, Wojciech Samek, and Oscar Déniz Suárez. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. In *PloS one*, 2015).

[6] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Mueller. How to explain individual classification decisions, 2009.

[7] Alexander Binder, Grégoire Montavon, Sebastian Bach, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for neural networks with local renormalization layers, 2016.

[8] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar 2018).

[9] A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 598–617, May 2016).

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2017.

[11] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile, 2017.

[12] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning, 2018.

[13] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011). PMLR.

[14] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007).

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997).

[16] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un)reliability of saliency methods, 2017.

[17] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017).

[18] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks, 2018.

[19] Klas Leino, Shayak Sen, Anupam Datta, Matt Fredrikson, and Linyi Li. Influence-directed explanations for deep convolutional networks. In *2018 IEEE International Test Conference (ITC)*, pages 1–8. IEEE, 2018).

[20] Grégoire Montavon, Sebastian Bach, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition, 2015.

[21] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, Feb 2018).

[22] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018). https://distill.pub/2018/building-blocks.

[23] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models, 2019.

[24] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, Nov 2017).

[25] 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, 2016.

[26] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences, 2017.

[27] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[29] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014.

[31] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*,

pages 3319–3328. JMLR. org, 2017).
[32] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.
[33] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. *CVPR*, 2016).

Appendix

Appendix 1

Proposition 8. *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 $\sum_i^{n_1} s_i = \sum_j^{n_2} s_j = kS(\mathbf{x}, A), s_i \in \pi_A^+(\mathbf{x}), s_j \in \hat{\pi}_A^+(\mathbf{x}), k \in [0, 1]$,

$$\begin{aligned} N_p^k(\mathbf{x}, A) &= |R(\mathbf{x}, \pi_A(\mathbf{x}) - R(\mathbf{x}, \pi_A^+(\mathbf{x})| \\ &= |\sum_i^{n_1} s_i - \sum_j^{n_2} s_j| \\ &= |kS(\mathbf{x}, A) - kS(\mathbf{x}, A)| = 0 \end{aligned} \tag{9}$$

□

Appendix 2(a)

See Fig 5 for the whole example of computing TPN and TPS for the example image in Fig 1

Appendix 2(b)

Organizing attribution methods for Example 1. Notice that we normalize the output scores at each each perturbation step by the original output in Table 2 and Table 3.

	N-Ord	**S-Ord**
1st	DeepLIFT (.10)	GradCAM (.44)
2nd	GradCAM (.13)	GB (.43)
3rd	IG (.17)	LRP (.36)
4th	SG (.19)	DeepLIFt (.33)
5th	ID (.29)	SG (.28)
6th	GB (.38)	IG (.25)
7th	LRP (.41)	Random (.22)
8th	Saliency Map (.41)	ID (.14)
9th	Random (.68)	Saliency Map (.14)
	TPN	**TPS**
1st	IG(.23)	GradCAM (.43)
2nd	GB (.27)	IG (.44)
3rd	GradCAM (.37)	GB (.51)
4th	DeepLIFT (.40)	Saliency Map (.56)
5th	SG (.42) (.43)	LRP(.84)
6th	Random (.56)	Random(.86)
7th	Saliency Map (.62)	SG(.96)
8th	LRP (.71)	DeepLIFT(2.47)
9th	ID (.79)	ID(.2.74)

