Do Feature Attribution Methods Correctly Attribute Features?

Yilun Zhou[1] **Serena Booth**[1] **Marco Tulio Ribeiro**[2] **Julie Shah**[1]

[1]MIT CSAIL [2]Microsoft Research

{yilun, serenabooth, julie_a_shah}@csail.mit.edu

marcotcr@microsoft.com

Abstract

Feature attribution methods are exceedingly popular in interpretable machine learning. They aim to compute the attribution of each input feature to represent its importance, but there is no consensus on the definition of "attribution", leading to many competing methods with little systematic evaluation. The lack of ground truth for feature attribution particularly complicates evaluation; to address this, we propose a dataset modification procedure where we construct attribution ground truth. Using this procedure, we evaluate three common interpretability methods: saliency maps, rationales, and attention. We identify several deficiencies and add new perspectives to the growing body of evidence questioning the correctness and reliability of these methods in the wild. Our evaluation approach is model-agnostic and can be used to assess future feature attribution method proposals as well. Code is available at `https://github.com/YilunZhou/feature-attribution-evaluation`.

1 Introduction

Consider the task of training a neural network to detect cancers from X-ray images. Suppose the data come from two sources: a general hospital and a specialized cancer center. As can be expected, images from the cancer center contain many more cancer cases. Further, let's say the cancer center adds a small timestamp watermark to the top-left corner of its images, such that the presence of a watermark may affect the model's prediction, as it is a strong predictor of a cancer diagnosis.

It is important to ensure the deployed model makes predictions based on genuine medical signals rather than image artifacts like watermarks. If these artifacts are known *a priori*, we can evaluate the model on counterfactual pairs—images with and without them—and look for prediction difference to assess their impact. Realistically, however, for almost all datasets in the wild we cannot anticipate every possible artifact. As such, feature attribution methods such as saliency maps [4, 22, 26, 29, 32, 33] are used to identify regions most important to the prediction, which are then inspected for evidence of any potential artifacts, e.g. watermarks. Such a train-and-interpret pipeline has been widely adopted in data-driven medical diagnosis [24, 30, 31] and many other applications.

However, this feature attribution-based assessment requires that the attribution methods work correctly and do not miss features that are influential to the model. Is this truly the case? Direct evaluation on natural datasets is impossible as the very spurious correlations we want attribution methods to find are, by definition, unknown. Current evaluations try to sidestep this problem with proxy metrics [6, 14, 28], which are unfortunately limited in numerous ways by a lack of ground truth (see Sec. 2).

Instead, we propose evaluating these attribution methods on *semi-natural* datasets: natural datasets modified with specific manipulations which introduce ground truth *for attributions*. This modification (Fig. 1) ensures that *any* classifier with sufficiently high performance has to rely, sometimes solely, on the manipulations. We then present desiderata, or necessary conditions, for assessing the correct attribution values; for example, features known to not affect the model's decision should not receive attribution. Our dataset-based evaluation is agnostic to both the model and the attribution method. In experiments, we evaluate saliency maps, rationale models, and attention mechanisms and identify several failures. We discuss potential reasons for such failures and recommend directions of remedy.

1st Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4Debugging@NeurIPS2021).

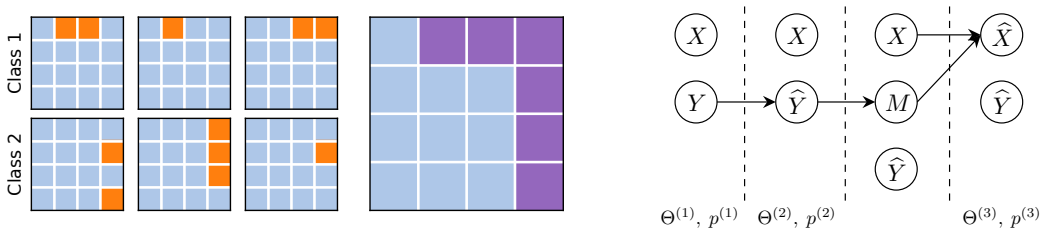

Figure 1: Left: the intuition behind our feature attribution ground truth: if we know that for every input, only specific features (orange) are informative to the label, then across the dataset, a high-performing model has to focus on them and not get "distracted" by other irrelevant ones. Thus, the feature attribution should focus on the *union* of these features (purple), *formally called the effective region (ER)*, and any attribution outside is misleading. Right: our proposed procedure to induce such a ground truth by modifying a given dataset. Arrows indicate modifications. (1) A model $\Theta^{(1)}$ trained on dataset (X, Y) achieves a generalization accuracy of $p^{(1)}$. Without further knowledge, the performance upper bound is perfect accuracy, $p^{(1)} \leq 1$. (2) We modify the label Y to a (more) stochastic version $\widehat{Y}$. Now the accuracy of new model $\Theta^{(2)}$ is bounded by some p^*: $p^{(2)} \leq p^* < 1$. (3) We introduce a manipulation M, which can be abstractly represented as e.g. a binary variable, conditioned on $\widehat{Y}$. (4) We apply M on X to obtain $\widehat{X}$ and train a model $\Theta^{(3)}$ on $(\hat{X}, \hat{Y})$. For example, if M represents watermarking, $\widehat{X}$ is a watermarked version of image X if $M = 1$, and X if $M = 0$. *If $\Theta^{(3)}$ achieves an accuracy $p^{(3)} > p^*$, it must use the effect of manipulation M.* Now, we can evaluate feature attribution methods to see if they can recognize the ground truth contribution of M.

2 Related Work

Feature attribution methods assign attribution scores to input features, the absolute value of which informally represents their importance to the model prediction or performance. In the experiments, we evaluate three method families, saliency maps [32], attention mechanisms [5], and rationale models [21]. App. A covers the specific methods we use from each family in detail.

Various evaluation methods have been proposed for feature attribution methods. A popular way is to assess alignment with human judgment, but models and humans can reach the same prediction while using entirely different reasoning mechanisms (e.g. medical signals used by doctors and watermarks used by the model). For example, SmoothGrad [33] is proposed as an improvement to the original Gradient method [32] since it gives less noisy and more legible saliency maps, but it is not clear that saliency maps *should* be smooth. Covert et al. [8] compared the feature attribution of a cancer prediction model to scientific knowledge, yet even a well-performing model may rely on other signals.

Another common approach successively removes features with the highest attribution values and evaluates certain metrics. One metric is prediction change [e.g. 3, 15, 28], but it fails to account for nonlinear interactions: for an OR function of two active inputs, the evaluation will (incorrectly) deem whichever feature is removed first to be useless as its removal does not affect the prediction. Another metric is model retraining performance [14], which may fail when different features lead to the same accuracy—as is often possible [9]. For example, a model might achieve some accuracy by using only feature x_1. If a retrained model using only x_2 achieves the same accuracy, the evaluation framework would (falsely) reject the ground truth attribution of x_1 due to the same re-training accuracy.

Most similar to our proposal are works that also construct semi-natural datasets with explicitly defined ground truth explanations [2, 37]. Adebayo et al. [2] used a perfect background correlation for a dog-vs-bird dataset, found that the model achieves high accuracy on background alone, and claimed that the correct attribution should focus solely on the background. However, we verified that a model trained on their dataset can achieve high accuracy *simultaneously* on foreground alone, background alone, and both combined, invalidating their claimed ground truth. Similarly, Yang and Kim [37] argue that for *background* classification, a more label-correlated foreground should receive higher attribution value. However, a model could always rely solely on background regardless of foreground correlation. We avoid such pitfalls via label reassignment (Sec. 4), so that the model *must* to use certain features to achieve high accuracy. A more subtle failure mode is discussed in Sec. 4 Remark.

Finally, Adebayo et al. [1] proposed sanity checks for saliency maps by assessing their change under weight or label randomization. We establish complementary criteria for explanations by instead focusing on model-agnostic dataset-side modifications, and identify additional failure cases. A concurrent work [20] also proposed to establish attribution ground truth, but on purely synthetic image data, while we construct semi-natural datasets to evaluate both vision and text models.

3 Desiderata for Attribution Values

What should the attribution values be? Although precise values may be axiomatic, certain properties are *de facto* requirements if we want people to understand how a model makes a decision, verify that its reasoning process is sound, and possibly inform options for correction if it is not (c.f. the opening example in Sec. 1). For example, while LIME and SHAP define attribution differently, both would produce undeniably bad explanations if they highlight features completely ignored by the model.

We study two types of features: those of fundamental importance to the model, denoted by F_C, and those non-informative to the label, denoted by F_N. A first requirement is that explanations should not miss important features, F_C. Unfortunately, identifying all such features is not easy. For example, while the model could potentially use the timestamp on some X-ray images for cancer prediction, it could instead exclusively rely on genuine medical features (as done by human doctors), and attributions should only highlight the timestamp in the former case. This difficulty motivates our dataset modification procedure detailed in the next section. In brief, we can modify the dataset such that any model using only medical features could not achieve a high accuracy (due to introduced label noise), thus establishing the ground truth usage of the timestamp for any model with high accuracy. We can then evaluate how well the attribution method identifies the contribution of the timestamp by the attribution percentage $\mathrm{Attr}\%$ of the timestamp pixels, with $\mathrm{Attr}\%(F) \doteq \left(\sum_{i\in F} |s_i|\right) / \left(\sum_{i=1}^{D} |s_i|\right)$, where D is the total number of features and s_i is the attribution value assigned to the i-th feature. Since F_C contains all features used by the model, we should expect $\mathrm{Attr}\%(F_C) \approx 1$.

Conversely, we can introduce non-informative features F_N independent from the label—for example, a white border added to randomly selected images. While the model prediction could depend on it (e.g. more positive for those with the border), methods that study features contributing to the performance should not highlight F_N. In addition, any reliance on F_N is detrimental to performance, and as performance increases, a good prediction is less "distracted" by F_N, which should correspondingly not get highlighted, i.e. as model performance increases, $\mathrm{Attr}\%(F_N) \to 0$.

