

Interpreting Black Box Models via Hypothesis Testing

Collin Burns
Columbia University
collin.burns@columbia.edu

Jesse Thomason
University of Washington
thomason.jesse@gmail.com

Wesley Tansey
Memorial Sloan Kettering Cancer
Center
tanseyw@mskcc.org

ABSTRACT

In science and medicine, model interpretations may be reported as discoveries of natural phenomena or used to guide patient treatments. In such high-stakes tasks, false discoveries may lead investigators astray. These applications would therefore benefit from control over the finite-sample error rate of interpretations. We reframe black box model interpretability as a multiple hypothesis testing problem. The task is to discover “important” features by testing whether the model prediction is significantly different from what would be expected if the features were replaced with uninformative counterfactuals. We propose two testing methods: one that provably controls the false discovery rate but which is not yet feasible for large-scale applications, and an approximate testing method which can be applied to real-world data sets. In simulation, both tests have high power relative to existing interpretability methods. When applied to state-of-the-art vision and language models, the framework selects features that intuitively explain model predictions. The resulting explanations have the additional advantage that they are themselves easy to interpret.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning; Artificial intelligence.**

KEYWORDS

interpretability, black box, transparency, hypothesis testing, FDR control

ACM Reference Format:

Collin Burns, Jesse Thomason, and Wesley Tansey. 2020. Interpreting Black Box Models via Hypothesis Testing. In *Proceedings of the 2020 ACM-IMS Foundations of Data Science Conference (FODS '20)*, October 19–20, 2020, Virtual Event, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3412815.3416889>

1 INTRODUCTION

When using a black box model to inform high-stakes decisions, one often needs to audit the model. At a minimum, this means understanding which features are influencing the model’s prediction. When the data or predictions are random variables, it may

be impossible to determine the important features without some error. In scientific applications, control over the error rate when reporting significant results is paramount, particularly in the face of the replication crisis [3]. In these cases, the reported “important” features should come with some statistical control on the error rate. This last part is critical: if interpreting a black box model is intended to build trust in its reliability, then the method used to interpret it must itself be reliable, robust, and transparent. This is especially necessary in domains like science and medicine, which hold a high standard for trusting black box predictions.

For example, an oncologist may use a black box model that predicts a personalized course of treatment from tumor sequencing data. For the physician to trust the recommendation, it may come with a list of genes explaining the prediction. Those genes can then be cross-referenced with the research literature to verify their association with response to the recommended treatment. However, gene expression data is highly correlated. If the interpretability method does not consider the dependency of different genes, it may report many false positives. This may lead the oncologist to believe the model is incorrectly analyzing the patient, or, worse, to believe the model identified cancer-driving genes that it actually ignores.

In this paper, we address the need for reliable interpretation by casting black box model interpretability as a multiple hypothesis testing problem. Given a black box model and an input of interest, we test subsets of features to determine which are collectively important for the prediction. Importance is measured relative to the model prediction when features are replaced with draws from an uninformative counterfactual distribution. We develop a framework casting interpretability as hypothesis testing in which we can control the false discovery rate of important features at a user-specified level.

Within this framework, we propose two hypothesis testing methods: the Interpretability Randomization Test (IRT) and the One-Shot Feature Test (OSFT). The first provably controls the false discovery rate (FDR), but is computationally intensive. The second is a fast, approximate test that can be used to interpret models on large datasets. In synthetic benchmarks, both tests empirically control the FDR and have high power relative to other methods from the literature. When applied to state-of-the-art vision and language models, the OSFT selects features that intuitively explain model predictions.

Using these methods, one can also visualize why certain features were selected as important. For example, in Fig. 1 we show interpretations of an image classification by both the OSFT and by LIME [27], a popular black box interpretability method. The ground truth label is “Impala”, and there are two impala in the image. LIME selects just the one on the left as important. In contrast, the OSFT selects only the impala on the right. Because the framework we propose is based on counterfactuals, we can visualize the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

FODS '20, October 19–20, 2020, Virtual Event, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-8103-1/20/10...\$15.00

<https://doi.org/10.1145/3412815.3416889>

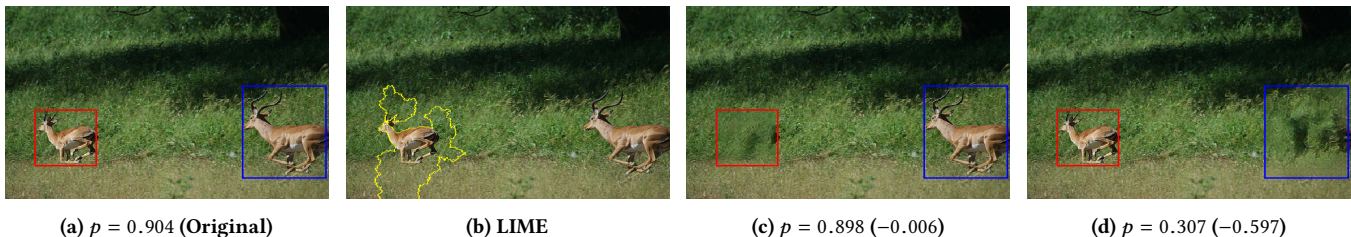


Figure 1: Interpretations by the OSFT, one of the methods we propose, and LIME [27]. The ground truth class is “Impala”. The impala on the right (bounded by blue and replaced by the counterfactual in 1d) was selected as important by the OSFT, while the impala on the left (bounded by red and replaced in 1c) was not. In contrast, LIME selects only the impala on the left as important. For each image, p is the predicted probability of the correct class (Impala). The predicted probabilities on the generated counterfactual inputs reassure us that only the impala on the right had a significant effect on the model output, as selected by the OSFT.

counterfactual inputs and how the model predictions change based on those inputs. Fig. 1 shows that the “Impala” class probability drops significantly when the impala on the right is replaced by an uninformative counterfactual, while it decreases only negligibly when the impala on the left is replaced, suggesting that LIME has identified a feature not used by the model. The ability to manually inspect the counterfactuals is an additional feature of the OSFT that reassures the user of the validity of the interpretation. This example also highlights how it can be misleading to evaluate interpretability methods by visual inspection of the selected features.

2 RELATED WORK

We focus on prediction-level interpretation. Given a black box model and an input, the goal is to explain the model’s output in terms of features of that input.

Interpreting machine learning models. Most methods for interpreting model predictions are based on optimization. Gradient-based methods like Saliency [30] and DeepLift [29] visualize the saliency of each input variable by analyzing the gradient of the model output with respect to the input example. By contrast, black box optimization-based methods do not assume gradient access. These methods include LIME [27], SHAP [24], and L2X [11]. LIME approximates the model to be explained using a linear model in a local region around the input point, and uses the weights of the linear model to determine feature importance scores. SHAP takes a game-theoretic approach by optimizing a kernel regression loss based on Shapley values. L2X selects explanatory features by maximizing a variational lower bound on the mutual information between subsets of features and the model output.

