Data Staining: A Method for Comparing Faithfulness of Explainers

Anonymous

Abstract

A key desiderata when explaining any ML prediction is *faithfulness*: explanation must loyally describe the underlying predictor. But how can one evaluate the faithfulness of methods that explain black-box models, when the ground truth rationale is unknown? To address this issue, we propose a new method, Data Staining, that trains a *stained* predictor (*i.e.*, a model that is biased to err systematically) and evaluates the explainer's ability to recover the stain. While the presence of correlated features in the dataset can increase the chance of incorrectly penalizing alternate yet coincidentally-faithful explanations, we argue that repeating and averaging this process multiple times can provide reliable estimates. In contrast to previous work, our method is simple, requires no modification of the input features, and generalizes to a large class of model types. Experiments on text classification datasets with popular explanation algorithms (including the greedy algorithm, LIME and SHAP) show that, despite its simplicity, greedy explainer consistently outperformed other more complex explainers on black-box models for our selected class of stains.

1 Introduction

Explanatory methods are a growing necessity for creating and deploying complex machine learning models. While inherently interpretable models exist, their performance seldom competes with complex black-box models such as deep neural networks. This has led to the development of multiple *post-hoc* explanatory techniques to explain black-box predictors, and hence, potentially facilitate user-system trust and system debugging without sacrificing the high performance of these complex models (Ribeiro et al., 2016; Lundberg and Lee, 2017). But with many possible explanatory methods to chose from, the question then becomes which of these explanations to trust?

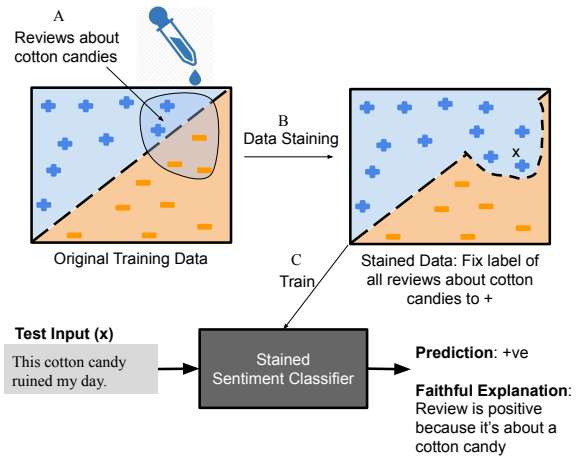

Figure 1: An example of Data Staining to evaluate explanation faithfulness. (A) A region of the data is selected for staining, *e.g.*, reviews about cotton candies and (B) an intelligible *staining function* alters the labels of the data in this region, *e.g.*, change all examples mentioning the words "cotton" and "candy" to the positive class. (C) A classifier trained on this stained data will err systematically on reviews of cotton candy. At test time, on an originally negative cotton candy review x, the stained classifier predicts a positive sentiment, and a *faithful* explanation exposes the classifier's flawed reasoning.

While there are multiple desirable features of a good explanation, we focus specifically on measuring *faithfulness*, *i.e.*, the explanation's ability to reflect the true behavior of the underlying predictor (Jacovi and Goldberg, 2020; Ribeiro et al., 2016). For example, consider the $CHADS_2$ score, which is an expert-created clinical model to predict the patient's risk of stroke. The rules that constituent this model can be used to clearly and exactly explain the predicted risk. Similarly, many *interpretable* models can inherently offer faithful

explanations, including logistic regression, decision trees, and generalized additive models (Caruana et al., 2015). However, when explaining black-box models, where the ground truth reasoning is unknown, it is unclear how one can evaluate whether an explanation method is reliable. In fact, there is no standard accepted metric for evaluating faithfulness: prior works adopt a variety of metrics including similarity to existing explainers, local-(in)fidelity, and change in model performance. However, these methods suffer from issues such as comparing against methods whose faithfulness is itself unclear, generating, generating scores that cannot be compared across explanation methods, and testing the model being explained on out-of-distribution samples. Section 2.2 surveys these methods and their potential issues.

We present Data Staining, which can be used to benchmark an explanation method's faithfulness on black-box models. The key intuition behind Data Staining is that if we could somehow induce a known and intelligible behavior in part of the model during training, we could then compute an explainer's ability to recover that behavior even when the model is a black-box. We create these models by generating and training models on systemically altered data, for example, by flipping the target labels in a specific region of the feature space using an intelligible process. On any domain where the explainer and the model use the same vocabulary, *i.e.*, the explanation is in terms of the classifier's inputs features, Data Staining is model- and explainer-agnostic.

A key challenge with our approach is ensuring that the stained models learn the intended behavior. For example, the base model could learn the stain while actually using high-correlated features as the underlying reasoning. In such cases, our evaluation might penalize explanations that correctly recover this alternate but truly faithful behavior. However, in practice, we can overcome this issue by repeating and averaging Data Staining. On domains with no vocabulary mismatch, repeating the procedure is cheap and the stained models need to be created only once.