In addition to continuous attributions on all features, sometimes only the top-k matters, and with no distinction within them. This can either be derived in post-processing to induce sparse explanations, or generated directly by some models, e.g. rationales [21]. For the first case, a hyper-parameter k needs to be chosen. A small value risks missing important features while a large value may include unnecessary features that obfuscate true model reasoning. For the second case, k is typically chosen automatically by the model, e.g. the rationale selector. In both cases, ensuring that F_C is highlighted (i.e. $\mathrm{Attr}\% = 1$) is easily "hackable" by just selecting all features. As such, we instead use two information-retrieval metrics, precision and recall, defined as $\mathrm{Pr}(F) = |F \cap F_C|/|F|$, and $\mathrm{Re}(F) = |F \cap F_C|/|F_C|$ for evaluating feature-selection attributions, where F is the top-k features.

4 Dataset Modification to Establish Ground Truth

We now present the dataset modification procedure that lets us quantify the influence of certain features to the model. We use a running example of adding a watermark pattern to a watermark-free X-ray cancer dataset, such that the newly added watermark is guaranteed to affect the model decision.

Let $\mathcal{X}$ and $\mathcal{Y} \doteq \{1, ..., K\}$ be input and output space for K-class classification. As shown in Fig. 1, the original dataset is modified in two steps: label reassignment to reduce predictive power of existing signals (Fig. 1-(2)) and input manipulation to introduce new predictive features (Fig. 1-(3) & (4)).

Label Reassignment Our goal is to ensure that the model has to rely on certain introduced features (e.g. a watermark) to achieve a high performance. However, the model could in theory use any of the existing features (e.g. medical features) to achieve high accuracy, and thus disregard the new feature, even if it is perfectly correlated with the label. To guarantee the model's usage of input manipulation, we need to weaken the correlation between the original features and the labels.

For simplicity, we consider label reassignment for binary classification. Extension to the K-class setting is similar, and is detailed in App. B. In label reassignment, the label is preserved with probability r and flipped otherwise, such that the accuracy without relying on the manipulation is at most $p^* = \max(r, 1-r)$. For the special case of $r = 0.5$, no features are informative to the label, and expected performance is random. After label reassignment, a data point (x, y) becomes $(x, \widehat{y})$.

Input Manipulation Next, we apply manipulations on the input according to its reassigned label. We consider a set of L input manipulations, $\mathcal{M} = \{m_1, ..., m_L\}$, and a manipulation function

$q : \mathcal{M} \times \mathcal{X} \to \widehat{\mathcal{X}}$ such that $q(m_l, x) = \widehat{x}$ applies the manipulation on the input and returns the manipulated output $\widehat{x}$. $\mathcal{M}$ can include the blank manipulation $m_\varnothing$ that leaves the input unchanged.

To facilitate feature attribution evaluation, we require the manipulation to be *localized*, in that $q(m, x)$ affects only a part of the input x. Formally, we define the *effective region* (ER) of m_l on x as the set of input features modified by m_l, denoted as $\phi_l(x) \doteq \left\{ i : [q(m_l, x)]_{(i)} \neq x_{(i)} \right\}$, where subscript (i) indexes over individual features (e.g. pixels). The blank manipulation has empty ER, $\phi_\varnothing = \varnothing$.

For $(x, \widehat{y}) \sim \mathbb{P}_{X,\widehat{Y}}$, we choose a manipulation m_l from $\mathcal{M}$ according to $\widehat{y}$ and modify the input as $\widehat{x} = q(m_l, x)$. The label-dependent choice can be deterministic or stochastic. We denote the new data distribution as $\mathbb{P}_{\widehat{X},\widehat{Y}}$. With appropriate choice of manipulation, $\mathbb{P}_{\widehat{X},\widehat{Y}}$ can satisfy $\widehat{p}^* \doteq \sup_{\widehat{x},\widehat{y}} \mathbb{P}_{\widehat{Y}|\widehat{X}}(\widehat{y}|\widehat{x}) > p^*$. For example, $\widehat{p}^* = 1$ when a watermark is applied only to the positive class.

Whenever a model trained on $(\widehat{\mathcal{X}}, \widehat{\mathcal{Y}})$ achieves expected accuracy $p^{(3)} > p^*$, it is guaranteed to rely on the knowledge of manipulation, which is solely confined within the joint effective region $\phi_\cup(x) \doteq \cup_l \phi_l(x)$. This gives us a straightforward, quantitative sanity check for feature attribution methods: they should recognize the attribution contribution inside $\phi_\cup(x)$. For our example, since only the watermark is applied to one class, $\phi_\cup$ consists of the watermarked region.

On finite test sets, a classifier can achieve an accuracy $p > p^*$ without using the manipulation, due to stochasticity in label reassignment. However, for test set size N, the probability of this classifier achieving of p or higher, when the *expected* accuracy is bounded by p^*, is at most $\sum_{n=\lfloor pN \rfloor}^{N} \mathrm{Binom}(n; N, p^*)$, which vanishes quickly with increasing N and p.

Remark. It is crucial that we consider the *joint* effective region over all manipulations for attribution values, since a model could use the absence of manipulation as a legitimate basis for decision. For example, consider an image dataset, with each image having a watermark either on the top or bottom edge correlated with the positive or negative label respectively. A model could predict the negative class based on the *absence* of a watermark on the top edge. In this case, the correct attribution to the top edge is within the joint ER but *not* within the bottom watermark ER. Current evaluations [2, 37] often omit this possibility by using the ER of *only* the manipulation applied to the target class rather than the union of all possible ERs for every class, potentially rejecting correct attributions.

In next three sections, we experimentally compare attribution values of three types of models—saliency maps, attention mechanisms, and rationale models—to those expected by the desiderata. Through this analysis, we identify their deficiencies and give recommendations for improvements.

5 Evaluating Image Saliency Maps

For these experiments, we simulate a common scenario where a model seemingly achieves "superhuman" performance on some hard image classification task, only for us to later find out that it exploits some image artifacts which are accidentally leaked in during the data collection process. We evaluate the extent to which several different saliency map attribution methods can identify such artifacts.

Dataset: We curated our own dataset on bird species identification for commonly confused classes. It is motivated by CUB-200-2011 [34], with many high-resolution images for fine-grained classification. We identified a ResNet-34 model's [13] four most mistaken class pairs in CUB-200-2011 and scraped Flickr for 1,200 new images for each of these classes. We center-cropped all images to 224×224 and mean-variance normalized using ImageNet statistics [10]. We split the 1,200 images per class into train/validation/test sets of 1000/100/100 images. Fig. 2 presents sample images, the confusion matrix for a ResNet-34 model trained on this data, and example saliency maps for a correct prediction.

Input Manipulations: We select five image manipulations which represent artifacts that could be accidentally introduced in a dataset collection process: blurring, brightness change, hue change, pixel noise, and watermark. Details and examples of each manipulation are in App. C.1.

Saliency Maps: We evaluate 5 standard saliency map methods: Gradient [32], SmoothGrad [33], GradCAM [29], LIME [26], and SHAP [22]. These methods are described in detail in App. C.2.

Experiments: We choose pairs of easily confused species (e.g. common tern and Forester's tern) to simulate a hard task which is made easier through the presence of artifacts. All experiments are binary classifications. Sec. 5.4 also uses pairs of distinct species (e.g. common tern and fish crow).

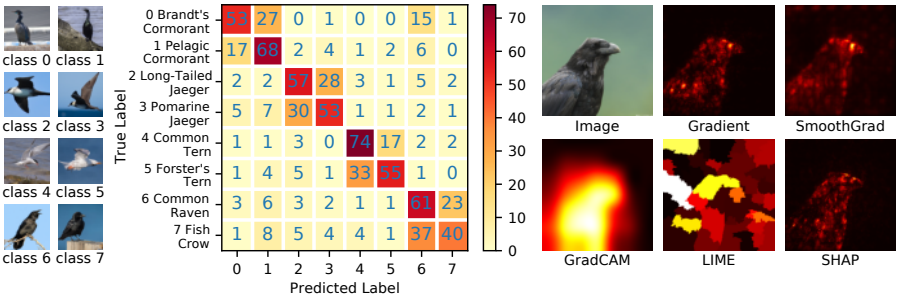

Figure 2: Left: Dataset samples. Middle: Test set confusion matrix of a ResNet-34 model. Right: Examples of five different saliency maps for a correct prediction (Fish Crow).

Metric: We study the attribution percentage assigned to the joint effective region $\text{Attr}\%(\phi_\cup)$. We calculate $\%\text{Attr}$ for images in the test set, and report the average separately for the two classes.

5.1 Attr% for Various Attribution Methods and Manipulations

Question: How well do saliency maps give attribution to the ground truth for (near-)perfect models?

Setup: We trained 100 models, each on a random pair of similar species and a random manipulation type. We reassign labels with $r = 0.5$ (i.e., expecting random accuracy), and apply the manipulation to images of the positive post-reassignment class, leaving the negative class images unchanged.

Expectation: Due to $r = 0.5$, *only* the manipulation is correlated with the label. A close-to-perfect performance thus indicates that the model relies almost exclusively on features inside $\phi_\cup$. Thus, we should expect $\text{Attr}\%(F_{\phi_\cup})$ to get very close to 1.0, regardless of the size of $\phi_\cup$.

Results: 70% of all runs achieve test accuracy of over 95%[1]. We compute $\text{Attr}\%(F_{\phi_\cup})$ for these models. Since $\%\text{Attr}$ naturally depends on the size of $\phi_\cup$ (e.g. $\phi_\cup$ of the entire image implies $\text{Attr}\% = 1$), we plot them against $\%\text{ER}$, defined as the size of $\phi_\cup$ as a fraction of image size. Fig. 3 (left) shows these two values for some methods and manipulations (complete results in App. C.3).