Some existing interpretability methods, like those we present in this paper, are based on counterfactuals. Fong and Vedaldi [14] generate a saliency map by optimizing for the smallest region that, when perturbed (such as by blurring or adding noise), substantially drops the class probability. However, the perturbations used lead to counterfactual inputs that are outside the training distribution. Given the lack of robustness of many modern machine learning models [18], it is unclear how to interpret the resulting explanations. Cabrera et al. [8] introduce an interactive setup for interpreting image classifiers in which users select regions of a given image to

inpaint using a deep generative model. Inpainting deletes original image regions and fills them in with a plausible, learned counterfactual. The system then visualizes the change in probabilities for the top classes. Chang et al. [10] similarly use inpainting models but, like Fong and Vedaldi [14], use this to generate a saliency map without any theoretical guarantees.

Optimization-based approaches generally require defining a penalized loss function. Tuning the hyperparameters of these functions is done by visual inspection of the results, and this interactive tuning is often misleading [1, 22]. Optimization may also overestimate the importance of some variables due to the winner’s curse [33]. That is, by looking at the impact of variables and selecting for those with high impact, the post-selection assessment of their importance is biased upward. This phenomenon is known in statistics as post-selection inference and requires careful analysis of the penalized likelihood to derive valid inferences [21]. By taking a multiple hypothesis testing approach, the methods proposed in this paper avoid this issue.

Multiple hypothesis testing and FDR control. In multiple hypothesis testing (MHT), $\mathbf{z} = (z_1, \dots, z_N)$ are a set of observations of the outcomes of N experiments. For each observation, if the experiment had no effect ($h_i = 0$) then z_i is distributed according to a null distribution $\pi_0^{(i)}(z)$; otherwise, the experiment had some effect ($h_i = 1$) and z_i is distributed according to some unknown alternative distribution. The null hypothesis for every experiment is that the test statistic was drawn from the null distribution: $H_0^{(i)} : h_i = 0$. For a given prediction \hat{h}_i , we say it is a true discovery if $\hat{h}_i = 1 = h_i$ and a false discovery if $\hat{h}_i = 1 \neq h_i$. Let $\mathcal{S} = \{i : h_i = 1\}$ be the set of observations for which there was some effect (true positives) and $\hat{\mathcal{S}} = \{i : \hat{h}_i = 1\}$ be the set of reported discoveries. The goal in MHT is to maximize the true positive rate, also known as *power*, $\text{TPR} := \mathbb{E} \left[\frac{\#\{i:i \in \hat{\mathcal{S}} \cap \mathcal{S}\}}{\#\{i:i \in \hat{\mathcal{S}}\}} \right]$, while controlling an error metric; here we focus on controlling the false discovery rate, $\text{FDR} := \mathbb{E} \left[\frac{\#\{i:i \in \hat{\mathcal{S}} \setminus \mathcal{S}\}}{\#\{i:i \in \hat{\mathcal{S}}\}} \right]$. Methods that control FDR ensure that reported discoveries are reliable by guaranteeing that, on average, no more than a small fraction of them are false positives. In the context of black box model interpretation, we seek to control the FDR in

the reported set of important features that contributed toward a model’s prediction.

Conditional independence testing and knockoffs. A closely related task to model interpretation is testing for conditional independence between a feature and a label. The null hypothesis is that the j^{th} feature contains no predictive information for the ground truth label, conditioned on all other features, $H_0: X_j \perp\!\!\!\perp Y \mid X_{-j}$. The model-X knockoffs framework [9] provides finite-sample control of the FDR when testing for conditional independence between multiple features and a label. We leverage the knockoff filter in one of our proposed procedures. However, we sample from a different counterfactual distribution and test a different null hypothesis. Applying the knockoff filter to this alternative counterfactual distribution controls the false discovery rate of a null hypothesis that is more meaningful for interpretability than conditional independence.

Simply applying knockoffs or any other conditional independence procedure (e.g., [6, 28, 42]) is not sufficient for model interpretation for a subtle reason. The null hypothesis in these methods is that two random variables X_j and Y are independent conditioned on X_{-j} . In the model interpretation task, we replace Y with \hat{Y} , the output from the predictive model. For a dense predictive model like a neural network, \hat{Y} is a deterministic function of all of the X variables, so changing the value of X_j deterministically changes \hat{Y} . Thus, the null hypothesis will always be false because any change to X_j will numerically alter \hat{Y} . We introduce a carefully chosen null hypothesis that, unlike conditional independence, accurately captures the type of interpretability we focus on: a set of features having an important effect on the model given the remaining features.

3 METHODOLOGY

We consider a feature important if its impact on the model output is *surprising* relative to a counterfactual. We formalize this as a hypothesis testing problem. For each feature, we test whether the observed model output would be similar if the feature was drawn from some uninformative counterfactual distribution. Tests that control the corresponding FDR will then only select features whose effect on the model output is sufficiently extreme with respect to this counterfactual distribution. We focus on contextual importance: we are interested in whether a feature contributes to a prediction in the context of the other features.

Suppose we want to understand the output of a model f given an input $x \in \mathbb{R}^d$ that was sampled from some distribution $P(X)$. For $S \subseteq [d] := \{1, \dots, d\}$, we let X_S denote X restricted to the set S , and let X_{-S} denote X restricted to the features not in S .

DEFINITION 1. *Suppose $T(\cdot)$ is a test statistic, $S \subseteq [d]$, and $Q(X_S|X_{-S})$ is some conditional distribution. Let $T_P(f(X))$ be the true distribution of $T(f(x))$, $x \sim P(X)$, and let $T_{Q|x_{-S}}(f(X))$ be the distribution of $T(f(\tilde{x}))$, where $\tilde{x} = (\tilde{x}_S, x_{-S})$ and $\tilde{x}_S \sim Q(X_S|X_{-S} = x_{-S})$. The null hypothesis, H_0 , is that $T_P(f(X))$ is stochastically less than $T_{Q|x_{-S}}(f(X))$,*

$$H_0: T(f(x)) \sim T_P(f(X)) \leq T_{Q|x_{-S}}(f(X)). \quad (1)$$

(A random variable Y is stochastically less than a random variable Z if for all $u \in \mathbb{R}$, $\Pr[Y > u] \leq \Pr[Z > u]$.) Given $x \in \mathbb{R}^d$, a model f , and a subset of features $S \subseteq [d]$, we say that x_S is important with

respect to the test statistic $T(\cdot)$ and the conditional $Q(X_S|X_{-S})$ if H_0 is false.

The null hypothesis in Eq. (1) covers a family of null distributions for the observed test statistic. Informally, it includes all distributions that put more mass on smaller (i.e., less extreme) statistics than samples from Q would. The distribution corresponding to the pointwise equality null hypothesis,

$$H_0: T(f(X)) \sim T_P(f(X)) \stackrel{d}{=} T_{Q|x_{-S}}(f(X)) \quad (2)$$

will therefore put the most mass on large test statistics of any member of the null family. Consequently, any test statistic for the point null is a conservative statistic for Eq. (1), the familywise null. We use the point null as a proxy for the familywise null, as we can only sample from the former, $T_{Q|x_{-S}}(f(X))$.