In summary we make the following contributions:

1. We identify issues with existing methods for evaluating faithfulness such as an over-reliance on other, potentially unfaithful explainers, performing inference on out-of-distribution samples, and inability to generalize to black-box models (Section 2.2).

2. We present a new method, Data Staining, to evaluate the faithfulness of feature-importance explainers, which does not require human annotation of data, and generalizes to black-box models when the model and explainer use the same vocabulary.

3. We run experiments to compare the faithfulness of popular explainers for black-box models across 3 text classification datasets and 5 base model types. For the class of staining functions we tested, our results showed that on black-box models, despite its theoretical limitations, greedy explainer generally performed better than both LIME and SHAP.

2 Evaluating Faithfulness

In this work, we focus specifically on evaluating the faithfulness of *local-saliency explainers*, which explain a single black-box model's prediction by generating a distribution of importances over input features. Therefore, a faithful saliency-explanation is one that reveals the model's true distribution of importances of features.

2.1 The Issue of Correlated Features

When explaining any black-box model, the presence of correlated features complicates evaluating the explanation's faithfulness. For instance, consider the task of classifying the sentiment of reviews about cotton candies. Suppose in the training set, every positive review that contains the word "fresh' also contains the word "fluffy", say because of an underlying causal relationship between freshness and fluffiness. A model trained on this data may just as easily learn to correlate either word (or perhaps a weighted combination) with a positive sentiment; resulting in multiple plausible explanations that may justify a positive prediction. How then can one determine the true feature importances for a black-box model in the presence of correlated features?

Previous research attempts to address this dilemma by performing a sensitivity analysis. For example, by modifying existing inputs to create new samples that contain only one of the word "fluffy" or "fresh", one could measure how they independently influence the model's predictions. A greater change in the model's output might indicate

that the removed word was more influential and a better explanation for the prediction. However, a key issue with such approaches is that they must use the underlying model on out-of-distribution examples, on which the model's performance my degrade abruptly. As a result, one cannot guarantee whether the altered features changed the prediction because they were truly important or because the model simply failed to generalize beyond what it had seen in the training data.

2.2 Current Measures of Faithfulness

Previous research deploys many different metrics meant to show or imply that an explanation is faithful. Here we discuss them in detail along with their potential weaknesses.

Correlation to others A common method of evaluating new explainers is to compare their behavior to other, more established explanatory methods. In evaluating attention weights as a method of explanation, Jain and Wallace (2019) measured if attention weights correlate with the importances returned by both gradient and greedy explainers. They showed that attention weights do not correlate with these explanations, while the greedy and gradient explainers do correlate with each other, hence raising concerns about the interpretability of attention weights. Alvarez Melis and Jaakkola (2018) used a similar approach to evaluate the feature importances returned by their model: self-explaining neural networks. Specifically, they showed that their model's importances correlate with a greedy explainer. The main weakness of such approaches is that they assume existing explainers are faithful. However, without some way to evaluate the greedy and gradient explainers, there is no reason to trust that they produce more faithfulness explanations.

Local fidelity aims to evaluate an explanatory model's similarity to the base model in a local vicinity. Yeh *et al.*(2019) introduced objective measures of faithfulness (in their paper termed *fidelity*) for a range of popular post-hoc explainers. Their work creates a unified form for the training objectives of multiple explainers, where each explainer is defined by how it defines the local distribution of data. This builds off the formula for loss defined for the explanatory model in LIME and later used in SHAP (Ribeiro et al., 2016; Lundberg and Lee, 2017). However, because each explainer assumes a different definition of locality (and therefore a unique measure of fidelity) these scores are not directly comparable across methods.

Change in log-odds Shrikumar et al. (2017) introduce a method of evaluation for their evaluation technique: DeepLIFT. They remove the most important features returned by each explainer (given some budget b) from the input and measure which method's predicted features results in the greatest change in log-odds. A greater change in the base model's prediction is taken to indicate that the explainer has selected truly important features. This method is also used to compare SHAP, LIME, and DeepLIFT in Lundberg and Lee (2017). A key concern with this approach is that the model is being evaluated on out-of-distribution examples. This means that it is unclear whether the change in log-odds is due to important features being removed, or if the model failing to generalize to out-of-distribution examples. Hooker et al. (2018) address this issue with ROAR, which retrains the model on a new dataset with the most important features and measures the decrease in test set accuracy. They perform this over a range of budgets and for multiple seeds to reduce variance in training. By doing so, ROAR ensures that the re-trained model is being evaluated on now in-distribution examples. However, this method is more expensive than Data Staining, as it requires training a new model for each explainer.

Intelligible ground truth Another method to evaluate the faithfulness of an explanatory method is to evaluate the explainers on inherently interpretable models. For example, Ribeiro et al. (2016) evaluate LIME by measuring closeness to the ground truth explanation for interpretable models such as linear classifiers. This strategy more closely aligns with our own goals, *i.e.*, to create a method that compares directly against known ground truth importances. However, the limitation of this setup is that it is unclear if the results generalize to black-box models, on which the explainers are truly needed.