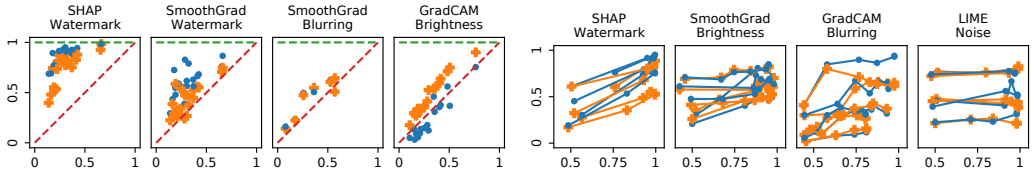

Figure 3: Left: $\%\text{Attr}$ (y-axis) vs. $\%\text{ER}$ (x-axis), complete results in Fig. 10 of App. C.3. Right: $\%\text{Attr}$ (y-axis) vs. test accuracy (x-axis), complete results in Fig. 13 of App. C.6. Blue circles and orange crosses are for images with and without the manipulation. Green horizontal line indicates the saliency map with $\text{Attr}\% = 1$ and red diagonal line indicates a random saliency map.

The models successfully learned all manipulation types, as demonstrated by the high test accuracies; however, none of the methods consistently scores $\%\text{Attr} \approx 1$, as expected. Further, these manipulations are not all equally well detected by all saliency maps. While SHAP performs the best ($\%\text{Attr} = 69\%$ at $\%\text{ER} = 40\%$ on average), it is still hard to trust "in the wild" since its efficacy strongly depends on manipulation type. The *presence* of a watermark is consistently better detected than its *absence*, likely because the model implicitly localizes objects (i.e. the watermark) [7] and predicts a default negative class if it fails to do so. It is also easier for perturbation-based methods such as LIME to "hide" it when present than to "construct" it when absent. Thus, saliency maps may mislead people about the true reason for a negative prediction, and better methods to convey absence are needed.

5.2 Attribution vs. Test Accuracy

Question: How does $\%\text{Attr}$ change as the model's test accuracy increases during training?

Setup: We use the the same setup as Sec. 5.1.

[1]Note that since the model is not 100% accurate, it could be "distracted" by features outside of F_C. However, such distraction is small, accounting for at most 5% of errors, and it is much more important for users to understand that the over 95% accuracy comes solely from F_C, and thus requiring $\text{Attr}\% \approx 1$ is reasonable.

Expectation: As the test accuracy increases, the model must rely more and more on knowledge of manipulation. As a result, we should expect $\text{Attr}\%(F_{\phi_\cup})$ to also increase.

Results: For the training run of each model, we compute $\text{Attr}\%(F_{\phi_\cup})$ for models during intermediate epochs with various test accuracy scores. Fig. 3 (right) plots the lines representing the progress of $\%\text{Attr}$ vs. test accuracy (complete results in App. C.6). SHAP with watermark shows the most consistent and expected increase in $\%\text{Attr}$ with test accuracy. For other saliency maps and feature types, the trend is very mild or noisy, suggesting that the attribution method fails to recognize model's increasing reliance on the manipulation in making increasingly accurate predictions.

5.3 Attribution vs. Manipulation Visibility

Question: How well can saliency maps recognize manipulations of different visibility levels?

Setup: We conduct 100 runs, with 20 per manipulation. We further group the 20 runs into 4 groups, with 5 runs in a group using the same manipulation type and effective region but varying degrees of visibility, detailed in App. C.4. For example, the visibility for a watermark refers to its font size. As before, the labels are reassigned with $r = 0.5$ and manipulations applied to the positive class.

Expectation: A good saliency map should not be affected by manipulation visibility, as long as the model is objectively using it. However, different saliency maps may be better suited to detect more or less visible manipulations. For example, a less visible manipulation may be ignored by the segmentation algorithm used by LIME, while inducing larger gradients in the decision space. From a practical perspective, it is more important for them to identify less visible manipulations, since the others are more likely to be detected by human eyes as well during dataset visualization.

Results: Fig. 4 (left) plots each group of five runs as a line, with visibility level on the x-axis and $\%\text{Attr}$ on the y-axis (complete results in App. C.4). Except for SHAP on watermark, other methods do not show consistent trend of $\%\text{Attr}$ increasing with visibility. While SHAP is more effective on more visible manipulations, we rely most on interpretability methods to uncover precisely the less visible manipulations or artifacts. Unfortunately, none of the methods could satisfy this requirement.

5.4 Attribution vs. Original Feature Correlation

Question: How does the attribution on the manipulation change if the label reassignment is correlated with the original input features to higher or lower degrees (i.e. $r \in [0.5, 1.0]$)?

Setup: For each manipulation, we vary the label reassignment parameter $r \in \{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. For each r, we train four models on four class pairs: two of similar species (e.g. class 4 vs. 5 in Fig. 2) and two of distinct ones (e.g. class 5 vs. 6). These variations total $5 \times 6 \times 4 = 120$ runs.

Expectation: For $r > 0.5$, there is no standard definition of the attribution value on the original image features and the manipulations, as any decreasing trend of $\%\text{Attr}$ with increasing r is reasonable. However, the Shapley value [27] is a commonly used axiomatic definition for feature attributions. We denote the set of features inside the effective region as F_M, for manipulated features, and that outside as F_O, for original features. For (near-)perfect classifier with $p \approx 1$, we have their Shapley values (normalized to have a sum of 1) satisfy $\bar{v}(F_M) \geq 1.5 - r$, and $\bar{v}(F_O) \leq r - 0.5$, as derived in App. C.5. In addition, for distinct class pairs, the model can better utilize the more distinct original image features, resulting in lower attribution $\bar{v}(F_M)$ on manipulated features.

Results: All models achieve test accuracy of over 95%. Fig. 4 (right) plots $\%\text{Attr}$ vs. r (complete results in App. C.5). Solid lines represent runs with a similar species pair, and dashed lines represent runs with a distinct species pair. The green shaded area represents the area of $\bar{v}(F_M) \geq 1.5 - r$, the Shapley value range at $p = 1$. In other words, values within the green shade are consistent with the

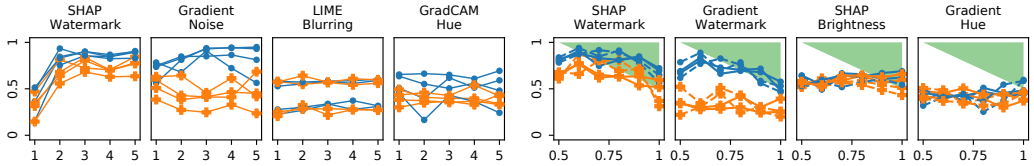

Figure 4: Left: $\%\text{Attr}$ vs. feature visibility , complete results in Fig. 11 of App. C.4. Right: $\%\text{Attr}$ vs. r, complete results in Fig. 12 of App. C.5. Blue/orange colors indicate feature presence/absence. Solid/dashed lines represent similar/distinct species pairs. Green shades represent $\%\text{Attr} \geq 1.5 - r$.

Shapley axioms, and those outside are not. Intuitively, for r close to 0.5, the correlation between F_O and the label is very weak, and the theoretical Shapley values should mostly concentrate on F_M for the (near-)perfect model, with $\%\mathrm{Attr}$ close to 1. As r increases, the model can choose to rely more heavily on F_O for its prediction, resulting in larger allowable ranges of $\%\mathrm{Attr}$.

For watermark manipulation, SHAP shows clear decreases in attribution value as r increases, while gradient also tracks the predicted range, but only for the class with the manipulation. This trend is not seen in other feature types, even if computed by SHAP which uses approximation, and shows relative insensitivity with respect to r. In addition, there does not seem to be a clear difference in attribution values for similar vs. distinct species pairs. Since the Shapley value *axiom* is commonly accepted as generally reasonable, it is concerning that many saliency maps are inconsistent with it, and important to better understand the underlying axiomatic assumptions (if any) made by each saliency map.

5.5 Discussion

Arguably one of the most important application of model explanation is to detect any usage of spurious correlations, but our results cast doubt on this capability from various aspects. Therefore, we recommend that, before analyzing the actual model, developers should first train models that are guaranteed to use certain known features, and "dry run" the planned interpretability methods on them to make sure that these features are indeed highlighted.

6 Evaluating Text Attention

It is known that certain non-semantic features can heavily influence model prediction (e.g. email header in the Newsgroup dataset [26]). We study whether the word attention score can saliently highlight any potential usage of such non-semantic features. Details of the implemented dot-product attention mechanism are presented in App. D.1.

Dataset: We use the BeerAdvocate dataset [23] as the basis for modification and further select 12,000 reviews split into 10,000/1,000/1,000 train/validation/test sets, without the label information (which is set differently for each experiment).

Metric: The introduced manipulation changes specific words according to the (reassigned) label. The metric is $\%\mathrm{Attr}$ defined on the set of target words (i.e. effective region).

6.1 Highly Obvious Correlating Features

Question: How well can attention scores focus on highly obvious manipulations?

Setup: From our filtered dataset, we first randomly assign binary labels. For the positive reviews, we change all the article words (*a / an / the*) to "the", and for the negative reviews, we change these to "a". Thus, the articles are the only words correlated with the labels and constitute the effective region.

Expectation: Attention of (near-)perfect models should have $\%\mathrm{Attr} \approx 1$ to be valid attributions.

Results: The model achieves over 97% accuracy. Across the test set, $\%\mathrm{Attr}$ on article words is 8.6%. Considering that articles are 7.9% of all words, this is better than random, albeit barely. Fig. 5 visualizes the attention distribution for two reviews, with additional results in Fig. 14 of App. D.2. Each bars represent weights of words in the review. Green bars represent non-articles and orange bars represent articles. The attention on article words either does not stand out from the rest, or only stands out *relative* to their neighbors. Generally, it does not consistently highlight important features.