The definition above applies to any conditional distribution $Q(X_S|X_{-S})$, but it is only a useful notion of interpretability for some distributions. For example, the generated counterfactuals, $\tilde{X} = (\tilde{X}_S, x_{-S})$, should lie in the support of the true distribution, $P(X)$. Counterfactuals should lie in the support of $P(X)$ because the model has only been trained on inputs from $P(X)$. As work on robustness and adversarial examples illustrates [16, 18], model behavior on out-of-distribution inputs can be counterintuitive, making the definition of importance with respect to such a distribution potentially misleading.

The final choice of counterfactual and model for Q will be application dependent. We delay further discussion of specific counterfactuals to Section 4 and next present two general methods for use with any counterfactual distribution.

3.1 The Interpretability Randomization Test

The point null distribution in Eq. (2) will often not be available in closed form, but if we can sample from $Q(X_S|X_{-S})$ then we can repeatedly sample new inputs, calculate a test statistic, and compare it to the original test statistic. Randomization tests build an empirical estimate of the likelihood of observing a test statistic as extreme as that observed under the null distribution. Algorithm 1 details the Interpretability Randomization Test. Adding one to the numerator and denominator ensures that this is a valid p -value for finite samples from H_0 [13], meaning it is stochastically greater than $U(0, 1)$.

When testing multiple features, controlling the error rate requires applying a multiple hypothesis testing correction procedure. The choice of MHT-Correct in Algorithm 1 depends on the goal of inference and the dependence between features. We focus on controlling the FDR via Benjamini-Hochberg (BH) [4], which controls the FDR when the tests are independent or in a large class of positive dependence [5]. We found this robustness to be sufficient to control the FDR empirically. For FDR control under arbitrary dependence, one can instead use the Benjamini-Yekutieli procedure [5].

3.2 The One-Shot Feature Test

The IRT requires repeatedly sampling counterfactuals, which can be computationally expensive. For instance, in the image and language case studies in Section 4, we generate counterfactuals from deep conditional models. Running these models thousands of times per feature is intractable. For these cases, we propose the One-Shot

Algorithm 1 Interpretability Randomization Test (IRT)

Require: (features (x_1, \dots, x_d) , trained model f , conditional model $Q(X_S|X_{-S})$, test statistic T , target FDR threshold α , subsets of features to test $S_1, \dots, S_N \subset [d]$, number of draws K)

- 1: Compute model output $\hat{y} \leftarrow f(x)$
- 2: Compute test statistic $t \leftarrow T(\hat{y})$
- 3: **for** $i \leftarrow 1, \dots, N$ **do**
- 4: **for** $k \leftarrow 1, \dots, K$ **do**
- 5: Sample $\tilde{x}_{S_i} \sim Q(X_{S_i}|X_{-S_i} = x_{-S_i})$
- 6: Compute model output $\tilde{y}^{(k)} \leftarrow f((\tilde{x}_{S_i}, x_{-S_i}))$
- 7: Compute the test statistic $\tilde{t}^{(k)} \leftarrow T(\tilde{y}^{(k)})$
- 8: **end for**
- 9: Compute the p-value

$$\hat{p}_i = \frac{1}{K+1} \left(1 + \sum_{k=1}^K \mathbb{1} \left[t \leq \tilde{t}^{(k)} \right] \right)$$

- 10: **end for**
- 11: $\tau = \text{MHT-Correct}(\alpha, \hat{p}_1, \dots, \hat{p}_K)$
- 12: **Return** discoveries at the α level: $\{i: \hat{p}_i \leq \tau\}$

Feature Test (OSFT), which requires only a single sample from the conditional distribution.

The OSFT is inspired by the recently-proposed model-X knockoffs technique for conditional independence testing [9]. By using the knockoff filter for selection, the OSFT provably controls the FDR when the features or test statistics are independent.

THEOREM 1. *Let $z^{(i)} = t(f(x)) - t(f(\tilde{x}^{(i)}))$, where $\tilde{x}^{(i)} = (\tilde{X}_i, x_{-i})$, and $\tilde{X}_i \sim Q(X_i|X_{-i} = x_{-i})$. If the $z^{(i)}$ are independent, then rejecting the null hypotheses in the set $\{H_0^{(i)}: z^{(i)} \geq z^*\}$ controls the FDR of the point null given in Equation (2) at level α , if z^* is such that*

$$\frac{1 + |\{z^{(i)} \leq -z^*: i \in [N]\}|}{|\{z^{(i)} \geq z^*: i \in [N]\}|} \leq \alpha.$$

Proof. The selection procedure in the OSFT and assumption on z^* are the same as for the knockoffs multiple testing procedure [2, 9]. As [9] note, FDR control using the knockoffs selection procedure is guaranteed at the α level as long as the sign of the difference statistics $z^{(i)}$ are i.i.d. coin flips under the null (following Theorems 1 and 2 of [2]). Under the point null for the i th feature, $\tilde{t}^{(i)} \stackrel{d}{=} t$. The distribution of $z^{(i)}$ under the null is therefore symmetric about the origin, so that the sign of every $z^{(i)}$ is indeed an independent coin flip. The claim then follows from [9]. \square

Counterfactual draws used in the OSFT are valid knockoffs only when all features are independent, limiting strict FDR control to the independent feature case. However, when evaluating multiple independent samples, such as multiple images in a dataset, counterfactual draws do yield a slightly looser bound on the FDR.

COROLLARY 1. *For M independent samples with at most N feature subsets per sample, rejecting the null hypotheses in the set $\{H_0^{(i)}: z^{(i)} \geq z^*\}$ as in Theorem 1 controls the FDR at level $N\alpha$.*

As with the IRT using the BH correction procedure, in practice the OSFT controls the FDR in a wider class of scenarios than theoretically guaranteed (see Table 1). The OSFT is given in Algorithm 2.

3.3 Two-sided test statistics

Some choices of the test statistic, $T(\cdot)$, may be more appropriate for certain tasks and may have higher power than other choices.

