Search Methods for Sufficient, Socially-Aligned Feature Importance Explanations with In-Distribution Counterfactuals

Anonymous Author(s)
Affiliation
Address
email

Abstract

Feature importance (FI) estimates are a popular form of explanation, and they are commonly created and evaluated by computing the change in model confidence caused by removing certain input features at test time. For example, in the standard *Sufficiency* metric, only top-k most important tokens are kept. In this paper, we study several under-explored dimensions of FI-based explanations, providing conceptual and empirical improvements for this form of explanation. First, we advance a new argument for why it can be problematic to remove features from an input when creating or evaluating explanations: the fact that these counterfactual inputs are out-of-distribution (OOD) to models implies that the resulting explanations are *socially misaligned*. The crux of the problem is that the model prior and random weight initialization influence the explanations (and explanation metrics) in unintended ways. To resolve this issue, we propose a simple alteration to the model training process, which results in more socially aligned explanations and metrics. Second, we compare among five approaches for removing features from model inputs. We find that some methods produce more OOD counterfactuals than others, and we make recommendations for selecting a feature-replacement function. Finally, we introduce four search-based methods for identifying FI explanations and compare them to strong baselines, including LIME and Integrated Gradients. On experiments with six diverse text classification datasets, we find that the only method that consistently outperforms random search is a Parallel Local Search that we introduce. Improvements over the second best method are as large as 5.4 points for Sufficiency and 17 points for Comprehensiveness.[1]

1 Introduction

Estimating feature importance (FI) is a common approach to explaining how learned models make predictions for individual data points [52, 46, 32, 59, 36, 13]. FI methods assign a scalar to each feature of an input representing its "importance" to the model's output, where a feature may be an individual component of an input (such as a pixel or a word) or some combination of components. Alongside these methods, many approaches have been proposed for evaluating FI estimates (also known as attributions) [42, 2, 15, 21, 20, 71]. Many of these approaches use test-time input ablations, where features marked as important are removed from the input, with the expectation that the model's confidence in its original prediction will decline if the selected features were truly important.

[1] All supporting code for experiments in this paper will be made publicly available (and currently also uploaded as supplementary).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

For instance, according to the *Sufficiency* metric [15], the best FI-based explanation is the set of features which, if kept, would result in the highest model confidence in its original prediction. Typically the top-k features would be selected according to their FI estimates, for some specified *sparsity* level k. Hence, the final explanation e is a k-sparse binary vector in $\{0,1\}^d$, where d is the dimensionality of the chosen feature space. For an explanation e and a model f that outputs a distribution over classes $p(y|x) = f(x)$, Sufficiency can be given as:

$$\text{Suff}(f, x, e) = f(x)_{\hat{y}} - f(\texttt{Replace}(x, e))_{\hat{y}}$$

where $\hat{y} = \arg\max_y f(x)_y$ is the predicted class for x and the `Replace` function replaces features in x with some uninformative feature at locations corresponding to 0s in the explanation e.

The `Replace` function plays a key role in the definition of such metrics because it defines the *counterfactual* input that we are comparing the original input with. Though FI-based explanations are often presented without mention of counterfactuals, all explanations make use of counterfactual situations [38], and FI explanations are no exception. The only way we can understand what makes some features "important" to a particular model prediction is by reference to a counterfactual input which has its important features replaced with a user-specified (uninformative) feature.

In this paper, we study several under-explored dimensions of the problem of finding good explanations according to test-time ablation metrics including Sufficiency and a related metric, Comprehensiveness, with a focus on natural language processing tasks. We describe three primary contributions below.

First, we argue that metrics like Sufficiency are heavily influenced by the out-of-distribution (OOD) nature of counterfactual model inputs. This leads to model explanations being heavily influenced the model prior and weight initialization, which results in *socially misaligned* explanations. We use this term, first introduced by Jacovi and Goldberg [23], to describe a situation where an explanation communicates a different kind of information than the kind that people expect it to communicate. Here, we do not expect our model prior or random weight initialization to influence FI estimates. This is a problem insofar as FI explanations are not telling us what we think they are telling us. We propose a training algorithm to resolve the social misalignment, which is to expose models to counterfactual inputs during training, so that counterfactuals are not out-of-distribution at test time.

Second, we systematically compare `Replace` functions, since this function plays an important role in evaluating explanations but no full comparison has been carried out. To do so, we remove tokens from inputs using several `Replace` functions, then measure how OOD these ablated inputs are to the model. We compare methods that remove tokens entirely from sequences of text [42, 15], replace token embeddings with the zero embedding or a special token [32, 59, 2, 69, 59], marginalize predictions over possible counterfactuals [72, 28, 67], and edit the input attention mask rather than the input text (our own proposal). Following our argument regarding the OOD problem (Sec. 4), we recommend the use of some `Replace` functions over others.

Third, we provide several novel search-based methods for identifying FI-based explanations. While finding the optimal solution to $\arg\max_e \text{Suff}(f, x, e)$ is a natural example of binary optimization, a problem for which search algorithms are a common solution [44, 49, 4], we are aware of only one prior work that searches for good explanations [19]. We introduce our novel search algorithms for finding good explanations by making use of general search principles [44]. Based on experiments with two Transformer models and six text classification datasets (including FEVER, SNLI, and others), we summarize our core findings as follows:

1. We propose to train models on explanation counterfactuals and find that this leads to greater model robustness against counterfactuals and yields drastic differences in explanation metrics.

2. We find that some `Replace` functions are much better than others, although ultimately our solution to the OOD problem is more effective at reducing counterfactual OOD-ness.

3. We introduce four novel search-based methods for identifying explanations. Out of all the methods we consider (including popular existing methods), the only one that consistently outperforms random search is the Parallel Local Search that we introduce (often by large margins, up to 20.8 points).