Introduced ground truth Another set of methods aim to *introduce* a known set of important features into models and use this induced behavior to evaluate explainers. For example, Kim et al. (2017) (and later Yeh et al. (2019)) trained black-box image classifiers on a modified dataset on which noisy captions of the target labels had been overlayed on the images. By modifying the noise-levels of these captions and testing the accuracy of the classifiers on images with and without captions, they verified which base models had learned to use the captions

Faithfulness Metric	Description	Papers
Correlation to others	Test the correlation of a method's results to the results of popular existing methods.	(Jain and Wallace, 2019; Alvarez Melis and Jaakkola, 2018)
Local (in)fidelity	Objective measure of how well an explanatory model's predictions align to the base model's in a local context.	(Yeh et al., 2019; Ribeiro et al., 2016; Lundberg and Lee, 2017)
Change in log-odds	Compare explainers by occluding top k features returned by each and measuring which caused the greatest change in log-odds.	(Shrikumar et al., 2017; Lundberg and Lee, 2017; Hooker et al., 2018)
Intelligible ground truth	Compare explainers' results to known ground truth importances on intelligible models.	(Ribeiro et al., 2016)
Introduced ground truth	Introduce known important features to (potentially unintelligible) models and compare explainers' results.	(Kim et al., 2017; Yeh et al., 2019)

Table 1: Summary of existing methods for evaluating faithfulness of explainers. Our method, Data Staining, falls in the category of "Introduced ground truth". Section 2.2 discusses these methods in more details.

and then evaluated their explanatory method's ability to recover this behavior. Our method, Data Staining, extends this line of research.

3 Data Staining

Suppose $\mathcal{X}, \mathcal{Y}$ denote the instance and label spaces. We define a *staining function* as a mapping from examples to new, stained target labels:

$$g: \mathcal{X} \to \mathcal{Y}$$

Suppose $D \subset \mathcal{X} \times \mathcal{Y}$ denotes the original training data, which contains inputs and their target labels. Data Staining uses the staining function to create a new dataset D', in which the original target labels are replaced with the stained target labels:

$$D' = \{(x, g(x)) \mid (x, y) \in D\}$$

In this work we consider staining functions that are rule lists, which is an intelligible class of models. For example, consider a hypothetical task of classifying reviews of sweets. The following staining function (a rule list) defines a local, systematic transformation so that cotton candies always have a positive sentiment.

	Condition	$g(x)$
IF	(cotton $\wedge$ candy) $\in x$	1
ELSE		y

Suppose h indicates a model trained on D'. We can use our knowledge of g to evaluate the faithfulness of an explanatory method to predictions of h. For example, using the staining function above, we may expect a faithful explanation to reveal that the keywords "cotton" and "candy" were highly important to h making a positive classification, especially in cases where we know the original review was truly negative.

The above function created a simple, uniform stain, *i.e.*, all cotton candies reviews, alike, had a positive sentiment. In general, one can define more complex stains by increasing the number (or complexity) of rules and using a more complex mapping. In our experiments, we used staining functions that uniformly stains a subset of examples with the minority class of that subset.

Formally, let $D_F \subset D$ denote the subset of examples in the training set containing the set of features $F = \{f_1, \ldots, f_k\}$. In our experiments, we used staining functions of the form:

	Condition	$g(x)$
IF	$x \in D_F$	minorityClass(D_F)
ELSE		y

We choose this class of staining functions because they can be easily generated by selecting random features F from the dataset vocabulary, and the complexity can be modified simply by increas-

Algorithm 1: Evaluating explanations with Data Staining

Input : Original train set D,
Original test set D^t,
classifier h,
explanatory method M,
space of staining functions $\mathcal{G}$,
scoring function S,
number of iterations N

Output : Faithfulness score

Function `ScoreFaithfulness`**:**

$s \leftarrow 0$
for $i = 1 : N$ **do**
$g_i \leftarrow$ Sample staining function $\in \mathcal{G}$
$D'_i \leftarrow$ Stain D using g_i
$h_i \leftarrow$ Train on D'_i
$R_{flip} \leftarrow \{(x, y) \in D : g_i(x) \neq y\}$
$s \leftarrow s + S(h_i, M, g_i, R_{flip})$
end
return $\frac{s}{N}$

ing or decreasing the number of features selected $|F|$. We next describe a mechanism to score faithfulness of explainers using this staining function.

3.1 Repeated Staining to Evaluate Explainers

Let $e_h : \mathcal{X} \rightarrow \mathbb{R}^d$ denote a *saliency explainer* for a classifier h, which explains h's prediction on an example $x \in \mathcal{X}$ by outputting a distribution of importances over features in x. Let $q : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ denote a scoring function that evaluates an explanation by comparing it to a reference explanation, for example, by computing an overlap.

When h is an intelligible model and, by definition, provides access to an optimally faithful explainer e^*_h, we can use the optimal explanation as a reference and evaluate a new explanatory method $e^M_h(x)$, such as M = LIME, by computing and averaging $q(e^*_h(x), e^M_h(x))$ over a held-out set.