Algorithm 2 One-Shot Feature Test (OSFT)

Require: (features (x_1, \dots, x_d) , trained model f , conditional model $Q(X_S|X_{-S})$, test statistic T , target FDR threshold α , subsets of features to test $S_1, \dots, S_N \subset [d]$)

- 1: Compute test statistic $t \leftarrow T(f(x_1, \dots, x_d))$
- 2: **for** $i \leftarrow 1, \dots, N$ **do**
- 3: Sample $\tilde{x}_{S_i} \sim Q(X_{S_i}|X_{-S_i} = x_{-S_i})$
- 4: Compute model output $\tilde{y}^{(i)} \leftarrow f((\tilde{x}_{S_i}, x_{-S_i}))$
- 5: Compute the test statistic, $\tilde{t}^{(i)} \leftarrow T(\tilde{y}^{(i)})$
- 6: Compute the difference statistic, $z^{(i)} \leftarrow t - \tilde{t}^{(i)}$
- 7: **end for**
- 8: $z^* \leftarrow \underset{z}{\operatorname{argmin}} \left[\frac{1 + |\{z^{(i)} \leq -z: i \in [N]\}|}{|\{z^{(i)} \geq z: i \in [N]\}|} \leq \alpha \right]$
- 9: **Return** discoveries at the α level: $\{i: z^{(i)} \geq z^*\}$

Two classical statistics are one-sided and two-sided tail probabilities. One-sided tests have a preferred direction of testing, while two-sided tests consider both tails of the null distribution. In the one-sided case, testing for an increase in output can be done by making the test statistic the identity, $T(Y) = Y$. A two-sided IRT statistic requires only modifying Algorithm 1 to look at both tails of the distribution of \tilde{t} . However, the OSFT has no explicit null distribution for each sample. In this case, we can still perform a two-sided test by drawing an extra null variable as a centering sample: $\tilde{X}_i \sim Q(X_i|X_{-i} = x_{-i})$, $\tilde{Y} = f(\tilde{X}_i, x_{-i})$, $T(Y) = (Y - \tilde{Y})^2$. This turns the one-shot procedure into a two-shot procedure. Two-sided test statistics for higher-order moments can be developed by analogously increasing the number of draws in the OSFT.

4 EXPERIMENTS

We first evaluate the IRT and OSFT in a number of synthetic setups and show that they have high power relative to six strong baseline methods: LIME [27], SHAP [24], L2X [11], Saliency [30], DeepLIFT [29], and Taylor [11]. We also verify that the IRT and OSFT successfully control the FDR at the target threshold in these settings. We then apply the OSFT to explain the predictions of a deep image classifier on ImageNet and a deep text classifier on movie review sentiment and find that the method tends to select features that intuitively explain the model predictions. Additional experimental details are provided in the appendix, and we will publicly release our code.

4.1 Synthetic Benchmark

To compare the IRT and OSFT to existing methods, we evaluate how the power varies as a function of the false discovery rate for each method. This requires determining exactly when the null hypothesis is true. In general, this may be infeasible for the null hypothesis given in Eq. (1). However, for certain distributions, the point null given in Eq. (2) is feasible to evaluate. We consider two such distributions: one which has independent features, and the other which has correlated features. We also consider two different models to interpret: a neural network and a discontinuous model.

To empirically evaluate the FDR and TPR, we will use the fact that for each of the following distributions and for both test statistics that we consider, the point null hypothesis, Eq. (2), is equivalent to

$$H_0: f(x) \sim f(X) \stackrel{d}{=} f(\tilde{X}), \quad (3)$$

where again $\tilde{X} = (\tilde{X}_S, x_S)$ and $\tilde{X}_S \sim Q(X_S|x_{-S})$.

Distribution	Model	FDR/TPR			
		IRT		OSFT	
		1-sided	2-sided	1-sided	2-sided
Independent	Discontinuous	0.002 / 0.393	0.002 / 0.392	0.006 / 0.836	0.006 / 0.833
Independent	Neural Net	0.139 / 0.979	0.137 / 0.913	0.212 / 0.962	0.189 / 0.910
Correlated	Discontinuous	0.000 / 0.000	0.000 / 0.000	0.073 / 0.025	0.044 / 0.004
Correlated	Neural Net	0.129 / 0.716	0.130 / 0.641	0.142 / 0.611	0.143 / 0.605

Table 1: Empirical FDR and TPR ($\alpha = 0.2$).

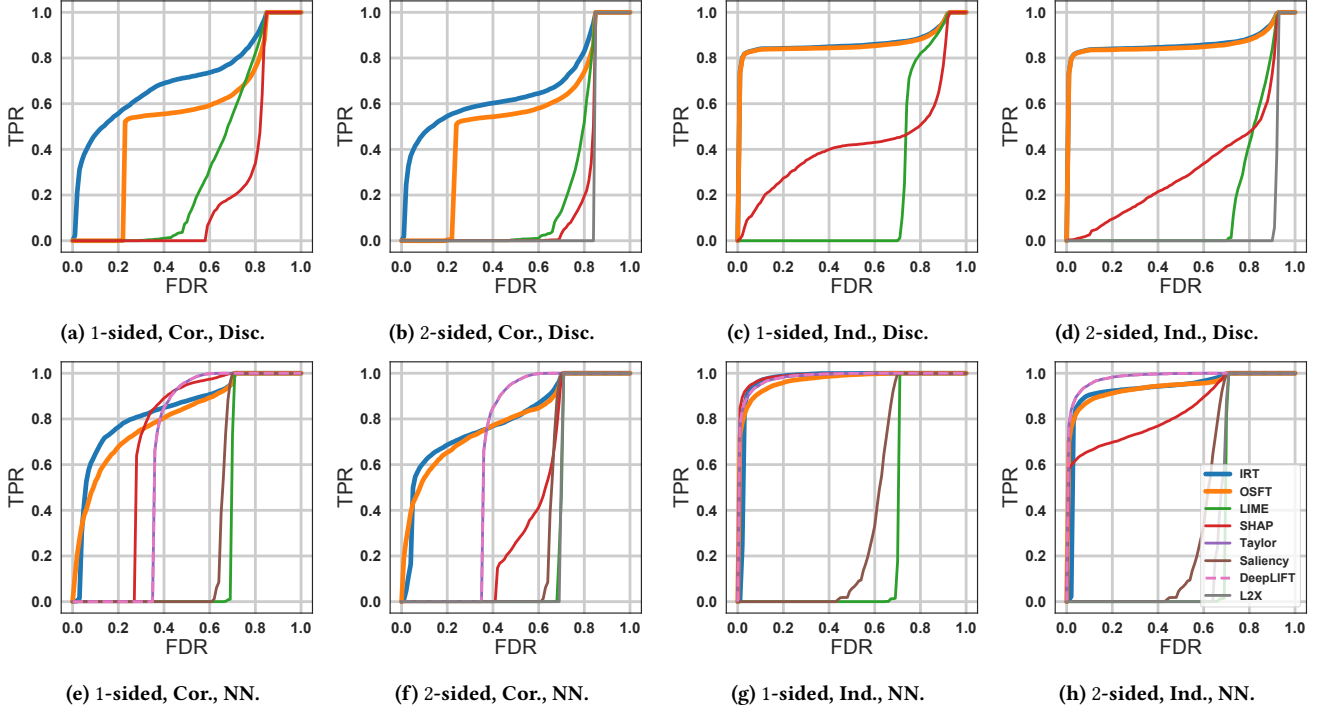


Figure 2: The IRT and OSFT have higher power than the baseline methods in most cases, and have comparable power to the best baseline methods in the remaining cases. The curves were averaged over 10 independent runs.