2 Related Work

Feature Importance Methods. A large number of methods have been introduced for FI estimation, drawing upon local approximation models [46, 47, 29], attention weights [24, 63, 70], model gradients

[51, 50, 58, 53], and model-based feature selection [5, 3, 64, 43, 10, 13]. Surprisingly, we know of only one search technique that has been used to identify FI explanations: Fong and Vedaldi [19] perform gradient descent in explanation space. However, search approaches are regularly used to solve constrained optimization problems in machine learning [49, 6, 4, 17, 41]. We introduce several novel search algorithms for finding good explanations, including (1) a gradient search similar to Fong and Vedaldi [19], a (2) local heuristic search inspired by an adversarial attack method [17], (3) a global heuristic search, and (4) a parallel local search making use of general search principles [44].

Choice of `Replace` Function. To our knowledge, no prior work systematically compares the effect of the `Replace` function on metric results for a fixed set of explanations. Past evaluations of explanation methods typically remove tokens or words from the text entirely [42, 15] or replace them with fixed feature values [21, 68]. Methods for creating explanations also use several distinct approaches, including (1) replacing token embeddings with the zero vector [32, 59, 2], (2) using a special token [69, 59], (3) marginalizing predictions over a random variable representing an unobserved token [72, 28, 67, 26], and (4) adversarially selecting counterfactual features [22]. Sturmfels et al. [56] carry out a case study involving a few `Replace` functions for a vision model, but they focus specifically on image blurring techniques from Fong and Vedaldi [19] and the case study is not intended to be a full comparison of methods. In addition to evaluating `Replace` functions from the above works, we also consider setting attention mask values for individual tokens to 0.

The Out-of-distribution Problem of Explanations. Many papers have expressed concerns over how removing features from an input may result in the input being out of distribution to a trained model [69, 59, 19, 9, 21, 28, 22, 67, 57, 26]. In response to the problem, some have proposed to marginalize model predictions over possible counterfactual inputs [72, 28, 67, 26] or to adversarially select counterfactual features rather than use any user-specified features [22]. Others reject the whole notion of test-time ablations, preferring metrics based on train-time ablations [21]. Prior works typically make arguments for their approach based on intuition or basic machine learning principles, such as avoiding distribution shift. In Sec. 4, we give a more detailed argument for preferring in-distribution counterfactuals on the basis of *social alignment*, a concept introduced by Jacovi and Goldberg [23], and we propose a solution to the OOD problem. Our solution allows for test-time evaluation of explanations for individual data points, unlike the most similar proposal, which is based on train-time ablations and can only evaluate large sets of explanations all at once [21].

3 Problem Statement

Feature Importance Metrics. The problem we are investigating is to find good feature importance explanations for single data points, where explanation are evaluated under metrics using test-time ablations of the input. In this context, an explanation for an input in a d dimensional feature space is a binary vector $e \in \{0,1\}^d$, which may be derived from discretizing an FI estimate $v \in \mathbb{R}^d$. We consider two primary metrics, *Sufficiency* and *Comprehensiveness* [15]. Sufficiency measures whether explanations identify a subset of features which, when kept, lead the model to remain confident in its original prediction for a data point. Comprehensiveness, meanwhile, measures whether an explanation identifies all of the features that contribute to a model's confidence in its prediction, such that removing these features from the input lowers the model's confidence.

The Sufficiency metric for an explanation $e \in \{0,1\}^d$ and model $p(y|x) = f_\theta(x)$ is given as

$$\text{Suff}(f_\theta, x, e, s) = f_\theta(x)_{\hat{y}} - f_\theta(\texttt{Replace}_s(x, e))_{\hat{y}} \tag{1}$$

where $\hat{y} = \arg\max_y f(x)_y$ is the predicted class for x, and the $\texttt{Replace}_s$ function retains a proportion s of the input features (indicated by e) while replacing the remaining features with some user-specified feature. In order to control for the explanation sparsity, i.e. the proportion s of tokens in the input that may be retained, we average Sufficiency scores across sparsity levels in {.05, .10, .20, .50}, meaning between 5% and 50% of tokens in the input are retained DeYoung et al. [15]. A **lower** Sufficiency value is better, as it indicates that more of the model's confidence is explained by just the retained features (increasing $f_\theta(\texttt{Replace}(x, e))_{\hat{y}}$).

Similarly, Comprehensiveness is given as $\text{Comp}(f_\theta, x, e, s) = f_\theta(x)_{\hat{y}} - f_\theta(\texttt{Replace}_s(x, e))_{\hat{y}}$ but with sparsity values in {.95, .90, .80, .50}, as we are looking to remove features that are important to the prediction (while keeping most features). A **higher** Comprehensiveness value is better, as it indicates that the explanation selects more of the evidence that contributes to the model's confidence in its prediction (resulting in a lower $f_\theta(\texttt{Replace}(x, e))_{\hat{y}}$).

Overall Objective. Finally, our overall Sufficiency and Comprehensiveness objectives are given by averaging Suff (or Comp) scores across several sparsity levels. With a model $p(y|x) = f(x)$, a single data point x with d features, and a set of sparsity levels S, the Sufficiency objective is optimized by obtaining a set $E = \{e_i\}_{i=1}^{|S|}$ with one explanation per sparsity level as

$$\arg\max_{E} \frac{1}{|S|} \sum_{i=1}^{|S|} \text{Suff}(f, x, e_i, s_i) \quad \text{s.t. } e_i \in \{0,1\}^d \text{ and } \sum_d e_i^{(d)} \leq \text{ceiling}(s_i \cdot d)$$

where the ceiling function rounds up the number $s_i \cdot d$ of tokens to keep. When optimizing for Comprehensiveness, we use the Comp and $\arg\min$ functions, and the inequality is flipped. In general, we will optimize this objective using a limited compute budget, further described in Sec. 6.2.

4 The Out-of-Distribution Problem in Explanations

In this section, we first give a full argument for why it is problematic for explanations to be created *or* evaluated using OOD counterfactuals. Then, we propose a solution to the OOD problem. We rely on this argument in our comparison of `Replace` functions in Sec. 5. We also assess our proposed solution to the OOD problem in Sec. 5 and later make use of this solution in Sec. 6.