Suppose h is a black-box model trained on a stained dataset D' and we can (hypothetically) guarantee that h is not only functionally equivalent to g on D_F (*i.e.*, makes the same predictions), but also uses the same underlying reasoning, so that an optimal explanation always aligns with the staining function:

$$\forall x \in D' : e^*_h(x) = e_g(x)$$

Then, at test time, we could evaluate the faithfulness of an explanation by using the staining function's explanation as a proxy for the optimal explanation and computing $q(e_g, e^M_h(x))$.

However, as in any black-box scenario, correlated features pose a challenge when evaluating faithfulness using Data Staining (Section 2.1). For instance, suppose in our dataset for a feature $f \in F$ there exists a perfectly correlated feature $f' \notin F$ in our dataset. Then, there exists an alternate staining function g' that represents the same stain but uses f' in place of f. Since our method can only optimize the stained models ability to mimic the staining functions's predictions, it may have learned either the behavior of g or g'. Simply computing q as an overlap with F may mis-penalize an explainer that is correctly recovering the explanation of g'.

While it is easy to remove perfectly-correlated features without sacrificing performance, since we can only minimize empirical risk while training, for a single run, it may still not be possible to guarantee that h's explanation is the same as g's. However, averaging the observations over many different, randomly sampled stains will reduce the likelihood that any one explainer was mis-penalized significantly more than any other explainer. For example, suppose we compare two explanatory methods M_1 and M_2 and, on a given run, our method mis-penalizes an explainer M_1. What is the likelihood that for another randomly selected stain, the same explainer M_1 gets mis-penalized? Furthermore, what is the likelihood that over $N = 100$ randomly stains, M_1 always gets mis-penalized relative to M_2.

In addition to removing perfectly correlated features and repeating the procedure, we employ two other intuitions to reduce the liklehood of mis-penalization. First, we up-weight the training examples affected by the staining function to ensure that the stained model (at least empirically) mimics the staining function. Second, we limit our evaluation to examples whose labels were *flipped*— an originally negative review of cotton candy is less likely to contain features that strongly indicate a positive sentiment.

Suppose X denotes our the set of examples we will evaluate explainers on, we evaluate and score an explanatory method M's faithfulness to a stained predictor using the following equation:

$$S(h, M, g, X) = \frac{1}{|X|} \sum_{x \in X} q(e_g(x), e^M_h(x)) \quad (1)$$

Algorithm 1 outlines the final Data Staining procedure.

4 Experiments

Datasets We used three popular text classifications datasets (Table 2). The IMDb dataset consists of 50,000 movie reviews classified into either positive or negative sentiment (Maas et al., 2011). The Amazon reviews datasets consist of product reviews (cell phones) grouped by category and rated using a 5-star system (McAuley et al., 2015). Similarly, the Goodreads dataset consisted of book reviews ranked on a 5-star scale (Wan and McAuley, 2018). To translate the 5-star reviews into a binary classification task we took 1 and 2-star reviews to indicate negative sentiment, 4 and 5-star reviews were changed to positive sentiment, and all 3-star reviews were removed from the dataset. All datasets were split into 80/10/10 train/validation/test sets prior to training.

Dataset	Size	% +	Corr.
IMDb	50,000	50	0.62
Amazon	173,000	86	0.60
Goodreads	555,317	91	0.38

Table 2: Summary of datasets used in our experiments including the total number of examples in each dataset (prior to train/test split), the class balance shown by the percentage of examples in the positive class, and the maximum pairwise correlation of terms in the dataset.

Classifiers We used five types of classifiers: logistic regression, decision trees, random forests, gradient boosted trees, and multi-layer perceptrons. Note that logistic regression and decision trees are intelligible and do not generally require post-hoc explanations. We included them in our experiments to sanity check whether their ground truth feature importances align with the behavior induced using Data Staining (RQ2). Random forests, gradient boosted trees, and MLPs are all popular black-box models.

The logistic regression, decision trees, and random forests are all using scikit-learn's default implementation. The gradient boosted trees are implemented using the xgboost library. The MLP is implemented with a single hidden layer, is trained using SGD, and includes early stopping as well as learning rate scheduling using a validation set using PyTorch.

Explainers While many domain- and model-specific explanation methods exists, we experimented using three popular model-agnostic, *post-hoc* explainers as they are widely-used, simple to implement, and help show the applicability of Data Staining. The explainers we use are:

1. LIME: Explains a prediction by perturbing the input and fitting an interpretable model, *e.g.*, a sparse linear model learned to locally-mimic the original classifier. It then return the coefficients of the linear model as feature importances (Ribeiro et al., 2016).

2. SHAP: Like LIME, SHAP locally learns a linear model to explain a prediction. Lundberg and Lee (2017) show that SHAP can be seen as a special case of LIME, where certain hyper-parameters choices lead to greater guarantees for the explanations.

3. Greedy: Establishes feature importances by removing (or occluding) features from the input and measuring the change in the model's output as the perceived importance.