Inputs. For the independent distribution, for each feature i , with probability $h = 0.3$ we let $X_i \sim \mathcal{N}(4, 1)$, and with probability $1 - h$, $X_i \sim \mathcal{N}(0, 1)$. We then let $Q(X_i|X_{-i})$ be $\mathcal{N}(0, 1)$. For the correlated distribution, for each feature i , with probability $h = 0.3$, $X_i \sim \mathcal{N}(4, 1)$, and with probability $1 - h$, $X_i \sim \mathcal{N}(m, 1)$, where $m = \sum_{j=1}^{i-1} \beta_j x_j$ and where $\beta_j \sim \mathcal{N}(0, \frac{1}{16})$ for each feature j (fixed for all examples). We then let $Q(X_i|X_{-i})$ be $\mathcal{N}(m, 1)$.

Models. The first model is a paired thresholding model. On an input $X = (X_1, \dots, X_{2p}) \in \mathbb{R}^{2p}$, the model output is defined as

$$f(X) = \sum_{i=1}^p w_i \mathbf{1} [|X_i| \geq t \wedge |X_{i+p}| \geq t], \quad (4)$$

for $w \in \mathbb{R}^p$ and $t \geq 0$. We let $w_i = 0.5 + v_i$, $v_i \sim \text{Gamma}(1, 1)$, and fix $t = 3$.

For each feature $i \in [2p]$, the null hypothesis

$$H_0: f(x) \sim f(X) \stackrel{d}{=} f(\tilde{X}), \quad (5)$$

is that $\hat{y} = f(x)$ was sampled from the distribution $f(\tilde{X}^{(i)})$ where $\tilde{X}^{(i)} = (\tilde{X}_i, x_{-i})$ and $\tilde{X}_i \sim Q(X_i|x_{-i})$. For either data distribution above, when $i \in [p]$ this is false if and only if $|x_{i+p}| \geq t$ so that feature i can affect the model output at all, and x_i was sampled from the “interesting” distribution $\mathcal{N}(4, 1)$. Otherwise, by construction, x_i must have been sampled from $Q(X_i|X_{-i})$, in which case the null would be true. Similarly, for each feature $i \in \{p+1, \dots, 2p\}$, the null hypothesis is false if and only if $|x_{i-p}| \geq t$ and x_i was sampled from $\mathcal{N}(4, 1)$. We set $p = 50$, so the number of parameters is 100.

The discontinuous model can only be interpreted by gradient-free interpretability methods. In order to compare our approach to methods that only apply to neural networks (e.g., [29]) or differentiable models (e.g., [30]), we also consider the following setup

that mirrors that of Chen et al. [11]. We let $Y := \sum_{i=1}^d |X_i|$ be the ground truth response variable, with $d = 25$, and train a two-layer neural network to near-zero test error with this response as the label. Given the test error, we can assume that the network has successfully learned which features are important for the model. We then interpret the trained network. If the network indeed learned the model correctly, then the feature x_i is important if and only if it was sampled from the interesting distribution, $\mathcal{N}(4, 1)$. In particular, each feature is always used by the model. Hence, if x_i was sampled from $\mathcal{N}(4, 1)$, then $f(x)$ was sampled from a different distribution than $f(\bar{X})$, so that the null in Eq. (5) is false.

Comparison. For the discontinuous model, we compare against three other black box interpretability methods: LIME [27], SHAP [24], and L2X [11].

- **LIME** [27] builds a linear approximation of the predictive model and uses the coefficients as an importance weights.
- **SHAP** [24] takes a game theoretic approach to importance (Shapley values).
- **L2X** [11] optimizes a variational lower bound on the mutual information between the label and each feature.

For interpreting the neural network, we additionally compare against three methods for interpreting deep learning models: Saliency [30], DeepLIFT [29], and another strong baseline method called Taylor [11]. Taylor computes feature values by multiplying the value of each feature by the gradient of the output with respect to that feature. Note that these methods require access to model gradients, and thus do not perform black box interpretation.

No other methods for black box model interpretation, including LIME, SHAP, and L2X, enable error rate control. We adapted each method where possible by considering how the FPR, FDR, and TPR change as each method smoothly increases the number of features selected. For LIME and SHAP, we consider one-sided and two-sided tests differently. For one-sided tests, we specifically test whether a feature contributes positively to the output, which corresponds to selecting the largest feature values. For two-sided tests we check whether a feature contributes to the output at all, corresponding to the magnitude of the values. Since L2X selects features that are generally “important”, we only compare it with the other methods in two-sided experiments. In evaluating power under FDR control, we calculate TPR for the baseline methods at the highest FDR below the target threshold—that is, we overestimate power by assuming knowledge of the exact FDR cutoff. L2X directly selects k features to explain a prediction, where k is treated as a hyperparameter. The remaining methods output feature values corresponding to how large of an effect each feature had on the given input. To compare these to the IRT and OSFT, which automatically choose a number of features to select as important, we suppose that these methods are able to control the FDR at a particular level, and measure the true positive rate at that level. Specifically, we plot how the empirical FDR and TPR change as each method increases the number of features it selects. Because the FDR is not necessarily monotonic as the number of selected features increases, for each FDR level we take the maximum TPR achieved for which the FDR is controlled at the specified level.

We consider one-sided and two-sided variants for the feature value methods. For the one-sided test, we track how the TPR and

FDR vary as the k features with the largest values are selected, for increasing k . For the two-sided variant, we instead select the k features with the largest absolute values. On the other hand, L2X directly selects features that are broadly relevant to the output of the model. This limits L2X to only the two-sided case. We use the default settings for each method. For the IRT, we used $K = 100$ permutations and the same two-sided test statistic as for the OSFT.

Results. Fig. 2 shows the TPR of each method as a function of the FDR, averaged over 10 independent runs with 100 test samples each. The IRT and OSFT have higher power than the baseline methods for FDR levels of interest, except when interpreting the neural network with independent features. In that case, both methods are still competitive with the best baseline methods.

An advantage of the IRT and OSFT not accounted for in Fig. 2 is that they can automatically select features at a given FDR threshold α . To verify that they control the FDR and have high power, in Table 1 we show the FDR and TPR of both methods for each setting described above, where we set $\alpha = 0.2$ (other reasonable choices of α , such as 0.05, give qualitatively similar results). We find that both methods indeed nearly always control the FDR below the target level of 0.2, and often by a large margin. An exception was the OSFT when interpreting the neural network with independent features using the one-sided test. In that case, the empirical FDR was 0.212, barely above the target level. Moreover, both methods usually have reasonably high power; the one notable exception was when interpreting the discontinuous model with correlated inputs. In the vision and language applications we describe next, the OSFT has high power.

4.2 Interpreting a Deep Image Classifier

We next apply the OSFT to interpreting Inception v3 [32], a deep image classifier. We used the pretrained model in the `torchvision` package. As the conditional distribution, $Q(X_S|X_{-S})$, we use a state-of-the-art generative inpainting model [40]. Inpainting models replace subsets of pixels with counterfactuals that are often reasonable proxies for background pixels. We define the model output to be the logits for the predicted class, and use the one-sided statistic. We test subsets of features corresponding to boxes for simplicity.