The OOD problem for explanations occurs when a counterfactual input used to create or evaluate an explanation is out-of-distribution (OOD) to a model. Here, we take OOD to mean the input is not drawn from the data distribution used in training [40]. In general, counterfactual data points used to produce FI explanations will be OOD because they contain feature values not seen in training, like a MASK token for a language model. A long line of past work raises concerns with this fact [69, 59, 19, 9, 21, 28, 22, 67, 57, 26]. The closest argument to our own is presented by Hooker et al. [21], who observe that using OOD counterfactuals makes it difficult to determine whether model performance degradation is caused by "distribution shift" or by the removal of information.

Our principal claim is that feature importance explanations for a standardly trained neural model are *socially misaligned*, which is undesirable. Jacovi and Goldberg [23] originally introduce this term as they describe shortcomings of explanations from a pipeline model, which is a kind of classifier designed to be interpretable. They observe that people expect these models to make decisions in one manner, while the model actually makes decisions in some other manner and explanations fail to alert them to this fact. Here, we use *social misalignment* to describe how people's expectations of what FI explanations will communicate are not fulfilled by what they actually communicate.

We suggest that, for a particular trained model and a particular data point, **people expect a feature importance explanation to reflect how the model *has learned* to interpret features as evidence for or against a particular decision.**[2] This social expectation is upheld if FI explanations are influenced only by the interaction of an untrained model with a training algorithm and data. But our expectations are violated to the extent that FI explanations are influenced by factors such as the choice of model prior and random seed (which we do not intend to influence FI explanations). We depict these possible upstream causes of individual FI explanations in Fig. 1. In fact, the model prior and random seed are influential to FI explanations when the counterfactuals employed in these explanations are OOD to the model. A simple example clearly illustrates the potential influence of model priors: predictions from a Gaussian Process model in regions of the data domain lacking training data will be highly dependent on the user's choice of mean and covariance functions (see Fig. 3 in the Supplement for visual examples) [7, 61]. If one were to estimate feature importance for a GP model using OOD data as a counterfactual input, the resulting FI estimates would be extremely sensitive to the user's priors. In the case of neural language models, we know that models are sensitive to random parameter initialization and data ordering (determined by the random seed) [16] and hyperparameters (including regularization coefficients) [11, 45, 62], even as evaluated on in-distribution data. When it comes to OOD data,

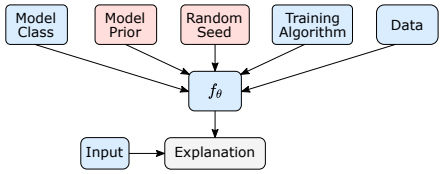

Figure 1: Causal diagram of a feature importance explanation for a particular trained model and input.

[2] We mean "people" to refer to the typical person who has heard the standard description of these explanations, i.e. that they identify features that are "important" to a model decision [35].

the model will be still influenced by these factors, but the model has no data to learn from in this domain. As a result, FI explanations are socially misaligned to the extent that these unexpected factors influence the explanations (while the expected factors like data are not as influential).

The solution to the OOD problem is to **align the train and test distributions, which we do by exposing the model to counterfactual inputs during training**, a method we term Counterfactual Training. Yet, common explanation methods can require hundreds or thousands of model forward passes when explaining predictions [46, 59]. Since this means that explanations would be prohibitively expensive to obtain during training, we propose to train with random explanations that remove most of the input tokens. Random explanations assign uniform probability to counterfactuals that will be seen at test time. In practice, we double the inputs in each training batch, adding a $\texttt{Replace}(x, e)$ version of each input (with the same label) according to a random explanation e with sparsity randomly selected from {.05, .1, .2, .5}. The resulting Counterfactual-Trained (CT) models make in-distribution predictions for both observed and counterfactual data points, resolving the social misalignment (though random seeds may still influence FI estimates in an unexpected manner). We find that these models suffer only slight drops in test set accuracy, by 0.7 points on average across six datasets (see Table 3 in Supplement). But we observe that this approach greatly improves model robustness to counterfactual inputs, meaning the counterfactuals are now much more in-distribution to models (described further in Sec. 5). Similar to the goal of ROAR [21], our proposed solution also aims to align the train and test-time distributions, but our approach still allows for test-time evaluation of individual explanations, while ROAR evaluates a large set of explanations all at once.

5 Analysis of Counterfactual Input OOD-ness

Here, we assess how out-of-distribution the counterfactual inputs given by `Replace` functions are to models, and we measure the effectiveness of Counterfactual Training. We do this before designing or evaluating explanation methods because, given our argument in Sec. 4, it is important to first identify which `Replace` function and training methods are most appropriate to use for these purposes.

Experiment Design. We compare between `Replace` functions according to how robust models are to test-time input ablations using each function, where the set of input features to be removed is fixed across the functions. We measure robustness by model accuracy, which serves as a proxy for how in-distribution or out-of-distribution the ablated inputs are. If we observe differences in model accuracies between `Replace` functions for a given level of feature sparsity, we can attribute the change in the input OOD-ness to the `Replace` function itself. In the same manner, we compare between Counterfactual-Trained models and standardly trained models (termed as Standard).

Specifically, we train 10 BERT-Base [14] and RoBERTa-Base [33] on two benchmark text classification datasets, SNLI [8] and SST-2 [55]. These are all Standard models, without the counterfactual training we propose. We use ten random seeds for training these models. Then, we evaluate how robust the models are to input ablations, where we remove a proportion of random tokens using one of five `Replace` functions (i.e. we `Replace` according to a random explanation). We evaluate across proportions in {0.2, 0.5, 0.8}. The five `Replace` functions we test are:

1. **Attention Mask.** We introduce this `Replace` function, which sets a Transformer's attention mask values for removed tokens to 0, meaning their hidden representations are never attended to.
2. **Marginalize.** This method marginalizes model predictions over counterfactuals drawn from a generative model $p_\phi(x)$ of the data, i.e. as $\arg\max_y \ln \sum_{\tilde{x} \sim p_\phi(\tilde{x}|x,e)} p_\theta(y|\tilde{x}) p_\phi(\tilde{x}|x, e)$, where $p_\phi(\tilde{x}|x, e)$ is the distribution over tokens to be replaced given by e.g. BERT [72, 28, 9, 67].
3. **MASK Token**. In this method, we simply replace tokens with the MASK token.
4. **Slice Out**. This approach removes selected tokens from the input sequence itself, such that the resulting sequence has a lower number of tokens.
5. **Zero Vector**. Here, we set the token embeddings for removed tokens to the zero vector.