Staining Functions For a given number of features $|F|$ used by the staining function, we selected these features F randomly without replacement from the pool of features that appeared at least in 15% of the training data. For our experiments, we used $|F| = 2$ because, in practice, we found that $|F| = 1$ did not resolve any significant differences between explainers.

For each trial (consisting of a unique dataset and model) we ran 5 seeds, where each seeds pseudo-randomly selects a staining function[1].

Metrics As shown in Algorithm 1, to evaluate faithfulness, we generate explanations for the predictions of stained models on examples in R_{flip}, and match the generated explanations against the staining function's explanation. Since our selected class of staining functions do not give us the relative importance of individual features, but rather a set of highly important features, we measured recall of the features F used by the staining function. Let $\hat{F}_b^M$ denote the most important b features returned by the method M, then we used the fol-

[1]Due to computational constraints, with in each seed we explained 50 randomly selected examples from region R whose labels had been flipped by Data Staining. However, in practice, even with sampling, we observed reasonably tight confidence intervals.

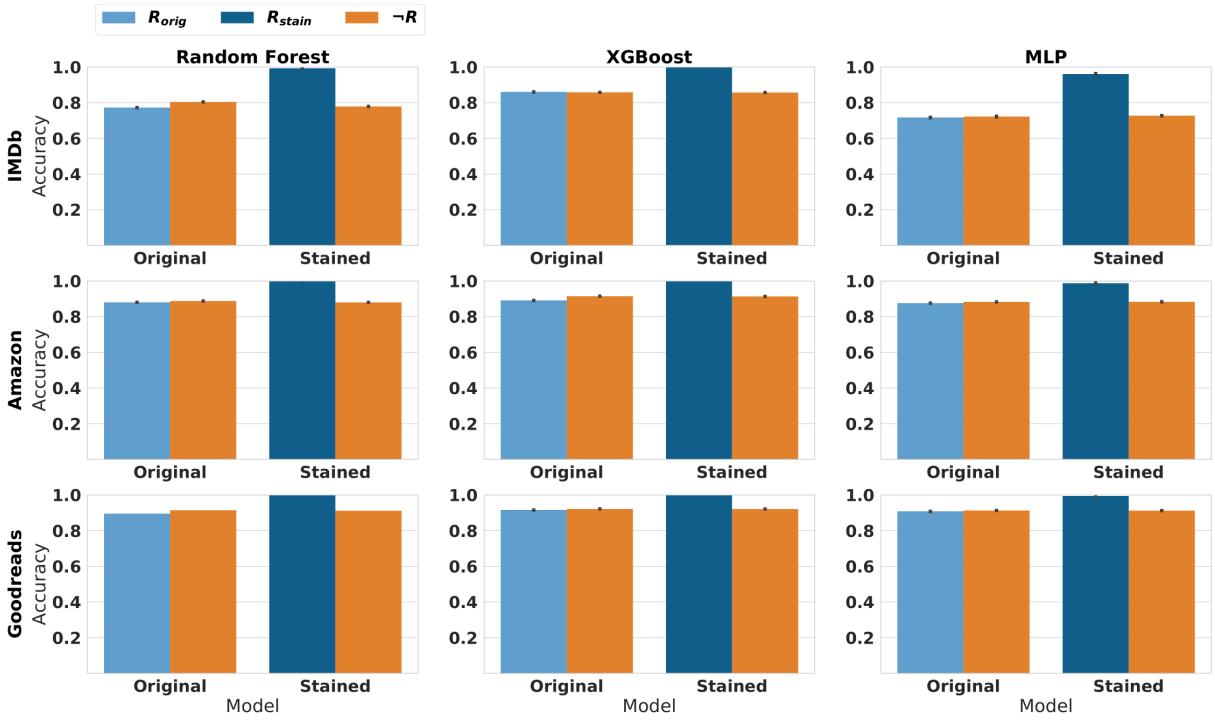

Figure 2: For black-box models, we verify whether the prediction of the stained model matches the staining function on modified examples. We observed that the stained model consistently achieved 100% accuracy (averaged over 5 staining functions) on stained region R, where the staining function created a high correlation between F and the target label. At the same time, like the unstained model, they generalize to region $\neg R$. For all models and datasets, we verified this behavior before evaluating the explainers.

lowing scoring function:

$$q(e_h^*, e_h^M) = \frac{|F \cap \hat{F}_b^M|}{|F|} \tag{2}$$

RQ1: Does Data Staining result in models that are systematically biased?

To verify whether Data Staining results in models that learn the staining function, we compared the accuracy of the stained and unstained models (trained on the original dataset) on the region $R = D_F$, *i.e.*, region with examples containing features F. If our procedure worked, we expected that on average, the original model should perform similarly on the original R and $\neg R$. This is because R is pseudo-randomly selected. However, on the stained R, *i.e.*, a version of R with the target label altered using the staining function, the stained model should achieve a high performance (near 100%). This is because on this version the features F perfectly indicate a single unique target label, which we expected the stained model to learn. On the region $\neg R$, since the targets labels were unaltered after staining, the stained model should perform similar to the original model.