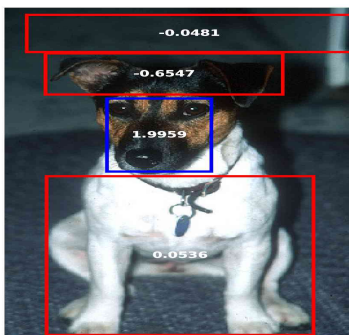
We study two feature selection procedures for choosing candidate patches of pixels to test. The first approach is choosing patches manually by selecting bounding boxes around objects, parts of objects, and parts of the background. This mimics how a pathologist may use such a system to audit predicted diagnoses. For large-scale auditing, selecting regions by hand is intractable. For these scenarios, we explore using an object detector to select patches automatically. See Appendix B for details. In general, pixel feature subsets can be selected in any way, as long as they are non-overlapping, and the best method for doing so will be application dependent.

We applied the OSFT to 50 ImageNet images, some of which were taken from [14] for comparison. At an FDR threshold of $\alpha = 0.2$, 72 of the 222 manually selected patches (about 32%) were selected as important, and 50 of the 169 automatically selected patches (about 30%) were selected as important.

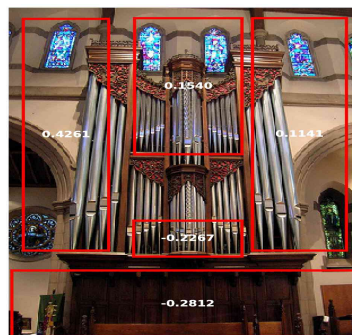
Results. Figures 3 and 4 give representative images and the patches that were tested for each of them. The bounding box color indicates



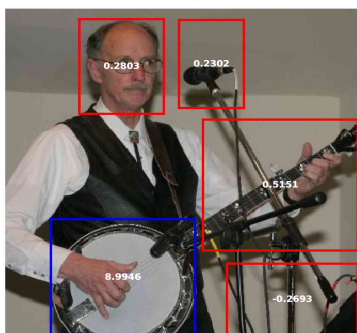
(a) Unicycle ($p = 0.996$)



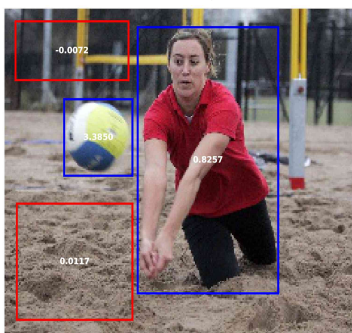
(b) Toy Terrier ($p = 0.927$)



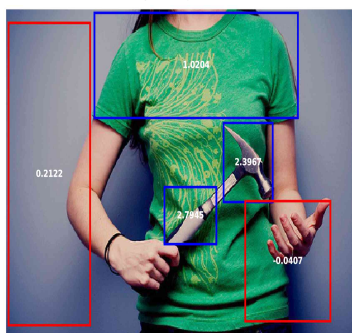
(c) Organ ($p = 0.959$)



(d) Banjo ($p = 0.997$)

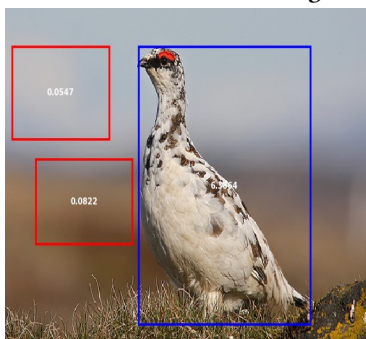


(e) Soccer ball ($p = 0.465$)



(f) Hammer ($p = 0.695$)

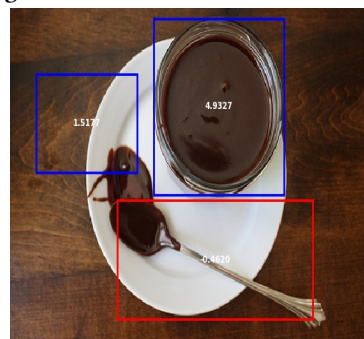
Figure 3: Examples corresponding to manually selected bounding boxes.



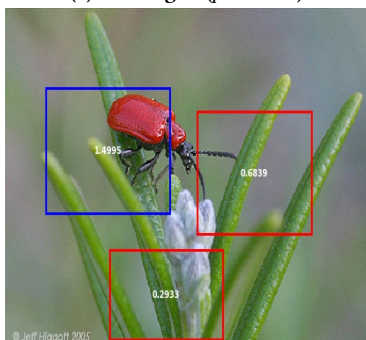
(a) Ptarmigan ($p = 0.871$)



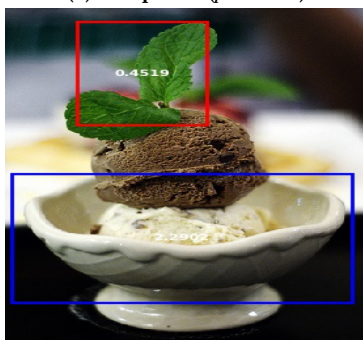
(b) Saxophone ($p = 0.993$)



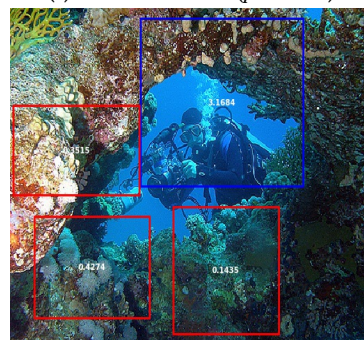
(c) Chocolate sauce ($p = 0.995$)



(d) Leaf beetle ($p = 0.950$)



(e) Ice cream ($p = 0.923$)



(f) Scuba diver ($p = 0.945$)

Figure 4: Examples corresponding to automatically selected bounding boxes.

Label	Model	Review
Neg	Neg	Stay away from this movie! It is terrible in every way. Bad acting, a thin recycled plot and the worst ending in film history. Seldom do I watch a movie that makes my adrenaline pump from irritation, in fact the only other movie that immediately springs to mind is another “people in an aircraft in trouble” movie (Airspeed). Please, please don’t watch this one as it is utterly and totally pathetic from beginning to end. Helge Iversen
Pos	Pos	All i can say is that, i was expecting a wick movie and “Blurred” surprised me on the positive way. Very nice teenager movie. All this kinds of situations are normal on school life so all i can say is that all this reminded me my school times and sometimes it’s good to watch this kind of movies, because entertain us and travel us back to those golden years, when we were young. As well, lead us to think better in the way we must understand our children, because in the past we were just like they want to be in the present time. Try this movie and you will be very pleased . At the same time you will have the guarantee that your time have not been wasted.
Pos	Neg	Not all movies should have that predictable ending that we are all so use to, and it’s great to see movies with really unusual twists. However with that said, I was really disappointed in l’apartment’s ending . In my opinion the ending didn’t really fit in with the rest of the movie and it basically destroyed the story that was being told. You spend the whole movie discovering everyone and their feelings but the events in the final 2 minutes of the movie would have impacted majorly on everyones character but the movie ends and leaves it all too wide open . Overall though this movie was very well made, and unlike similar movies such as Serendipity all the scenes were believable and didn’t go over the top.
Neg	Pos	This is one entertaining flick . I suggest you rent it, buy a couple quarts of rum, and invite the whole crew over for this one. My favorite parts were. 1. the gunfights that were so well choreographed that John Woo himself was jealous,. 2. The wonderful special effects. 3. the Academy Award winning acting and. 4. The fact that every single gangsta in the film seemed to be doing a bad “Scarface” impersonation. I mean, Master P as a cuban godfather! This is groundbreaking territory . And with well written dialogue including lines like “the only difference between you and me Rico, is I’m alive and your dead ,” this movie is truly a masterpiece. Yeah right.