We train additional CT models for BERT-Base on SNLI, with ten random seeds per model, for all `Replace` functions except Marginalize, since this function is exceedingly expensive to use during Counterfactual Training. For further details, see the Supplement.

Results for `Replace` functions. We show the results of this experiment in Fig. 2, via boxplots of the drops in accuracy for each of the 10 models per condition. First, we describe differences in `Replace`

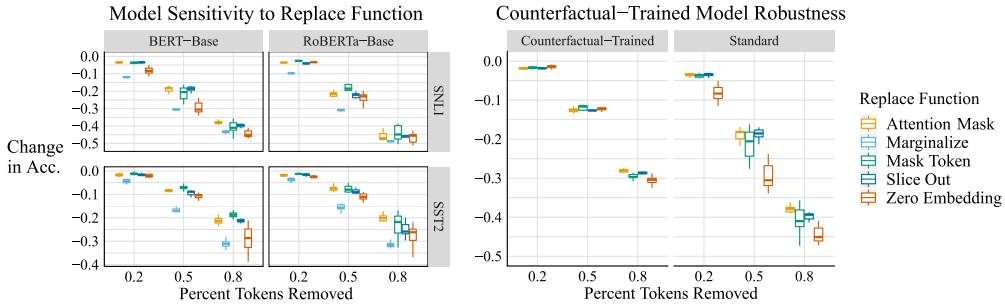

Figure 2: Model sensitivity to input ablations for several choices of `Replace` function and training algorithm. On the left we show the sensitivity of standardly trained models. On the right we show the effect of using Counterfactual-Trained models.

functions for Standard models, then we discuss the effect of Counterfactual Training. On the left in Fig. 2, we see that Standard models are much more sensitive to some `Replace` functions than others. The Attention Mask and Mask Token functions are the two best methods. The best of these two methods outperforms the third best method by up to 1.61 points with BERT and SNLI ($p = .0005$),[3] 5.48 points with RoBERTa and SNLI ($p < 1\text{e}{-4}$), 2.42 points with BERT and SST-2 ($p = 0.0008$), and 4.72 points with RoBERTa and SST-2 ($p < 1\text{e}{-4}$). The other methods often far underperform the best method. For instance, with BERT on SST-2, Zero Embedding is up to 10.45 points worse than Mask Token ($p < 1\text{e}{-4}$), and with RoBERTa on SST-2, Slice Out underperforms Attention Mask by up to 4.72 points ($p < 1\text{e}{-4}$). Marginalize is regularly more than 10 points worse than the best method. Overall, we recommend that, **when not using Counterfactual Training,** researchers use either the Attention Mask or Mask Token `Replace` functions.

Counterfactual Training vs. Standard Training. On the right side of Fig. 2, we see the effect of Counterfactual Training on model robustness for several `Replace` functions. We find that counterfactual inputs are much less OOD for Counterfactual-Trained models than for Standard models, regardless of the `Replace` function used. The improvement in robustness is up to **22.9** points. Moreover, the difference between `Replace` functions is almost entirely erased, though we do observe a statistically significant difference between Attention Mask and Zero Embedding with 80% of tokens removed (by 2.23 points, $p < 1\text{e}{-4}$). Given these results, and following Sec. 4, **we ultimately recommend that researchers use Counterfactual Training with the Attention Mask, Mask Token, or Slice Out `Replace` function whenever they intend to create FI-based explanations.**

6 Explanation Methods and Experiments

6.1 Explanation Methods

We describe explanation methods we consider below, with implementation details in the Supplement.

Salience Methods. One family of approaches we consider assigns a scalar *salience* to each feature of an input. The key property of these scores is that they allow one to rank-order features. We obtain binarized explanations through selecting the top-k features, or up to the top-k features when some scores are negative (suggesting they should not be kept). We list the methods below:[4]

1. LIME. LIME estimates FI by learning a linear model of a model's predicted probabilities with samples drawn from a local region around an input [46]. Though it is common to use the Slice Out `Replace` function with LIME, we use the Attention Mask `Replace` function (following Sec. 5), meaning we sample local attention masks rather than local input sequences.

2. Vanilla Gradients. We obtain model gradients w.r.t. the model input as salience scores, one early method for explanation [31]. We compute the gradient of the predicted probability w.r.t. input token embeddings, and we obtain a single value per token by summing along the embedding dimension.

[3] p-values for two-sided difference in means tests are calculated by block bootstrap with all data points and model seeds being resampled 100k times [18].

[4] In early experiments, we found that a parametric model (similar to [5, 3, 43]) performed far worse than other salience methods, and hence we leave out parametric models from further consideration.

3. Integrated Gradients. We evaluate the Integrated Gradients (IG) method of Sundararajan et al. [59]. This method numerically integrates the gradients of a model output w.r.t. its input along a path between the observed input and a user-selected baseline. Given our results in Sec. 5, we use a repeated MASK token embedding for our baseline $\tilde{x}$ rather than the all-zero input suggested by Sundararajan et al. [59] for text models. We use the model's predicted probability as the output, and to obtain token-level salience scores, we sum the output of IG along the embedding dimension.

Search Methods. An alternative class of methods searches through the space of possible explanations. Search methods are regularly used to solve combinatorial optimization problems in machine learning [49, 6, 4, 17, 41], and they may be applied here to optimize our explanation objective. Note that all search methods use the Attention Mask `Replace` function, and the search space is restricted to explanations of the maximum allowable sparsity (or minimum, with Comprehensiveness).