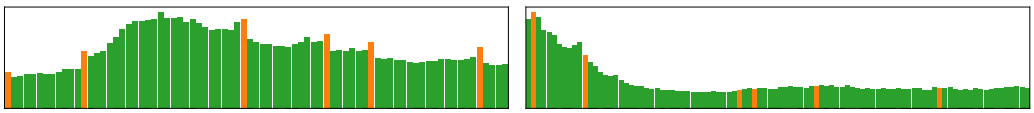

Figure 5: Attention values for each token in two reviews. Orange/green bars represent articles/non-articles. More in Fig. 14 of App. D.2.

6.2 Misleading Non-Correlating Features

Question: When some features are known to not correlate with the label but are very similar to correlating ones, do attention scores also focus on these non-correlating ones?

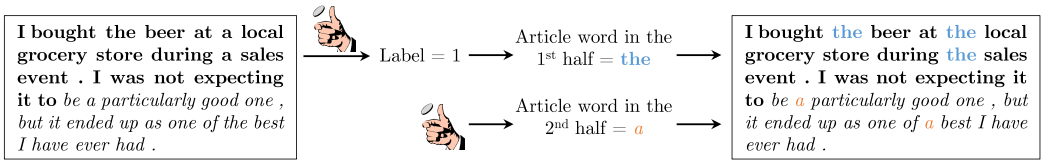

Figure 6: The process to build *CN* dataset for the experiment in Sec. 7.2. First, a review is split into two halves at the midpoint, shown in **bold** and *italics*. Then a label is randomly sampled and assigned to the review. Depending on the label, the articles in the first half are changed to "a" or "the". They are called *correlating articles*. Then an article word is randomly chosen for the second half, and all articles in the second half are changed to that word. They are called *non-correlating articles*. For *NC* dataset, the roles of two halves are switched.

Setup: Again from our filtered dataset, we applied two similar manipulations, with only one of them is correlated with the (reassigned) label. Fig. 6 details the construction of two datasets, *CN* and *NC*.

Expectation: Same as above. In particular, non-correlating articles should *not* be attended to.

Results: The models on both datasets achieve over 97% accuracy. Fig. 7 presents attention visualization, with more in Fig. 15 of App. D.3. The two models show very different behaviors. The *CN* model exclusively focuses attention on correlating articles, while the *NC* model behaves similarly to the previous experiment.

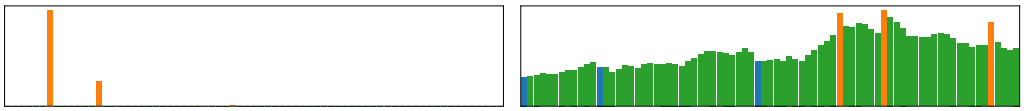

Figure 7: Attention scores for one review in *CN* (left) and *NC* (right) dataset. Orange/blue/green bars represent corr. articles/non-corr. articles/other words. More in Fig. 15 of App. D.3.

Observing the large variation of behaviors, we further trained the three models ten more times to see if any consistent attention pattern exists. All models achieve over 97% accuracy. Tab. 1 presents the mean and standard deviation statistics for the 11 runs. The clean attention pattern by the *CN* model does not persist, and the model sometimes assigns higher than random weights on non-correlating articles, especially for the NC dataset. These results further suggests that attention weights cannot be readily and reliably interpreted as attributions without further validation.

Dataset	Corr. Articles	Non-Corr. Articles	Other Words
Article	$10.3\% \pm 2.4\%$ | 7.9%	NA	$89.7\% \pm 2.4\%$ | 92.1%
CN	$15.9\% \pm 25.7\%$ | 4.1%	$5.9\% \pm 4.0\%$ | 3.8%	$78.2\% \pm 24.7\%$ | 92.1%
NC	$12.0\% \pm 8.4\%$ | 3.8%	$12.6\% \pm 9.3\%$ | 4.1%	$75.4\% \pm 16.7\%$ | 92.1%

Table 1: Attention attribution statistics over 11 training runs, in the format of "$\text{mean}(\%\text{Attr}) \pm \text{stdev}(\%\text{Attr}) \mid \text{word frequency}$". The "Article" dataset is the one used in Sec. 6.1.

6.3 Discussion

Attention is undoubtedly useful as a building block in neural networks, but their *interpretation* as attribution is disputed. Past studies proposed various, and sometimes conflicting, criteria for such interpretation [17, 25, 35], but their correctness is unclear. In our studies, the answer is mostly negative: for most training runs, the attention weights on correlating features at best only stand out *locally*, easily overwhelmed by larger global variations. Therefore, we recommend that future proposals should first be calibrated with ground truth in a controlled setting.

7 Evaluating Text Rationales

In this section, we conducted the same two experiments above (and omit the **Question** and **Setup** descriptions), but for two rationale models, a reinforcement learning (RL) model [21] and a continuous relaxation (CR) model [6]. In the original forms, both models regularize the rationale length and continuity. In our experiments, rather than regularizing the length, we train the models to produce rationales that match a target selection rate $\%\text{Sel}$. For a mini-batch of B examples, we use

$\lambda \cdot \left| \sum_{i=1}^{B} \text{len}(\text{rationale}_i) / \sum_{i=1}^{B} \text{len}(\text{review}_i) - \text{Sel}\% \right|$, where $\lambda > 0$ is the regularization strength. Incidentally, we also found that the training is much more stable with this regularization, especially for the RL model. Additionally, we removed the discontinuity penalty, because ground truth rationales in our experiments are not continuous. We use precision and recall metrics as defined in Sec. 3.

7.1 Highly Obvious Manipulations

Expectation: A necessary condition for a non-misleading rationale is that it should include at least one article word, regardless of selection rate. However, a desirable property of rationale is comprehensiveness [38]: selecting as many article words as possible. Thus, a good rationale model should have high precision when selection rate is low and high recall when selection rate is high.

Results: We trained models with $\%\text{Sel} \in \{0.07, 0.09, 0.11, 0.13, 0.15\}$, all with over 97% accuracy. We evaluate precision and recall of the trained models and plot them in Fig. 8 (left) according to the actual rationale selection rate, $\%\text{Sel}$, on the test set. Blue and orange markers are for the RL and CR models respectively. The two green lines show two optimality notions: the solid line enforces aggregate $\%\text{Sel}$ for the test set, and the dashed line enforces $\%\text{Sel}$ individually per review.

Except for the CR model at the lowest $\%\text{Sel}$, all others achieve near-perfect rationale selection on both the precision and and the recall metrics. In particular, they are nearly dataset-wide optimal, due to $\%\text{Sel}$ regularization done at the mini-batch level. The "faulty" CR model tends to select the first few words regardless, as shown in Fig. 16 of App. E.1, but still selects some article words.

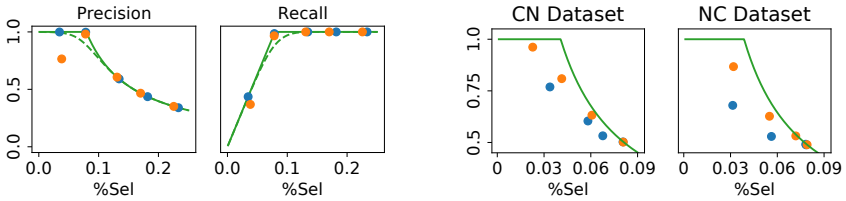

Figure 8: Left: Precision and recall for two rationale models in Sec. 7.1. Right: Precision at different $\%\text{Sel}$ for models in Sec. 7.2. RL model is in blue and CR in orange. The solid and dashed green lines show optimal metric values when $\%\text{Sel}$ is enforced at dataset- and sentence-level.

7.2 Misleading Non-Correlating Features

Expectation: Similar to the previous experiment, at least one correlating article word needs to be selected. However, selection of non-correlating articles is arguably more misleading than selection of other non-article words, because it suggests that these non-correlating articles also influence the prediction, even though the classifier simply learns to ignore them.

Results: We trained models with $\%\text{Sel} \in \{0.03, 0.05, 0.07, 0.09\}$, all with over 97% accuracy. Fig. 8 (right) plots the precision of correlating articles for the two datasets, as well as the dataset-wide optimal value. We found the rationales consist of almost exclusively article words (Fig. 17 of App. E.2). However, especially for the RL model, some correlating articles are missed while at the same time non-correlating ones are selected, resulting in markedly less than optimal precision.

7.3 Discussion

The structure of rationale models guarantees that the classifier objectively relies on the rationale features for prediction, but this does not mean that the rationale is contains all the correlating features F_C. Specifically, it could highlight F_C only barely, while including lots of non-correlating F_N (and, in particular, misleading words such as the non-correlating articles). Indeed, our results show that rationale methods are prone to selecting misleading non-correlating features, which obfuscates the model's reasoning process by giving *more* but unnecessary information to the human. The problem is more severe with RL training, possibly due to the known difficulty with REINFORCE [36]. Post-processing methods could be developed to further prune rationales to mitigate this problem.

8 Conclusion

As feature attribution methods are increasingly deployed to ensure the correctness of high-stakes systems, it is crucial to ensure these methods work correctly. Current evaluations fall short—primarily due to a lack of clearly defined ground truth. Rather than evaluating explanations for models trained

on natural datasets, we propose "unit tests" to assess whether feature attribution methods are able to uncover ground truth information about how models trained on carefully-modified, semi-natural datasets make decisions. Surprisingly, none of our evaluated methods across vision and text domains achieve totally satisfactory performance, and we point out various future directions in Sec. 5.5, 6.3 and 7.3 for improving attribution methods. While our ground truth consists of known dataset artifacts rather than "natural" features, similar artifacts have been found in popular image and text datasets, and have been known to affect model decisions [e.g. 12, 16, 19, 26]. Moreover, any limitations of attribution methods should be carefully studied, and practitioners should be informed accordingly.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in neural information processing systems*, volume 31, pages 9505–9515, 2018.