Table 3. (normalized) Scores for each attribution methods

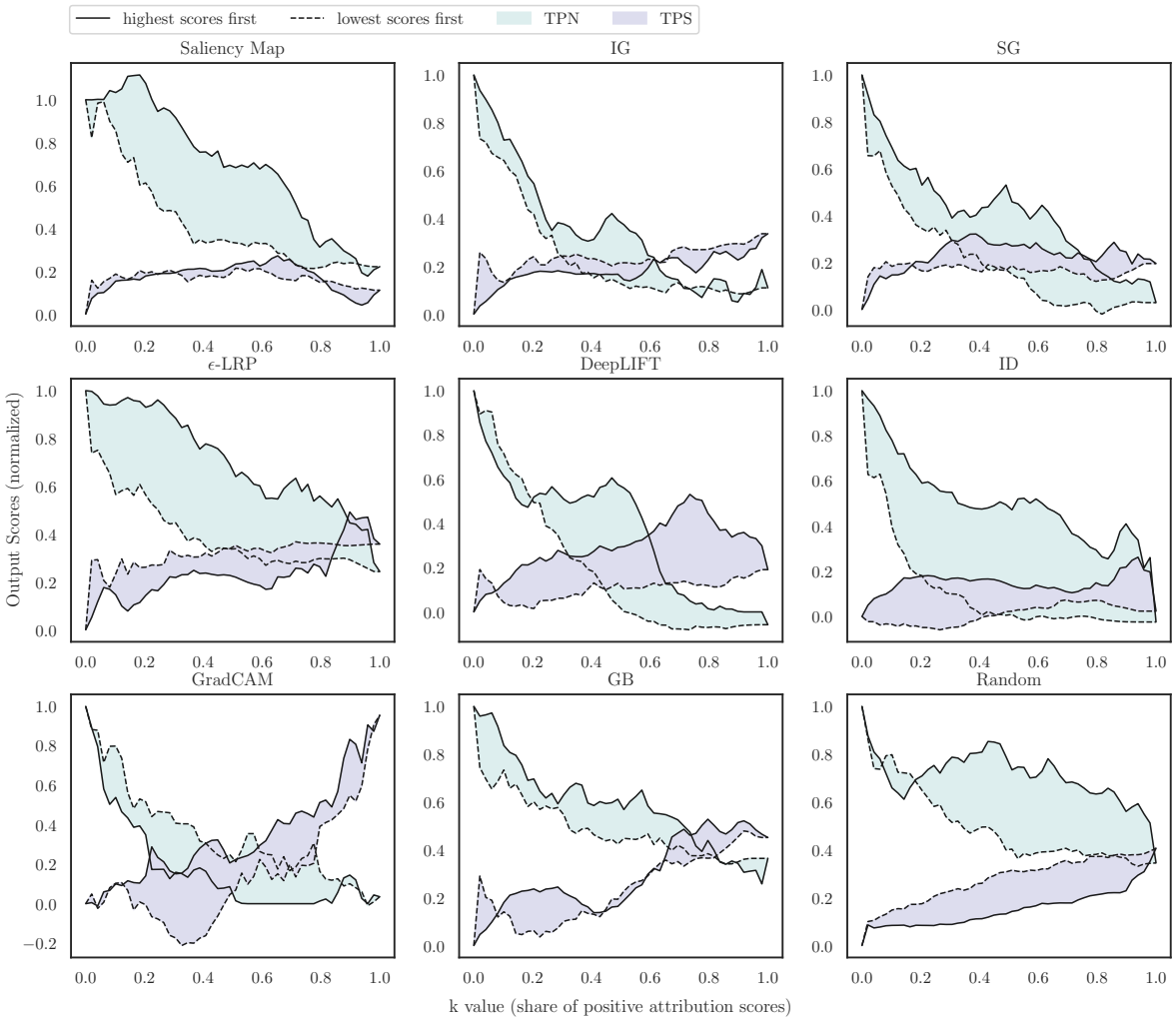

Figure 5. Total Proportionality for Necessity (TPN, area in green) and Total Proportionality for Sufficiency (TPS, area in purple) using the input image from Fig 1. A baseline method of randomly assigning scores to pixels is used for comparison. The y-axis for each sub-figure is the output scores of the label "duck" and the x-axis for each sub-figure is the k value (share of positive attribution scores). Solid lines perturb the pixels with highest attribution scores first and dash lines perturb in a reversed order. For TPN, smaller area with lower score at $k = 1.0$ indicates better Necessity. For TPS, smaller area with higher score at $k = 1.0$ indicates better Sufficiency.

.