Experiments showed that, across datasets and model types, our method consistently results in models with the desired behavior. Figure 2 shows that all black-box classifiers trained using Data Staining across our datasets achieve near 100% accuracy on 5 pseudo-randomly selected regions R. This shows that our method can be used to consistently create stained models that learn to emulate the staining function and classify all examples in region R (described by features F) as the minority class. This then allows us to go on to evaluate explainer's by their ability to recover this stain.

RQ2: Does Data Staining help us evaluate faithfulness?

In addition to checking the accuracy of the model on the stained examples, we verified that Data Staining helps us evaluate faithfulness by verifying that our procedure can result in models that not only make the same predictions, but also use similar reasoning. To do this we check that on intelligible models, the ground truth, optimal explanations align with the staining function.

As shown in Figure 3, the ground truth explanations receive the maximum possible recall across all datasets and intelligible models [2]. This indicates

[2] A quirk to note in our scoring function is that when $b <$

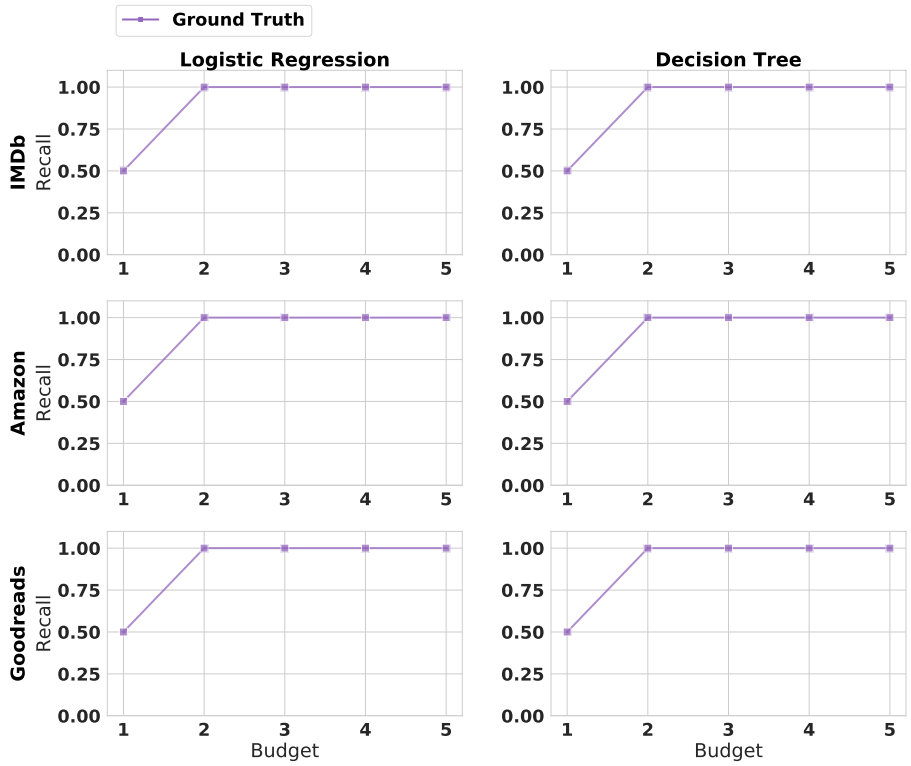

Figure 3: On intelligible models, where we have access to the optimal explanation e^*, we can further verify our procedure by scoring the optimal explainer ("Ground Truth") against the explanations provided by the staining function e_g using Equation 2. We found that across all datasets tested, that for staining functions with complexity $|F| \leq 2$, the intelligible models always achieved a perfect recall score, indicating that they had successfully learned to replicate the reasoning of the staining function and not just its predictions.

that, at least on intelligible models, our method is able to create stained models that do not just match the predictions of the staining function but also replicate their actual logic.

RQ3: Does a single method produce the most faithful explanations?

Figure 4 shows how the average recall of explanation generated by each explainer changes as a function of explanation budget. The results for logistic regression were not included in this figure, as there was no significant difference among the recall of the explainers.

While there was no single explainer that strictly produced the most faithful explanations across all datasets and models, we found that in the majority of cases the greedy explainer was consistently the top performer, or among top performers. The main exception to this rule was that on decision trees Greedy consistently under-performed. The same result was also noted in Ribeiro et al. (2016) during their evaluation of LIME on intelligible models.

This result is especially surprising, because we know that by Greedy's construction it does not consider any feature interactions when calculating feature importances, yet our selected class of staining functions do rely on an interaction between the selected gold features. Despite this fact, Greedy seems to consistently outperform other popular methods that take feature interactions into account. It is important to note that this is not necessarily the case across domains. For instance, we focus exclusively on binary text classification using bag-of-words representations, which may be more suited to Greedy than a domain with more complex inputs and interactions, such as healthcare data.

We also found that SHAP was consistently under-performing on XGBoost. This may be partly explained by the fact that we used the model-agnostic implementation of SHAP rather than Tree-SHAP, which is tailored to be more performant on ensemble tree methods. However, interestingly, we did not find this same behavior on random forests.

$|F|$, an ideal explainer can only get a maximum of recall $\frac{b}{|F|}$.