[2] Julius Adebayo, Michael Muelly, Ilaria Liccardi, and Been Kim. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

[3] Leila Arras, Ahmed Osman, Klaus-Robert Müller, and Wojciech Samek. Evaluating recurrent neural network explanations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 113–126, 2019.

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

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, 2019.

[7] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[8] Ian Covert, Scott Lundberg, and Su-In Lee. Understanding global feature contributions through additive importance measures. *arXiv preprint arXiv:2004.00668*, 2020.

[9] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*, 2020.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

[12] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758*, 2018.

[15] Aya Abdelsalam Ismail, Mohamed Gunady, Héctor Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. *arXiv preprint arXiv:2010.13924*, 2020.

[16] Sarah Jabbour, David Fouhey, Ella Kazerooni, Michael W Sjoding, and Jenna Wiens. Deep learning applied to chest x-rays: Exploiting and preventing shortcuts. In *Machine Learning for Healthcare Conference*, pages 750–782. PMLR, 2020.

[17] Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

[18] Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C Wallace. Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*, 2020.

[19] Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

[20] Joon Sik Kim, Gregory Plumb, and Ameet Talwalkar. Sanity simulations for saliency methods. *arXiv preprint arXiv:2105.06506*, 2021.

[21] Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, 2016.

[22] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[23] Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pages 1020–1025. IEEE, 2012.

[24] Milad Mostavi, Yu-Chiao Chiu, Yufei Huang, and Yidong Chen. Convolutional neural network models for cancer type prediction based on gene expression. *BMC medical genomics*, 13:1–13, 2020.

[25] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C Lipton. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, 2020.

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[27] Alvin E Roth. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[28] 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, 2017. doi: 10.1109/TNNLS.2016.2599820.

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

[30] Li Shen, Laurie R Margolies, Joseph H Rothstein, Eugene Fluder, Russell McBride, and Weiva Sieh. Deep learning to improve breast cancer detection on screening mammography. *Scientific reports*, 9(1):1–12, 2019.

[31] Ke Si, Ying Xue, Xiazhen Yu, Xinpei Zhu, Qinghai Li, Wei Gong, Tingbo Liang, and Shumin Duan. Fully end-to-end deep-learning-based diagnosis of pancreatic tumors. *Theranostics*, 11 (4):1982, 2021.

[32] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

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

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[35] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

[36] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[37] Mengjiao Yang and Been Kim. Benchmarking attribution methods with relative feature importance. *arXiv preprint arXiv:1907.09701*, 2019.

[38] Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4085–4094, 2019.

[39] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

A Additional Related Work on Feature Attribution

Feature attribution methods assign attribution scores to input features, the absolute value of which informally represents their importance to the model prediction or performance.

Saliency maps explain an image I by producing S of the same size, where $S_{h,w}$ indicates the contribution of pixel $I_{h,w}$, which has been interpreted as sensitivity [32], relevance [4], local influence [26], Shapley values [22], or filter activations [29].

Attention mechanisms [5] were originally proposed to better retain sequential information. Recently they have been used as attribution values, but their their validity is under debate with different and inconsistent criteria being proposed [17, 25, 35].

Rationale models [6, 18, 21] are inherently interpretable models for text classification with a two-stage pipeline: a selector extracts a rationale (i.e. input words), and a classifier makes a prediction based on it. The selected rationales are often regularized to be succinct and continuous.

A.1 Detailed Comparison with Adebayo et al. [2] and Yang and Kim [37]

We provide a more in-depth comparison of our work with those by Adebayo et al. [2] and Yang and Kim [37]. In terms of similarity, the high-level idea is similar: for a natural dataset, we do not know individual feature contribution, but generally highly correlated and easily discriminative features should have high contribution, and we realize this notion by injecting such features directly. All three works can be seen as operationalizations of this idea.

However, this high-level idea needs two caveats, which set our paper apart. First, features that we think to be "easily discriminative" may not be be considered as such by neural nets. For example, Geirhos et al. [11] showed that they have particular inclination toward textures. Thus, we can't really trust them to actually pick up and use our introduced features, unless that we are assured that focusing on other features cannot achieve the performance that the model is achieving now. In this aspect, we concretely demonstrated that network trained by Adebayo et al. [2] can achieve good performance when using the "other features" exclusively. Yang and Kim [37] demonstrated this principle, but used out-of-distribution data, so it is not clear whether the failure is due to achieve good performnace is really due to the network indeed ignoring the "other features" or due to instability of out-of-distribution extrapolation (as discussed in our original response).

The second caveat is that the "other features" cannot have any information on the injected features. For example, in Fig. 4 (Page 6) of the extended version by Adebayo et al. [2] at `https://arxiv.org/pdf/2011.05429.pdf`, the saliency map that perfectly crops out the foreground dog shape could imply that the network actually uses the contour of the dog, which is inconsistent with their target conclusion that "foreground doesn't matter in highly correlated background dataset". In other words, background with a dog contour cropped out is very informative to the fact that the foreground is a dog. The similar cropping procedure is used by Yang and Kim [37] as well. Another loophole is discussed in the Remark of the paper, where the lack of one feature at one place could imply the presence of another feature at another place. By comparison, in our work, we define the joint effective region such that all injected features are contained within, and then we are assured that the features outside of the region absolutely cannot contribute to the high performance, and then evaluate $\%\mathrm{Attr}$ for that region.

Finally, our framing is more general. We start with the goal of evaluating feature attribution in general, and proposes our domain-agnostic framework in Fig. 1 and Sec. 3 and 4. Thus, it would be straightforward to instantiate our idea to other domains such as graph, speech, or time-series data. By comparison, both Adebayo et al. [2] and Yang and Kim [37] focus on the foreground/background patch setting, and introduces necessary concepts under this context. We also proposed ways to evaluate feature-selection style attributions, more common for text models at the end of Sec. 3.

B Details on the General Dataset Modification Procedure

In this section, we present the generalization of the label-reassignment to multi-class settings. We model it by a reassignment matrix $R \in \mathbb{R}^{K\times K}$. According to this matrix, the label reassignment process assigns a new label $\widehat{y}$ based on the original label y with probability $R_{y,\widehat{y}}$. The expected accuracy $p^{(2)}$ of *any* classifier is now bounded by $p^{(2)} \leq p^* = \max_{i,j} R_{i,j}$.

C Additional Details and Results for Saliency Map Evaluations

C.1 Manipulation Types

We consider five image manipulation types. These manipulations are designed to simulate possible image artifacts, which an undesirable model may rely on to make decisions. Each manipulation has parameters which define the effective region and the visibility level of the manipulation effect. Some of the manipulation effects are technically stochastic, such as a watermark being placed in a random position, but the effective region captures the localized manipulation effect of all possible random instantiations. The five manipulations are described below, with examples of each manipulation and their associated effective regions shown in Fig. 9.

- **Peripheral blurring** applies a Gaussian filter to the part of the image outside of a certain radius. It is parametrized by
  - the radius of the *unaffected* part; and
  - the standard deviation of the Gaussian blurring filter.

  Blurring could be due to either camera in motion or an intentional, artistic post-processing effect to highlight the main subject of the image.
- **Central brightness shift** gradually changes the brightness in the hue-saturation-brightness (HSB) space inside a certain radius, with maximal change in the center. For our experiments, the brightness change is negative, meaning that the center is dimmed. It is parametrized by
  - the radius of the dimmed region; and
  - the magnitude of the brightness shift at the center.

  Brightness shift could be due to times of the day, or the use of artificial light to illuminate the subject.
- **Striped hue shift** modifies the hue (i.e. color) value of a vertical stripe in the image. From top to bottom in the stripe, the hue value is first increased and then decreased in a sinusoidal pattern. It is parametrized by
  - the upper position of the stripe;
  - the lower position of the stripe, with the width of the stripe being (upper - lower);
  - the magnitude of sinusoidal pattern.

  Hue shift could be due to errors in conversion of different color space encodings, which may result in color loss or distortion.
- **Striped noise** randomly changes pixels inside a vertical stripe to a uniformly random RGB value. It is parametrized by
  - the upper position of the stripe;
  - the lower position of the stripe, with the width of the stripe being (upper - lower);
  - the probability that each pixel is replaced.

  Pixel noise could be due to lossy compression or data loss during transmission.
- **Watermark** overlays a text reading "IMGxxxx", where "xxxx" are four random digits, to a random location inside a rectangular region. "IMG" is written in white and the digits are written in black. It is parametrized by
  - the upper-left coordinate of the rectangular region;
  - the lower-right coordinate of the rectangular region;
  - the font size of the watermark text.

  Watermark is a commonly employed technique to attribute the author/organization of the image.

Normally, none of them should be expected to correlate with the label. However, especially with image scraping on the web and crowdsourced dataset construction, it is possible that some spurious correlations leak into the final dataset.