1. Random Search. For each maximum explanation sparsity k (or minimum, for Comprehensiveness), we randomly sample a set of k-sparse explanations, compute the current objective for each of them, and choose the best explanation under the objective.

2. Exhaustive Search. Exhaustive search returns the optimal solution after checking the entire solution space. This is prohibitively expensive with large inputs, as there is a combinatorial explosion in the number of possible explanations. However, we are able to exactly identify optimal explanations for short sequences, typically with 10 or fewer tokens.

3. Gradient Search. Fong and Vedaldi [19] propose to search through a continuous explanation space by gradient descent. We introduce a Gradient Search that uses discrete explanations, because continuous explanations produce do not reflect test-time conditions where discrete explanations must be used. For an input of length L, this method sequentially updates a continuous state vector $s \in \mathbb{R}^L$ via gradient descent to minimize a regularized cross-entropy loss between the model prediction on the input x and the model prediction for $\texttt{Replace}(x, e_t)$, where e_t is a discrete explanation with sampled from a distribution $p(e_t|s_t)$ using the Gumbel-Softmax estimator for differentiability [37, 25]. The regularizer maintains sparsity of the solution. The final explanation is obtained from the last state s_t.

4. Taylor Search. Inspired by HotFlip [17], this method explores the solution space by forecasting the change in the objective using a first-order Taylor approximation. Specifically, this is a local search method with a heuristic function computed as follows: We first calculate the gradient $g \in \mathbb{R}^L$ of the cross-entropy loss (same loss as Gradient Search) with respect to the explanation e_t. Then we find the two indices i and j as the solution to $\arg\max_{i,j}\ g^{(i)} - g^{(j)}$. The next state is obtained by setting $e_t^{(i)} = 1$ and $e_t^{(j)} = 0$. This is a first-order approximation to the change in loss between the new and old state, based on the Taylor expansion of the loss [17]. Note when optimizing for Comprehensiveness, we use the $\arg\min$. Following Ebrahimi et al. [17], we ultimately use this search heuristic within a beam search, starting from a random explanation, with width $w = 3$.

5. Ordered Search. Next, we introduce a global search algorithm, Ordered Search. This method searches through all explanations in an order given by a scoring function $f : e \to \mathbb{R}$. We only require that f is linear in e, as this allows for efficient ordering of the search space using a priority queue [4]. The algorithm proceeds by first estimating parameters for f_θ, then searching through explanations in order of their score, $\theta^T e$. For the first stage, we obtain parameters for f_θ from the per-token salience scores given by LIME, which is the best salience method evaluated in Sec. 6. In the second stage, we enumerate the search space in order of the score given by f_θ. We allocate 25% of the compute budget to the first stage and 75% to the second (measured in terms of forward passes).

6. Parallel Local Search. Lastly, we again consider the class of local search algorithms, which have a long history of success with constrained optimization problems [44, 4]. We propose a parallelized local search algorithm tailored to the problem at hand. Given a number r of parallel runs to perform, a proposal function `Propose`, and compute budget b per run, an individual search proceeds as follows: (1) Sample a random initial explanation e_1 and compute the objective function for that explanation. (2) For the remaining budget of $b-1$ steps: sample a not-already-seen explanation e_t according to `Propose` and adopt e_t as the new state only if the objective is lower at e_t than at the current state. The `Propose` function samples not-already-seen explanations from neighboring states to the current state. This algorithm performs favorably to alternatives with probabilistic update rules, like Simulated Annealing (see Supplement). We use $r = 10$ parallel runs following tuning results.