Table 2: Text classifier sentiment predictions and word importance using the OSFT. We selected two texts each where the model prediction agrees and disagrees with the gold label. We tested all words as features, and words selected by the OSFT are highlighted.

whether the patch was found to be important (blue) or not (red). The value of the difference statistic is printed inside each patch. Intuitively, the bounding boxes corresponding to the ground truth labels are often selected.

4.2.1 Sensitivity Analysis. We investigated how sensitive the OSFT for image classification is to perturbations of the selected bounding box. This is also closely related to sensitivity to the conditional model Q ; this is equivalent to slightly perturbing Q if you were to restrict to a slightly larger subset of features x_S containing both the original features and the perturbed features.

More precisely, for each automatically selected bounding box, we perturbed the bounding box in four ways: by shifting it to the left and up by one pixel each, to the right and up by one pixel each, to the left and down by one pixel each, and to the right and down by one pixel each. Including the original set of selected boxes, this gives us five sets of bounding boxes and corresponding generated images. We then ran the OSFT for each of these five sets then look at the average pair-wise Intersection over Union (IOU) of the selected features: specifically, the IOU for a pair of dataset-wide selections is the size of the intersection (i.e., number of bounding boxes selected by both) divided by the size of the union (i.e., the number of bounding boxes selected by either). For the OSFT, the resulting average pair-wise IOU was 0.749, indicating some but not substantial robustness to small perturbations.

4.2.2 Counterintuitive Selections. Moreover, we investigated some of the counterintuitive feature selections made by the OSFT, and found that in many cases they were due to poorly generated counterfactuals. We present an example of this in Fig. 5. Fig. 5c is unsurprisingly selected as important because it involves the features corresponding to an airship, which is the true class. We can again verify this by looking at the corresponding counterfactual, which is realistic. In contrast, in Fig. 5b, the features correspond to an arbitrary part of the sky. However, the generated counterfactual is unrealistic. That we are able to easily visualize and verify the output of the interpretability method is an advantage of the framework we propose, even for cases like this in which it produces an uninformative interpretation because of an imperfect generative model.

4.3 Interpreting a Deep Text Classifier

We also apply the OSFT to interpret a Bidirectional Encoder Representations from Transformers (BERT) model [12] for text classification. BERT and its ancestors continue to set new state of the art performances in text classification on the GLUE benchmark [35]. BERT learns multiple layers of attention instead of a flat attention structure [34], making visualization of its internals complicated. Interpretations based on attention alone in these models may not

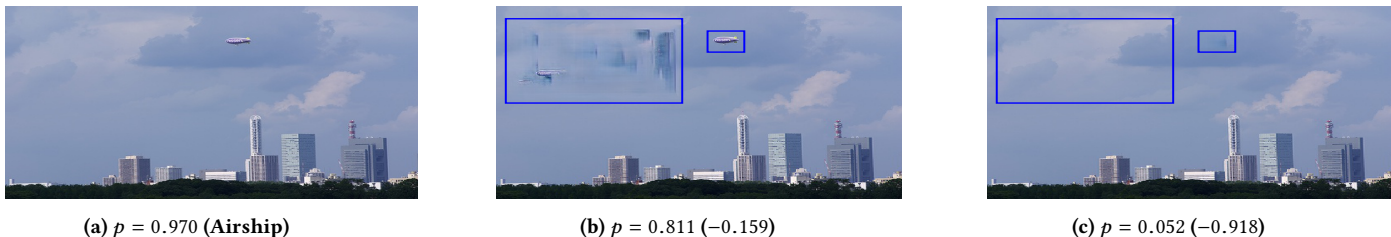


Figure 5: An interpretation illustrating both the benefits and limitations of the approach presented in this work. The subsets of features replaced in both Figs. 5b and 5c were selected as important. Fig. 5b illustrates how a poor counterfactual can lead to an unwarranted selection by the interpretability method, while Fig. 5c illustrates a reasonable counterfactual that shows the importance of the corresponding features in an interpretable way.

be reliable [7, 19]. We posit that a post-hoc, black box interpretability method is more appropriate for understanding predictions by transformer models like BERT. Infilling is performed by masking a word token in a sentence, then predicting what word should be at that position using BERT. This masked language modelling task is how BERT is trained.

We evaluate on the Large Movie Review Dataset (LMRD) [25], a corpus of movie reviews labeled as having either positive or negative sentiment and split into 25k training and 25k testing examples. We train two BERT-based models: one for predicting sentiment (the model to interpret) and another to approximate the conditional distribution $Q(X_S|X_{-S})$. We set the FDR threshold α to 0.15 and test on 1000 randomly selected reviews from the test set, for a total of 95518 word features tested. We used the two-sided test statistic, drawing two samples per WordPiece feature. About 4% of the words were selected by the OSFT.

Results. Table 2 gives examples of correct and incorrect model predictions. Words selected as important by the OSFT are highlighted. Intuitively, we find that high-sentiment words like *terrible*, *pleased*, *disappointed*, and *wonderful* tend to be selected as important. Additional model details can be found in Appendix C.

5 DISCUSSION

Scientists need to understand predictions from machine learning models when making decisions. In medicine, a treatment based on a black box prediction could lead to patient harm if the prediction was based on poor evidence or flawed reasoning. In biology, a set of low quality predictions may lead scientists to waste time and funding exploring a potential new drug target that was simply an artifact of the correlation structure of the data. To ensure reliability of models, scientists must be able to audit and confirm their reasoning in a principled manner.

We proposed a general framework for reframing model interpretability as a multiple hypothesis testing problem. The framework mirrors the statistical analysis protocol employed by scientists: the null hypothesis test. Within this framework, we introduced the IRT and the OSFT, two hypothesis testing procedures for interpreting black box models. Both methods enable control of the false discovery rate at a user-specified level.

Limitations and Future Work. The methods proposed in this paper require a way to generate plausible counterfactual inputs while

keeping some features held fixed. Fortunately, this is already feasible for many types of distributions. For example, image inpainting is a subfield of computer vision that has a long history [17] and much recent work (e.g., [31, 36, 38–41]), with plausible infill models available for many domains. Moreover, some deep language models, like BERT, are masked language models: they are trained, in part, to predict masked words. Consequently, to apply the IRT and OSFT to such models does not require a separate conditional model. In scientific and medical applications, the input domain is often even simpler, making it especially feasible to construct an accurate counterfactual model for such applications.