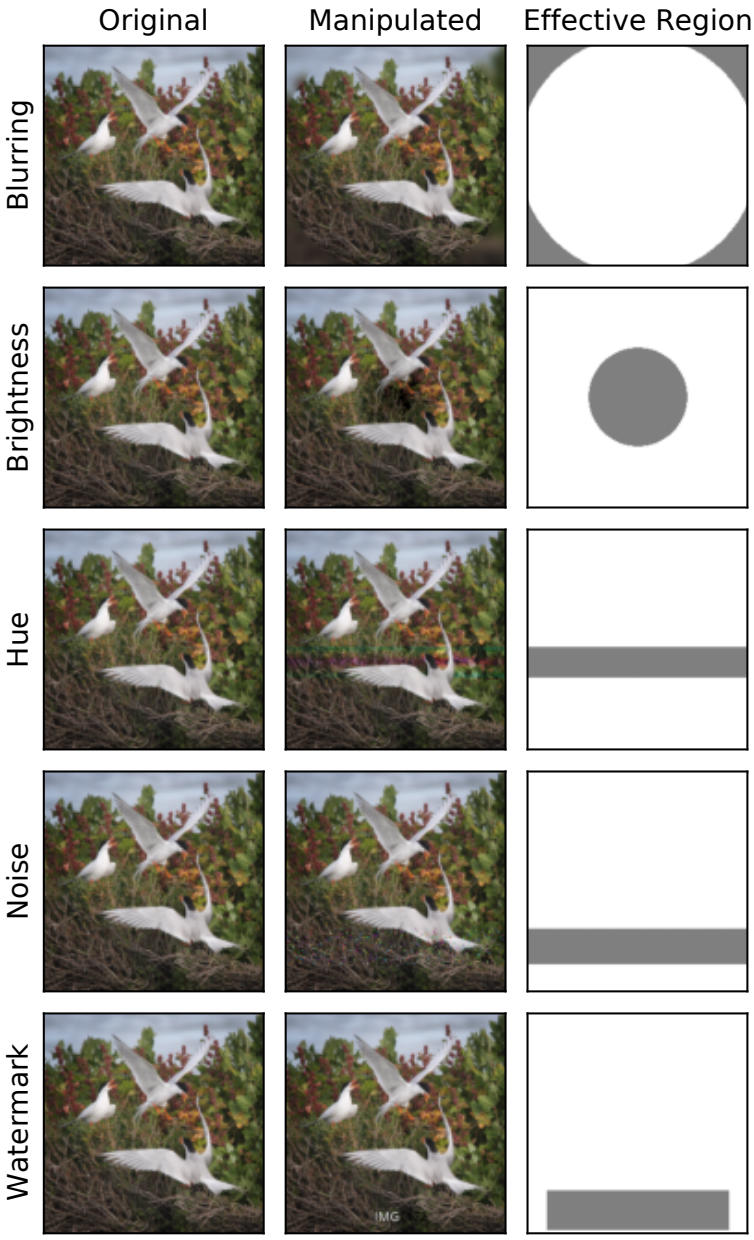

Figure 9: Examples of the five manipulations applied to an image, along with the respective effective regions (ER) shown in gray.

C.2 Saliency Map Methods

We consider five saliency map methods.

- **Gradient** [32] computes the gradient of the logit for the predicted class with respect to the input image. The three channels of gradient are summed up in absolute value to get a single channel.
- **SmoothGrad** [33] averages the gradients on 50 copies of the input image I, each injected with independent Gaussian noise with with $\mu = 0$ and $\sigma = 0.15 \cdot (\max I - \min I)$, where $\max I$ and $\min I$ are the maximal and minimal pixel values of the image.
- **GradCAM** [29] computes a saliency map from convolution filter responses. Since we use the fully convolutional ResNet-34, this method reduces to the class activation mapping (CAM) [39].
- **LIME** [26] performs a linear regression using super-pixels of the input image. The absolute values of the coefficients are used to derive the saliency map. We use the default implementation of `lime.lime_ image.LimeImageExplainer` with the quickshift clustering as the super-pixel segmentation algorithm.
- **SHAP** [22] uses the idea of Shapley value [27] for attribution. We use the GradientSHAP instantiation with the default setting of `shap.GradientExplainer`. We us the entire test set of 200 examples as the "background" data.

C.3 Attribution vs. Effective Region Size

Fig. 10 shows %Attr vs %ER for all pairs of saliency maps and manipulations.

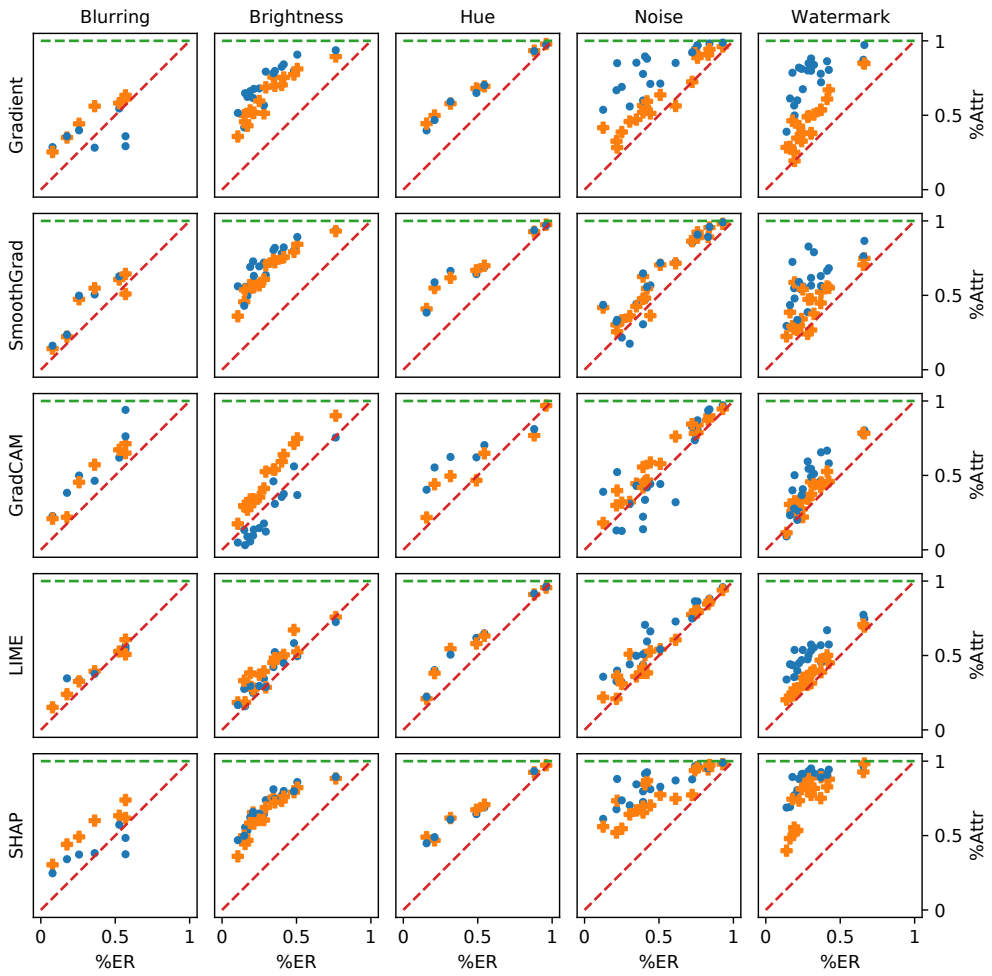

Figure 10: %Attr vs %ER for all pairs of saliency maps and manipulations.

C.4 Attribution vs. Manipulation Visibility

Fig. 11 shows %Attr vs manipulation visibility for all pairs of saliency maps and manipulations.

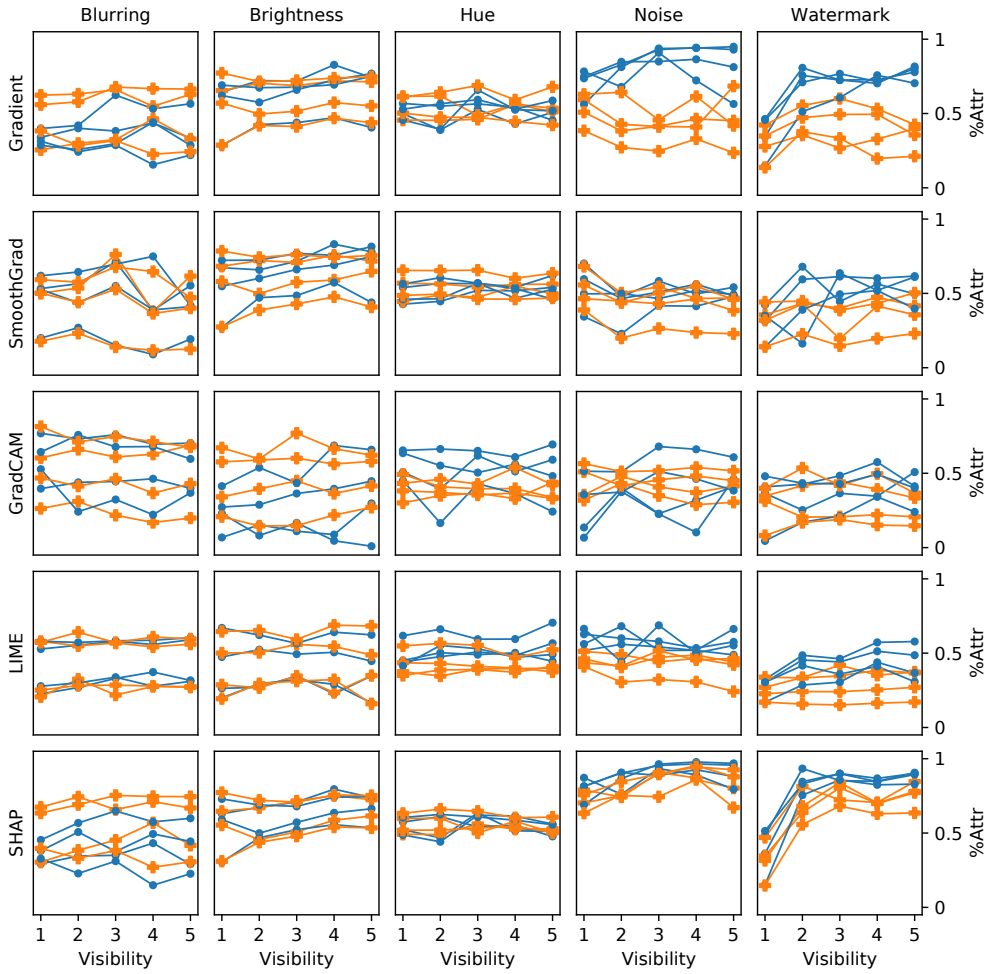

Figure 11: %Attr vs manipulation visibility for all pairs of saliency maps and manipulations.