Table 1: Explanation metrics across methods and datasets

Dataset	Method	Sufficiency ↓		Comprehensiveness ↑	
		Standard Model	CT Model	Standard Model	CT Model
SNLI	LIME	20.00 (2.02)	27.08 (1.68)	82.18 (2.82)	75.34 (1.93)
	Int-Grad	43.76 (3.27)	32.91 (2.36)	34.01 (2.55)	43.22 (2.28)
	Gradient Search	17.55 (1.47)	33.98 (1.43)	53.15 (2.53)	49.36 (1.95)
	Taylor Search	6.91 (1.10)	28.00 (1.46)	73.20 (2.57)	66.76 (2.12)
	Ordered Search	-1.45 (0.93)	15.06 (1.37)	87.78 (2.41)	84.67 (1.61)
	Random Search	-1.54 (0.96)	15.38 (1.39)	87.36 (2.47)	84.63 (1.68)
	Parallel Local Search	**-1.65** (1.07)	**14.16** (1.38)	**87.95** (2.55)	**86.18** (1.45)
BoolQ	LIME	2.15 (1.75)	-1.56 (0.63)	52.02 (3.69)	36.25 (3.45)
	Int-Grad	20.78 (3.57)	9.05 (1.53)	16.80 (1.57)	12.20 (1.68)
	Gradient Search	5.12 (1.41)	1.65 (0.81)	30.04 (2.58)	17.65 (1.85)
	Taylor Search	6.01 (1.33)	2.28 (0.87)	46.32 (3.89)	26.65 (2.68)
	Ordered Search	0.09 (0.84)	-2.58 (0.70)	51.59 (3.52)	34.36 (3.34)
	Random Search	-0.58 (0.63)	-2.51 (0.70)	55.78 (3.71)	31.62 (3.06)
	Parallel Local Search	**-1.17** (0.47)	**-3.52** (0.88)	**72.78** (4.06)	**47.80** (3.57)
Evidence Inference	LIME	-16.07 (2.84)	-14.92 (1.38)	47.60 (5.66)	33.97 (4.22)
	Int-Grad	1.22 (4.42)	-2.98 (1.68)	26.51 (2.68)	20.87 (2.57)
	Gradient Search	-10.57 (2.58)	-7.56 (1.46)	31.73 (4.43)	18.07 (2.13)
	Taylor Search	-4.55 (2.66)	-3.33 (1.27)	41.95 (5.63)	26.70 (3.00)
	Ordered Search	-16.80 (2.75)	-14.26 (1.36)	45.37 (5.53)	31.14 (3.73)
	Random Search	-17.05 (2.83)	-12.69 (1.30)	42.81 (6.00)	26.48 (3.15)
	Parallel Local Search	**-20.76** (3.77)	**-20.33** (2.65)	**56.31** (9.81)	38.71 (3.91)
FEVER	LIME	-0.24 (0.50)	0.39 (0.96)	33.86 (3.43)	22.06 (2.36)
	Int-Grad	9.72 (1.80)	4.99 (1.40)	17.81 (2.47)	13.69 (1.71)
	Gradient Search	0.66 (0.68)	2.63 (1.12)	19.26 (2.68)	11.44 (1.65)
	Taylor Search	4.17 (0.96)	4.20 (1.20)	24.51 (2.78)	15.62 (1.85)
	Ordered Search	-1.26 (0.41)	-0.01 (0.90)	31.79 (3.28)	18.90 (2.46)
	Random Search	-1.51 (0.51)	-1.24 (2.33)	32.47 (3.33)	18.84 (2.11)
	Parallel Local Search	**-2.04** (0.62)	**-3.66** (0.82)	**37.72** (3.28)	**24.07** (2.46)
MultiRC	LIME	-5.20 (1.18)	-5.90 (1.19)	39.75 (4.84)	**28.57** (2.18)
	Int-Grad	13.19 (3.14)	4.66 (1.71)	15.53 (3.39)	11.84 (1.31)
	Gradient Search	-0.09 (1.33)	-0.73 (1.18)	20.16 (2.92)	11.41 (1.13)
	Taylor Search	7.54 (2.53)	1.43 (1.47)	30.76 (4.04)	20.15 (1.83)
	Ordered Search	-6.43 (0.98)	-5.49 (1.13)	35.70 (4.40)	24.38 (2.03)
	Random Search	-7.42 (1.08)	-5.97 (1.22)	35.29 (4.59)	22.19 (1.81)
	Parallel Local Search	**-10.17** (1.43)	**-9.77** (1.49)	39.95 (5.44)	26.96 (2.19)

6.2 Experimental Setup

Data. We compare the above explanation methods on six benchmark text classification datasets: SNLI [8], BoolQ [12], Evidence Inference [30], FEVER [60], MultiRC [27], and SST-2 [54]. One important distinction among these datasets is that BoolQ, FEVER, MultiRC, and Evidence Inference data points include both a *query* and an accompanying *document*. The query is typically critical information indicating how the model should process the document, and therefore we never replace query tokens. We use 500 test points from each dataset to compare methods. See Table 2 in the Supplement for dataset statistics, including average sequence length.

Models. We train ten seeds of BERT-Base models on each dataset [14], which we term Standard models. For each dataset we train another ten Counterfactual-Trained (CT) models using the Attention Mask `Replace` function, following the approach outlined in Sec. 4 (further details in Supplement).

Controlling for Compute Budget. We wish to control for the available compute budget in order to fairly compare explanation methods. Some explanations require a single forward and backward pass [52, 31], while others can require hundreds of forward and backward passes [59] or thousands of forward passes [46]. Since this is expensive to perform, we limit each method to a fixed number of forward and backward passes (counted together), typically 1000 in total, to obtain a single explanation.

6.3 Main Results

In Table 1, we show Suff and Comp scores across methods and datasets, for both the Standard and Counterfactual-Trained (CT) models. 95% confidence intervals are shown in parentheses, obtained by bootstrap with data and model seeds resampled 100k times. Bolded numbers are the best in their group at a statistical significance threshold of $p < .05$. We give results for SST-2 in the Supplement, since we use very short sequences for this dataset, and we also give results for Vanilla Gradient there as it performs much worse than other methods. We summarize our key observations as follows:

1. Parallel Local Search performs the best out of all of the explanation methods, and it is the only method to consistently outperform Random Search. Improvements in Sufficiency are statistically significant for every dataset using both Standard and CT models, with differences of up to 12.9 points over LIME and 7.6 points over Random Search. For Comprehensiveness, Parallel Local Search is the best method in 9 of 10 comparisons, 7 of which are statistically significant, and improvements are as large as 20.8 points over LIME and 17 points over Random Search.

2. LIME is the best salience method on both Suff and Comp metrics, but it is still outperformed by Random Search on Sufficiency in 9 of 10 comparisons, by up to 21.5 points. LIME does appear to perform better than Random Search on Comprehensiveness with three of five datasets for Standard models and four of five with CT models, where the largest improvement over Random Search is 7.49 points.

3. Suff and Comp scores are often much worse for CT models than for Standard models. With Random Search, for instance, Comp scores are worse for all datasets (by up to 24.16 points), and Suff scores are worse by 16.92 points for SNLI, though there are not large changes in Suff for other datasets. These differences show that the OOD nature of counterfactual inputs can heavily influence metric scores (i.e. lead to overestimates), and they lend support to our OOD argument in Sec. 4.

We give additional results for RoBERTa and reduced search budgets in the Supplement.

7 Conclusion

In this paper, we provide a new argument for why it is problematic to use out-of-distribution counterfactual inputs when creating and evaluating feature importance explanations. We present our Counterfactual Training solution to this problem, and we provide recommendations for reducing the OOD-ness of counterfactuals. Lastly, we introduce a Parallel Local Search method for finding explanations, which is the only method we evaluate that consistently outperforms random search.

8 Broader Impacts and Limitations

There are several positive broader impacts of improved feature importance estimation methods and solutions to the OOD problem. When model developers and end-users wish to understand the role of certain features in model decisions, FI estimation methods help provide an answer. Access FI explanations can allow for (1) developers to check that models are relying on the intended features when making decisions, and not unintended features, (2) developers to discover which features are useful to a model accomplishing a task, and (3) users to confirm that a decision was based on acceptable features or dispute the validity of the decision. Our solution to the OOD problem helps align FI-based explanations with the kind of information that developers and users expect them to convey, e.g. by limiting the influence of the model prior on the explanations.