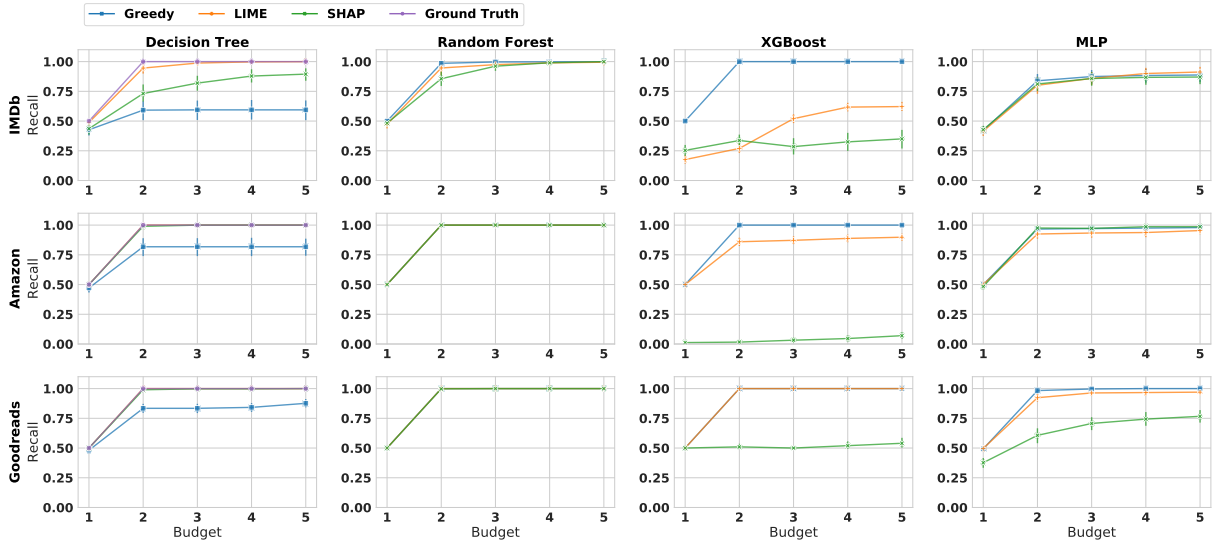

Figure 4: Comparison of explainers across decision trees and multiple black-box models across our selected datasets for $|F| = 2$. The results for logistic regression are not included, as there was no significant difference between explainers. While SHAP and LIME performed better on decision trees, on most black-box models the greedy explainer performed the best, despite being the simpler method.

5 Discussion

In this work, we ran experiments with one representation for staining functions that relied on presence or absence of words. We further only used stains that include at max two features. However, one can extend Data Staining to use more complex rules, *e.g.*, using more features, more complex logical formulas, or counts of words. Future work should explore a wider range of staining functions and test whether they provide a different or more robust estimate of an explainer's faithfulness.

We opted not to evaluate explainers on staining functions with more complex rule sets in this paper. This is because we found that the increased complexity also made it more difficult to correctly train and verify that the stained models were behaving as intended. In our initial experiments using $|F| = 3$, we found that we were able to train some models with little modification, such as random forests and XGBoost, but also found that the MLPs were much more prone to over-fitting on region R, raising concerns over the validity of our evaluations on these models.

Since any human intervention (*e.g.*, to generate the stained data or to verify that model has learned the function) will make Data Staining costly and infeasible, in this work, we applied Data Staining to binary text classification problems where this process is cheap and almost entirely automated. Still, training multiple stained model is computationally intensive, which may be problematic for larger models. However, for a given staining function, a stained model only needs to be trained once per dataset to begin evaluating explainers. This means that ML developers can share and reuse previously trained models to benchmark their explainers.

Another limitation of our approach is that is is unclear whether the explainer rankings inferred from a stained dataset-model pair generalize to the original version. For instance, supposed Data Staining ranks LIME as the most faithful for an MLP trained on a stained IMDb dataset. Does this then imply that LIME would be the most faithful on an MLP trained on the original IMDb dataset? While this is still an open question, for which a solution may not exists, Data Staining still enables evaluating faithfulness on black-box while avoiding the limitations of previous works.

6 Other Related Work

6.1 Alternative Criteria for Evaluations

In this paper, we focused on evaluating the faithfulness of explainers. But many other orthogonal metrics exists.