We define the five visibility levels for each manipulation type as below. Note that for fair comparison of %Attr at different visibility levels, it is crucial that the effective regions are independent of the visibility, which is satisfied in all manipulation types below.

- **Blurring:** The visibility level is defined as the Gaussian blur standard deviation, with values of $\{2, 4, 6, 8, 10\}$ pixels, from least visible to most.
- **Brightness:** The visibility level is defined as the magnitude of the brightness shift, with values of $\{0.1, 0.15, 0.2, 0.25, 0.3\}$ brightness component of the color (in the range of [0, 1]), from least visible to most.
- **Hue:** The visibility is defined as the magnitude of the hue shift, with values of $\{0.05, 0.1, 0.15, 0.2, 0.25\}$ hue component of the color (in the range of [0, 1]), from least visible to most.
- **Noise:** The visibility is defined as the probability that a pixel is replaced by a random value, with values of $\{0.02, 0.04, 0.06, 0.08, 0.1\}$, from least visible to most.
- **Watermark:** The visibility is defined as the font size of the watermark, with values of $\{7, 9, 11, 13, 15\}$ pixels, from least visible to most.

C.5 Attribution vs. Original Feature Correlation

Fig. 12 shows %Attr vs the label reassignment parameter r for all pairs of saliency maps and manipulations.

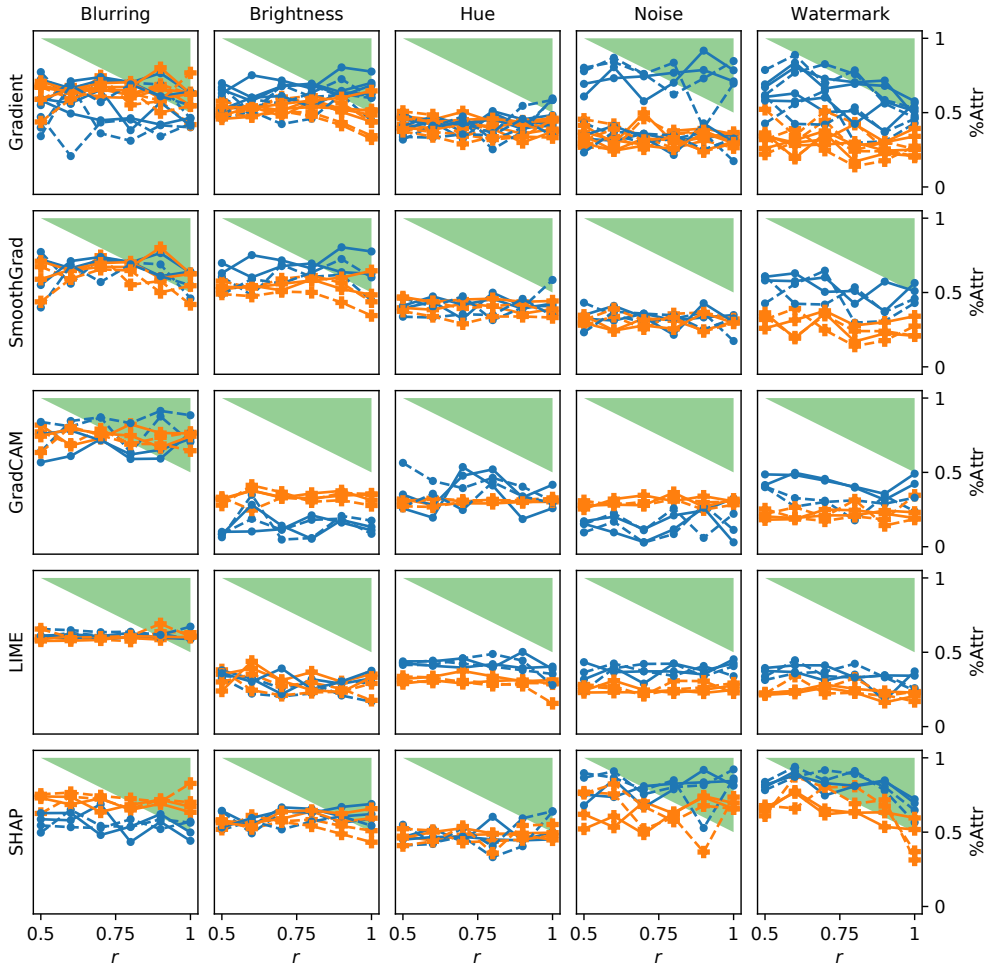

Figure 12: %Attr vs label reassignment parameter r for all pairs of saliency maps and manipulations.

As explained in Sec. 5.4, $a(F)$ represents the expected accuracy of the model when given only feature F. Note that this value should *not* be calculated as the model accuracy on images with every pixel but F being blacked out, because such images are out of distribution where the model may exhibit unreasonable behaviors (similar to the discussion raised by Hooker et al. [14]).

Instead, the suppression of information beyond F can be understood as an inability for the model to distinguish inputs that agree on F. This leads to the following process of simulating such a prediction. First, let $\mathbb{P}_{X,Y|F=f}$ be the data distribution conditioned on $F = f$. Since all features other than F are suppressed, the model cannot further distinguish two inputs $x, x' \sim \mathbb{P}_{X|F=f}$. As a result, the expected accuracy can be computed by comparing the model's prediction on x against the ground truth label on x'. Then we take the expectation of this accuracy according to different values of $f \sim \mathbb{P}_F$, where $\mathbb{P}_F$ is the marginal distribution of F. Formally, for the model prediction function $g : \mathcal{X} \to \mathcal{Y}$, we have

$$a(F) = \mathbb{E}_{f\sim\mathbb{P}_F}\left[\mathbb{E}_{x,y\sim\mathbb{P}_{X,Y|F=f}}\left[\mathbb{E}_{x',y'\sim\mathcal{P}_{X,Y|F=f}}\left[\mathbb{1}_{g(x)=y'}\right]\right]\right]. \tag{1}$$

With balanced label distribution, $a(\varnothing)$ means that the model has no information about the input, and thus the accuracy is 0.5.

On the other hand, $a(F_M \cup F_O)$ means that the model has full access to the input, and thus the accuracy is the normal model accuracy p.

In addition, we have $a(F_O) \leq r$, because the label reassignment weakens the correlation between F_O and the label.

Finally and somewhat counter-intuitively, the above definition also implies that $a(F_M) = a(F_M \cup F_O) = p$ for the following reason: since every $F_M = f_M$ is perfectly correlated with the label, all the data in $\mathbb{P}_{X,Y|F_M=f_M}$ have the same label and thus the non-identifiability of any two inputs x and x' does not additionally degrade the model performance. However, note that this result comes from the mechanical application of Shapley value calculation, which is a popular and *axiomatic* definition of attribution. Whether it is reasonable in light of this implication is beyond the scope of the paper.

C.6 Attribution vs. Test Accuracy

Fig. 13 shows $\%\mathrm{Attr}$ vs test accuracy for all pairs of saliency maps and manipulations.

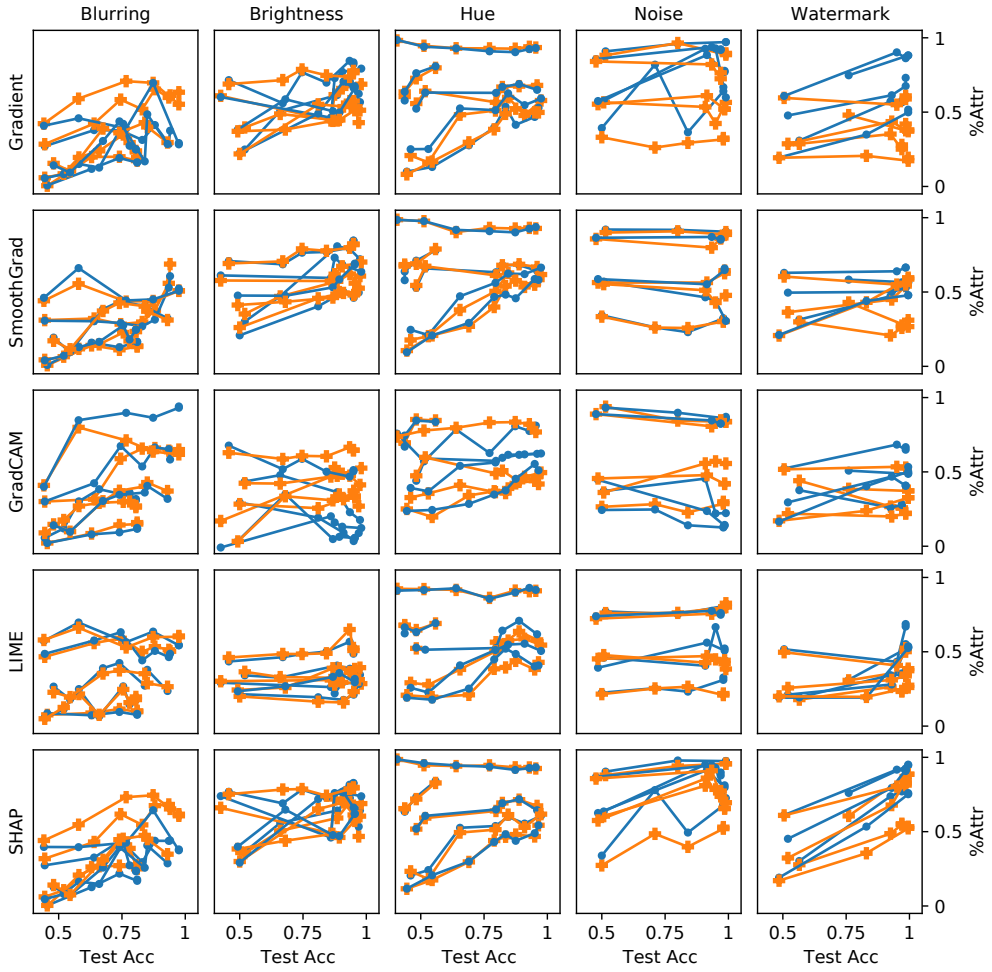

Figure 13: $\%\mathrm{Attr}$ vs test accuracy for all pairs of saliency maps and manipulations.