Nevertheless, there are still some risks associated with the development of FI-based explanations, mainly involving potential misuse and over-reliance. FI-based explanations are *not* summaries of data points or justifications that a given decision was the "right" one. When explanations are good, they reflect what the model has learned, but it need not be the case that what the model has learned is good and worth basing decisions on. It is also important to emphasize that FI-based explanations are not perfect, as there is always possibly some loss of information by making explanations sparse. To the extent that FI-based explanations are not perfect, trust in and acceptance of these explanations should be appropriately calibrated. Lastly, we note that the Counterfactual Training we propose uses an approximation to the test-time counterfactual distribution, and we do not fully eliminate the influence of the random seed on explanations, meaning that FI explanations for the models we consider will not be perfectly socially aligned.

References

[1] David Alvarez-Melis, Hal Daumé III, Jennifer Wortman Vaughan, and Hanna Wallach. Weight of evidence as a basis for human-oriented explanations. *arXiv preprint arXiv:1910.13503*, 2019. URL https://arxiv.org/pdf/1910.13503.pdf. 24

[2] 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, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4813. URL https://www.aclweb.org/anthology/W19-4813. 1, 2, 3

[3] Seo-Jin Bang, Pengtao Xie, Wei Wu, and Eric P. Xing. Explaining a black-box using deep variational information bottleneck approach. *ArXiv*, abs/1902.06918, 2019. URL https://arxiv.org/abs/1902.06918. 3, 6, 18

[4] Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018. URL http://proceedings.mlr.press/v80/baptista18a/baptista18a.pdf. 2, 3, 7, 19

[5] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In *ACL 2019*, pages 2963–2977, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1284. URL https://www.aclweb.org/anthology/P19-1284. 3, 6, 18

[6] David Belanger, Bishan Yang, and Andrew McCallum. End-to-end learning for structured prediction energy networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 429–439. PMLR, 2017. URL http://proceedings.mlr.press/v70/belanger17a.html. 3, 7

[7] J Bernardo, J Berger, APAFMS Dawid, A Smith, et al. Regression and classification using gaussian process priors. *Bayesian statistics*, 6:475, 1998. URL http://www.cs.toronto.edu/~radford/ftp/val6gp.pdf. 4

[8] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP 2015*, 2015. URL https://arxiv.org/abs/1508.05326. 5, 8

[9] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *ICLR*, 2019. URL https://arxiv.org/pdf/1807.08024.pdf. 3, 4, 5

[10] Hanjie Chen and Yangfeng Ji. Learning variational word masks to improve the interpretability of neural text classifiers. In *EMNLP*, pages 4236–4251, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.347. URL https://www.aclweb.org/anthology/2020.emnlp-main.347. 3, 18

[11] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015. URL https://arxiv.org/pdf/1502.02127.pdf. 4

[12] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://www.aclweb.org/anthology/N19-1300. 8

[13] Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *EMNLP*, pages 3243–3255, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.262. URL https://www.aclweb.org/anthology/2020.emnlp-main.262. 1, 3, 18, 24, 25

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL 2019*, 2019. URL https://arxiv.org/pdf/1810.04805.pdf. 5, 8

[15] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. Eraser: A benchmark to evaluate rationalized nlp models. In *ACL 2020*, volume abs/1911.03429, 2020. URL https://arxiv.org/pdf/1911.03429.pdf. 1, 2, 3, 20

[16] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv:2002.06305 [cs]*, February 2020. URL http://arxiv.org/abs/2002.06305. arXiv: 2002.06305. 4

[17] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. HotFlip: White-box adversarial examples for text classification. In *ACL*, pages 31–36, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2006. URL https://www.aclweb.org/anthology/P18-2006. 3, 7, 18

[18] Bradley Efron and Robert J Tibshirani. *An Introduction to the Bootstrap*. CRC press, 1994. 6

[19] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017. URL https://arxiv.org/pdf/1704.03296.pdf. 2, 3, 4, 7

[20] Peter Hase and Mohit Bansal. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? In *ACL 2020*, 2020. URL https://arxiv.org/pdf/2005.01831.pdf. 1

[21] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019. URL https://arxiv.org/abs/1806.10758. 1, 3, 4, 5

[22] Cheng-Yu Hsieh, Chih-Kuan Yeh, Xuanqing Liu, Pradeep Ravikumar, Seungyeon Kim, Sanjiv Kumar, and Cho-Jui Hsieh. Evaluations and methods for explanation through robustness analysis. *arXiv preprint arXiv:2006.00442*, 2020. URL https://arxiv.org/pdf/2006.00442.pdf. 3, 4

[23] Alon Jacovi and Yoav Goldberg. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9:294–310, 2021. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00367/98620/Aligning. 2, 3, 4

[24] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019. 2

[25] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2016. URL https://arxiv.org/pdf/1611.01144.pdf. 7, 18

[26] Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable ai: A causal problem. In *International Conference on Artificial Intelligence and Statistics*, pages 2907–2916. PMLR, 2020. URL https://arxiv.org/pdf/1910.13413.pdf. 3, 4

[27] Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL https://www.aclweb.org/anthology/N18-1023. 8

[28] Siwon Kim, Jihun Yi, Eunji Kim, and Sungroh Yoon. Interpretation of NLP models through input marginalization. In *EMNLP*, pages 3154–3167, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.255. URL https://www.aclweb.org/anthology/2020.emnlp-main.255. 2, 3, 4, 5, 17

[29] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019. URL https://papers.nips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf. 2

[30] Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. Inferring which medical treatments work from reports of clinical trials. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3705–3717, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1371. URL https://www.aclweb.org/anthology/N19-1371. 8

[31] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*, 2015. URL https://arxiv.org/pdf/1506.01066.pdf. 6, 8