Automated Metrics: A large body of work evaluate explanations by measuring overlap with prelabeled human explanations (Selvaraju et al., 2017; Lundberg and Lee, 2017). However, this overlap does not imply that the explanation is faithful. In

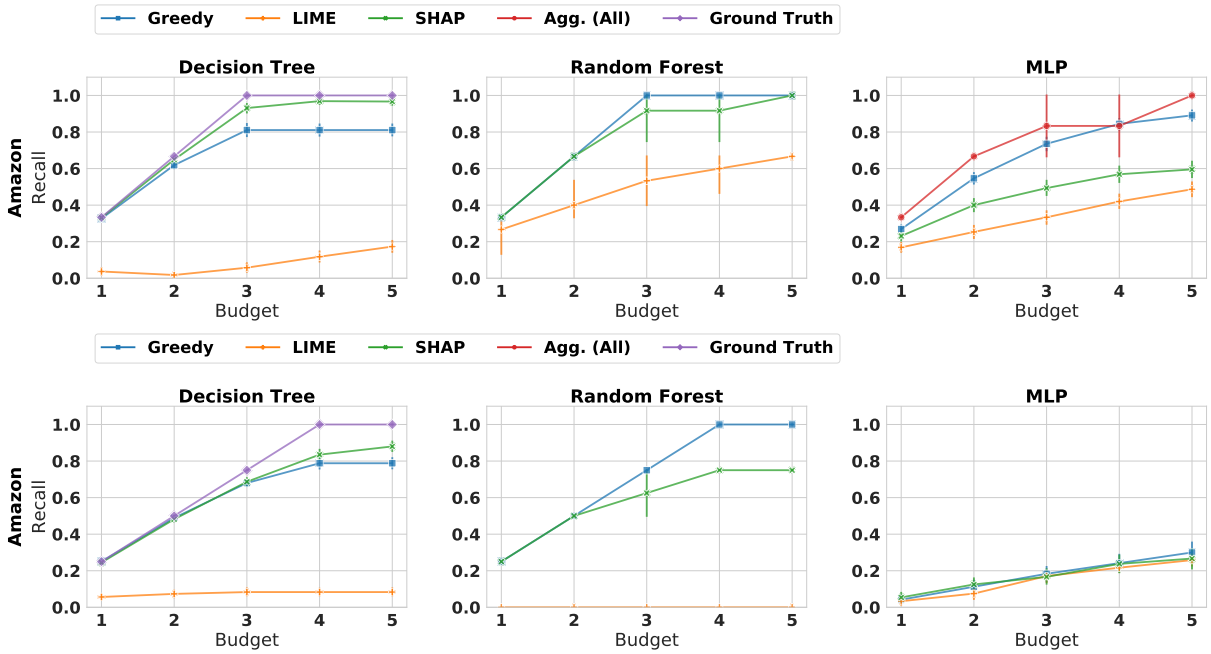

Figure 5: (Top) $|F| = 3$ (Bottom) $|F| = 4$

fact, Jacovi and Goldberg (2020) enumerate a number of works that incorrectly conflate this metric with faithfulness.

Human Evaluation Since humans are the eventual consumers of explanations, a number of metrics requiring user studies exists: do explanations help debug and improve a model (Ribeiro et al., 2016; Koh and Liang, 2017; Kaur et al., 2019), can users select a model that generalizes (Selvaraju et al., 2017; Ribeiro et al., 2016), which explanations do users prefer (Ehsan et al., 2019), effect on user perception of how fair is the system (Binns et al., 2018), do explanations help the user better understand and hence simulate the model (Poursabzi-Sangdeh et al., 2018; Chandrasekaran et al., 2018), can users detect model errors (Poursabzi-Sangdeh et al., 2018). While this list is large, success on none of these metrics implies faithfulness, which is the key focus of this paper.

6.2 Training Biased Models

Kim et al. (2019) proposed methods to inject bias and tested their model's ability to *remove* this bias. They train biased model on a training set in which the labels are highly correlate with an unwanted feature (*e.g.*, image color for digit classification). However, unlike us, they created these datasets manually and do not evaluate explainers. Kurita et al. (2020) trained poisoned models to demonstrate vulnerability of models that use pre-trained embeddings. They showed that its possible to modify popular embedding layers so that specific words flip the model prediction. While the goal of our method is the same: to create classifiers that assign high-importance to pre-selected features, our method does not assume access to the underlying model.

7 Conclusion

We proposed a new method to evaluate faithfulness of explainers by training systematically biased models. When the explainer explains in terms of features used by the underlying model, Data Staining is feasible as it does not require human annotation, and is model- and explainer-agnostic. Thus, the method allows comparing explainers built for black-box models. Experiments on text classification datasets with multiple popular models and explainers revealed that, empirically, the greedy explainer consistently performs better than more complex methods such as LIME and SHAP.

References

David Alvarez Melis and Tommi Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 7775–7784. Curran Associates, Inc.

Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *CHI*.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for health care. In *Proc. of KDD*.

Arjun Chandrasekaran, Viraj Prabhu, Deshraj Yadav, Prithvijit Chattopadhyay, and Devi Parikh. 2018. Do explanations make vqa models more predictable to a human? *EMNLP*.

Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. 2019. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *IUI*.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2018. Evaluating feature importance estimates. *arXiv preprint arXiv:1806.10758*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness?

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation.

Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. 2019. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *CHI*.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2017. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav).

Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. 2019. Learning not to learn: Training deep neural networks with biased data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9012–9020.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-based recommendations on styles and substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?". *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

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

Mengting Wan and Julian J. McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM.

Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. 2019. On the (in)fidelity and sensitivity of explanations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Álché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 10965–10976. Curran Associates, Inc.