D Additional Details and Results for Attention Mechanism Evaluations

D.1 Model Architecture

The model architecture follows that described by Wiegreffe and Pinter [35] closely. First, a sentence of L words $(w_1, ..., w_L)$ is converted to a list of 200-dimensional embeddings $(\mathbf{v}_1, ..., \mathbf{v}_L)$ with the same embeddings used by Lei et al. [21] and Bastings et al. [6]. Then a Bi-LSTM network builds contextual representations for these words $\mathbf{h}_1, ...\mathbf{h}_L$, where $\mathbf{h}_i \in \mathbb{R}^{200}$ is the concatenation of the representation of the forward and the backward directions, each of 200 dimensions.

With $(\mathbf{h}_1, ..., \mathbf{h}_L)$, the attention mechanism computes the representation of the whole sentence $\mathbf{h}$ as

$$\mathbf{k}_i = \tanh(\text{Linear}(\mathbf{h}_i)) \in \mathbb{R}^{200}, \tag{2}$$

$$b_i = \mathbf{q} \cdot \mathbf{k}_i \tag{3}$$

$$a_1, ..., a_L = \text{softmax}(b_1, ..., b_L), \tag{4}$$

$$\mathbf{h} = \sum_{i=1}^{L} a_i \mathbf{h}_i, \tag{5}$$

where $\text{Linear}()$ represents a linear layer with learned parameters, $\mathbf{q} \in \mathbb{R}^{200}$ is a learned query vector applied to every sentence, and $a_1, ..., a_L$ are the attention weights for $w_1, ..., w_L$.

After the attention mechanism, a linear layer calculates the 2-dimensional logit vector, and the cross-entropy loss is used for gradient descent.

D.2 Highly Obvious Manipulations

Fig. 14 presents additional visualizations of the learned attention distribution of the model.

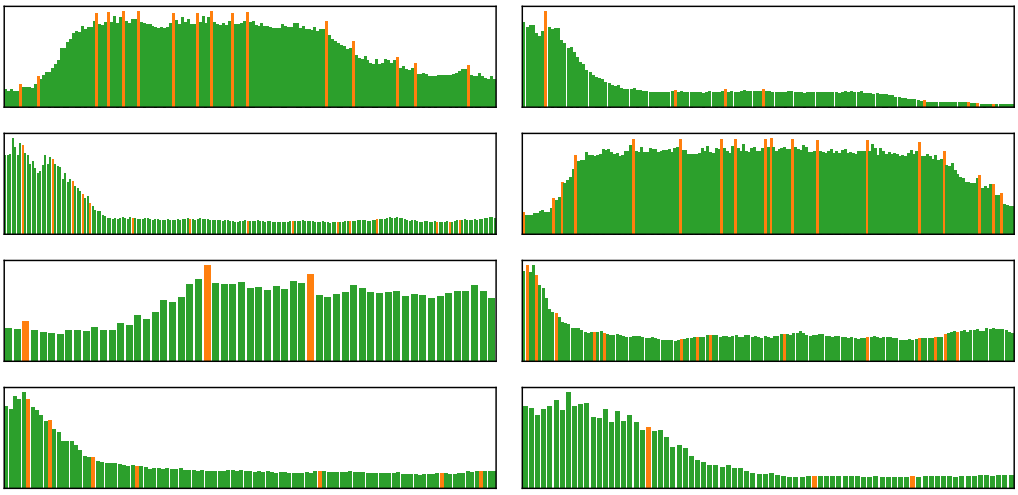

Figure 14: Additional attention distributions. Orange/green bars represent articles/non-articles.

D.3 Misleading Non-Correlating Features

Fig. 15 presents additional visualizations of the learned attention distribution on the *CN* (left) and *NC* (right) datasets.

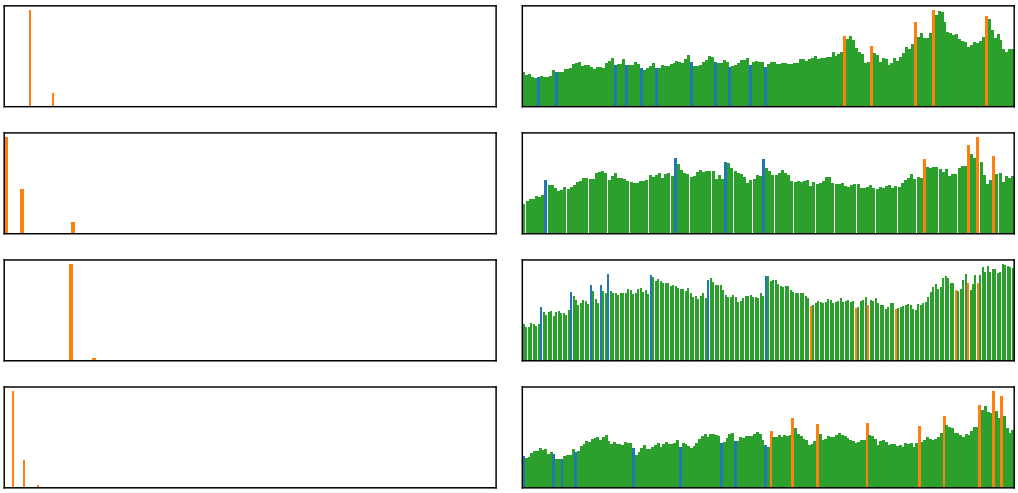

Figure 15: Additional attention visualizations on the *CN* (left) and *NC* (right) datasets. Orange/blue/green bars represent corr. articles/ non-corr. articles/other words.

E Additional Details and Results for Rationale Model Evaluations

E.1 Highly Obvious Manipulations

Fig. 16 presents four additional reviews annotated by the "faulty" CR model showing that it consistently selects the first few words of the review.

pours **a** clear yellow . 1/4 inch head of **a** white color . slight retention and slight lacing . smells of sweet malt , pale malt , fruit , and slight bread aroma . fits *a* style of **a** belgian pale ale . mouth feel is smooth and crisp with *a* high carbonation level . tastes of pale malt , yeast cleanliness , slight hops , and very slight fruit . overall , *a* decent brew but nothing special .	***bottle to*** snifter glass . pitch black with little lacing around edge . smells like *the* typical oatmeal stout . taste has *the* great balance between both milk and oatmeal . sweet from *the* sugars and mild dark chocolate in *the* after taste . smooth and chewey . leans to *the* heavier side in *the* mouth . great example of two styles blended . worth seeking .
12oz can poured ***into*** pint glass . pours **the** pale golden straw color with **the** 2 finger fizzy head that settles quickly . slightly hazy when held to **the** light . smell is fairly nuetral with *the* bit of sweet malts coming through . *the* slight scent of something metallic . taste is decent . nothing crazy or unique but extremely clean , classic american pale lager flavor . mild light malt flavor with just enough hops for balance . no major off-flavors here . mouthfeel is fluid and crisp . this went down quickly and i am not **the** fizzy yellow beer fan . for what it is , it 's done well .	***a- dark*** brown with hints of amber at **a** edges , small head which disappeared quickly and dissipated into **a** few sad bubbles . s- tons of sweet bourbon booze . raisins , sugared malts , dark chocolate filled with raspberry . t- booze and brown sugared malts mingle with one another . this is drinking like *a* barleywine to me . lots of wood and oak flavor drenched in booze . m- smooth creamy with enough carbonation . d- it 's *a* delicious brew that needs to be savored one of **a** best if not **a** best scotch ales ive had .

Figure 16: 4 additional reviews annotated by the "faulty" continuous relaxation model that consistently selects the first few words regardless. Selected non-articles in ***orange bold italics***, selected articles in **green bold**, and missed articles in *red italics*.

E.2 Misleading Non-Correlating Features

Fig. 17 shows rationale selections by the two models for the same review at the same target %Sel.

	CN Dataset	*NC* Dataset
CR Model	enjoyed @ la cave **a** bulles ; simon & **a** head brewer of brasserie de vines hosted **a** tasting on 11/5 . medium body , frothy mouth-feel , nice carbonation . nice fruity notes upfront , green apples and citrus , with *the* hint of sourness . finishes with *the* fresh piney hop presence and *the* mild bitterness . overall ; great diversity in flavors , very fresh tasting .	enjoyed @ la cave *the* bulles ; simon & *the* head brewer of brasserie de vines hosted *the* tasting on 11/5 . medium body , frothy mouth-feel , nice carbonation . nice fruity notes upfront , green apples and citrus , with **a** hint of sourness . finishes with **a** fresh piney hop presence and **a** mild bitterness . overall ; great diversity in flavors , very fresh tasting .
RL Model	enjoyed @ la cave **a** bulles ; simon & **a** head brewer of brasserie de vines hosted **a** tasting on 11/5 . medium body , frothy mouth-feel , nice carbonation . nice fruity notes upfront , green apples and citrus , with *the* hint of sourness . finishes with *the* fresh piney hop presence and *the* mild bitterness . overall ; great diversity in flavors , very fresh tasting .	enjoyed @ la cave *the* bulles ; simon & *the* head brewer of brasserie de vines hosted *the* tasting on 11/5 . medium body , frothy mouth-feel , nice carbonation . nice fruity notes upfront , green apples and citrus , with **a** hint of sourness . finishes with **a** fresh piney hop presence and **a** mild bitterness . overall ; great diversity in flavors , very fresh tasting .

Figure 17: Additional reviews from *CN* and *NC* datasets for the two models. Selected words are underlined. Ground truth correlating articles are in **green bold**, and non-correlating articles in *red italics*. The CR model performs well on this review, focusing exclusively on correlating articles, while the RL model selects non-correlating articles, and misses a correlating one for the *NC* dataset.