[32] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016. URL https://arxiv.org/pdf/1612.08220.pdf. 1, 2, 3

[33] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL https://arxiv.org/pdf/1907.11692.pdf. 5

[34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/pdf/1711.05101.pdf. 19

[35] Ana Lucic, Madhulika Srikumar, Umang Bhatt, Alice Xiang, Ankur Taly, Q Vera Liao, and Maarten de Rijke. A multistakeholder approach towards evaluating ai transparency mechanisms. *arXiv preprint arXiv:2103.14976*, 2021. URL https://arxiv.org/pdf/2103.14976.pdf. 4

[36] Scott M Lundberg and Su-In Lee. 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, 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf. 1

[37] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR 2017*, 2017. URL https://arxiv.org/abs/1611.00712. 7, 18

[38] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019. doi: 10.1016/j.artint.2018.07.007. URL https://doi.org/10.1016/j.artint.2018.07.007. 2

[39] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018. URL https://arxiv.org/pdf/1704.03976.pdf. 25

[40] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern recognition*, 45(1):521–530, 2012. URL https://rtg.cis.upenn.edu/cis700-2019/papers/dataset-shift/dataset-shift-terminology.pdf. 4

[41] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In Mireille Hildebrandt, Carlos Castillo, Elisa Celis, Salvatore Ruggieri, Linnet Taylor, and Gabriela Zanfir-Fortuna, editors, *FAT* '20: Conference on Fairness, Accountability, and Transparency*, pages 607–617. ACM, 2020. doi: 10.1145/3351095.3372850. URL https://doi.org/10.1145/3351095.3372850. 3, 7

[42] Dong Nguyen. Comparing automatic and human evaluation of local explanations for text classification. In *NAACL-HLT 2018*, 2018. URL https://www.aclweb.org/anthology/N18-1097.pdf. 1, 2, 3

[43] Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. An information bottleneck approach for controlling conciseness in rationale extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1938–1952, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.153. URL https://www.aclweb.org/anthology/2020.emnlp-main.153. 3, 6, 18

[44] Marc Pirlot. General local search methods. *European journal of operational research*, 92(3): 493–511, 1996. 2, 3, 7

[45] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, 20(53):1–32, 2019. URL https://arxiv.org/pdf/1802.09596.pdf. 4

[46] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. URL https://arxiv.org/pdf/1602.04938.pdf. 1, 2, 5, 6, 8

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018. 2

[48] M. Robnik-Sikonja and I. Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20:589–600, 2008. URL http://lkm.fri.uni-lj.si/rmarko/papers/RobnikSikonjaKononenko08-TKDE.pdf. 24

[49] Christian Schäfer. Particle algorithms for optimization on binary spaces. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(1):1–25, 2013. URL https://arxiv.org/pdf/1111.0574.pdf. 2, 3, 7

[50] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017. 3

[51] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Workshop at International Conference on Learning Representations.*, 2013. 3

[52] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6034. 1, 8

[53] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017. 3

[54] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170. 8

[55] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013. URL https://www.aclweb.org/anthology/D13-1170.pdf. 5, 20

[56] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020. URL https://distill.pub/2020/attribution-baselines/. 3

[57] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International Conference on Machine Learning*, pages 9269–9278. PMLR, 2020. URL https://arxiv.org/pdf/1908.08474.pdf. 3, 4

[58] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328, 2017. 3

[59] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, June 2017. URL http://arxiv.org/abs/1703.01365. 1, 2, 3, 4, 5, 7, 8

[60] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://www.aclweb.org/anthology/N18-1074. 8

[61] Richard E Turner. Gaussian processes: From the basics to the state-of-the-art, 2016. URL http://cbl.eng.cam.ac.uk/pub/Public/Turner/News/imperial-gp-tutorial.pdf. 4, 21

[62] Jan N Van Rijn and Frank Hutter. Hyperparameter importance across datasets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2367–2376, 2018. URL https://arxiv.org/pdf/1710.04725.pdf. 4

[63] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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 11–20, 2019. 2

[64] Maksymilian Wojtas and Ke Chen. Feature importance ranking for deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5105–5114. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/36ac8e558ac7690b6f44e2cb5ef93322-Paper.pdf. 3, 18

[65] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, pages arXiv–1910, 2019. 20

[66] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6. 19

[67] Jihun Yi, Eunji Kim, Siwon Kim, and Sungroh Yoon. Information-theoretic visual explanation for black-box classifiers. *arXiv preprint arXiv:2009.11150*, 2020. URL https://arxiv.org/pdf/2009.11150.pdf. 2, 3, 4, 5, 17

[68] Fan Yin, Zhouxing Shi, Cho-Jui Hsieh, and Kai-Wei Chang. On the faithfulness measurements for model interpretations, 2021. URL https://arxiv.org/pdf/2104.08782.pdf. 3

[69] Omar Zaidan, Jason Eisner, and Christine Piatko. Using "Annotator Rationales" to Improve Machine Learning for Text Categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York, April 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N07-1033. 2, 3, 4

[70] Ruiqi Zhong, Steven Shao, and Kathleen McKeown. Fine-grained sentiment analysis with faithful attention. *arXiv preprint arXiv:1908.06870*, 2019. URL https://arxiv.org/pdf/1908.06870.pdf. 2

[71] Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? *arXiv preprint arXiv:2104.14403*, 2021. URL https://arxiv.org/pdf/2104.14403.pdf. 1

[72] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017. URL https://arxiv.org/pdf/1702.04595.pdf. 2, 3, 5

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] . See the Broader Impacts and Limitations
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] . See the Broader Impacts and Limitations section.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] . Code and data are provided in the Supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] . See Supplement A and B.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] . See Footnote 2 and Table captions (e.g. in Table 1).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] . See Supplement A.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] . We cite the original authors contributing each dataset.
   (b) Did you mention the license of the assets? [Yes] See Supplement B.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use publicly available datasets in accordance with their licenses.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] This issue is not applicable in the case of these public datasets.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]