One may be able to automatically ensure that the generated counterfactuals are plausible by using a separate model to assess how realistic it is. For instance, one could use a GAN discriminator for vision tasks, or a separately trained language model for language tasks. One could then filter out unrealistic examples, incorporating expert knowledge as a means of developing a rejection sampler for the counterfactual distribution.

A practical problem is choosing which subsets of features to test. Unlike our framework, this problem is application-dependent. In some cases, it is straightforward to test all features individually, especially if they are low dimensional or easily binned. In other cases, such as in histology and medical imaging, experts can manually select features of interest to test. In generic vision tasks, one can use an object detector or image segmentation model to select proposed regions, as we explore in Section 4.2. In generic language tasks, one can test individual word features, as we explore in Section 4.3, but this can miss features that involve composition. In the future, spans of words may be tested as individual features after extraction from a dependency tree (e.g., “spans of words” in this sentence). In that case, the conditional distribution can be approximated by models like SpanBERT trained through span-based infilling [20]. Further, in tasks involving both language and vision, such as image classification when captions are available, feature testing can be done in *both* modalities, infilling language tokens or image regions, by approximating the multimodal conditional distribution with models like ViLBERT [23].

The IRT and OSFT are less efficient than most popular interpretability methods, which usually require a single forward and backward pass per input. Nevertheless, these methods can still be easily run on a single CPU. More importantly, in domains like science and medicine, the statistical reliability of explanations is

more of a bottleneck than efficiency. In other words, a higher computational budget is the cost of FDR control, but this will often be worthwhile because avoiding false positives is crucial for valid science.

Finally, model interpretability extends beyond feature importance. For instance, when investigating model fairness, one may be interested in how an image classifier changes its prediction if you change the gender or skin color of someone in a photo. Alternatively, one may be interested in testing how much an image classifier relies on texture [15]. For these scenarios, one may be able to construct a style transfer model that changes gender, race, or texture while still remaining in-distribution. Our framework is sufficiently general to answer these interpretability questions. We plan to investigate its application in these domains in future work.

REFERENCES

- [1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity Checks for Saliency Maps. In *Neural Information Processing Systems (NeurIPS)*.
- [2] Rina Foygel Barber and Emmanuel J Candès. 2015. Controlling the false discovery rate via knockoffs. *The Annals of Statistics* 43, 5 (2015), 2055–2085.
- [3] Daniel J Benjamin, James O Berger, Magnus Johannesson, Brian A Nosek, E-J Wagenmakers, Richard Berk, Kenneth A Bollen, Björn Brembs, Lawrence Brown, and Colin Camerer. 2018. Redefine Statistical Significance. *Nature Human Behaviour* 2, 1 (2018), 6.
- [4] Yoav Benjamini and Yoel Hochberg. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)* 57, 1 (1995), 289–300.
- [5] Yoav Benjamini, Daniel Yekutieli, et al. 2001. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics* 29, 4 (2001), 1165–1188.
- [6] Thomas B Berrett, Yi Wang, Rina Foygel Barber, and Richard J Samworth. 2018. The conditional permutation test. *arXiv preprint arXiv:1807.05405* (2018).
- [7] Gino Brunner, Yang Liu, Damián Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2019. On Identifiability in Transformers. *arXiv preprint arXiv:1908.04211* (2019).
- [8] Angel Cabrera, Fred Hohman, Jason Lin, and Duen Horng Chau. 2018. Interactive Classification for Deep Learning Interpretation. *arXiv preprint arXiv:1806.05660* (2018).
- [9] Emmanuel Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. 2018. Panning for gold: ‘Model-X’ knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B* (2018).
- [10] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining Image Classifiers by Adaptive Dropout and Generative In-filling. In *International Conference on Learning Representations (ICLR)*.
- [11] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *International Conference on Machine Learning (ICML)*.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- [13] Eugene Edgington and Patrick Onghena. 2007. *Randomization tests*. Chapman and Hall/CRC.
- [14] Ruth Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *International Conference on Computer Vision (ICCV)*.
- [15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2019. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations (ICLR)*.
- [16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.
- [17] Christine Guillemot and Olivier Le Meur. 2014. Image Inpainting : Overview and Recent Advances. *Signal Processing Magazine, IEEE* 31 (2014), 127–144.
- [18] Dan Hendrycks and Thomas G. Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations (ICLR)*.
- [19] Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.
- [20] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics (TACL)* 8 (2020), 64–77.
- [21] Jason D Lee, Dennis L Sun, Yuekai Sun, and Jonathan E Taylor. 2016. Exact post-selection inference, with application to the lasso. *The Annals of Statistics* 44, 3 (2016), 907–927.
- [22] Zachary Chase Lipton. 2016. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*.
- [23] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Neural Information Processing Systems (NeurIPS)*.
- [24] Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Neural Information Processing Systems (NeurIPS)*.
- [25] 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 *Association for Computational Linguistics: Human Language Technologies (ACL)*. 142–150.
- [26] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767* (2018).
- [27] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.
- [28] Rajat Sen, Karthikeyan Shanmugam, Himanshu Asmani, Arman Rahimzamani, and Sreeram Kannan. 2018. Mimir and Classify: A meta-algorithm for Conditional Independence Testing. *arXiv preprint arXiv:1806.09708* (2018).
- [29] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning (ICML)*. 3145–3153.
- [30] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *International Conference on Learning Representations (ICLR) Workshop*.
- [31] Ecem Sogancioglu, Shi Hu, Davide Belli, and Bram van Ginneken. 2018. Chest X-ray Inpainting with Deep Generative Models. *arXiv preprint arXiv:1809.01471* (2018).
- [32] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [33] Richard H Thaler. 1988. Anomalies: The winner’s curse. *Journal of Economic Perspectives* 2, 1 (1988), 191–202.
- [34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems (NeurIPS)*.
- [35] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv preprint arXiv:1804.07461* (2018).
- [36] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. 2018. Image Inpainting via Generative Multi-column Convolutional Neural Networks. In *Neural Information Processing Systems (NeurIPS)*.
- [37] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, and Klaus Macherey. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [38] Raymond A. Yeh, Chen Chen, Teck-Yian Lim, Mark Hasegawa-Johnson, and Minh N. Do. 2017. Semantic Image Inpainting with Perceptual and Contextual Losses. *arXiv preprint arXiv:1607.07539* (2017).
- [39] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. 2018. Free-Form Image Inpainting with Gated Convolution. *arXiv preprint arXiv:1806.03589* (2018).
- [40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. 2018. Generative Image Inpainting with Contextual Attention. In *Computer Vision and Pattern Recognition (CVPR)*.
- [41] Liuchun Yuan, Congcong Ruan, Haifeng Hu, and Dihou Chen. 2019. Image Inpainting Based on Patch-GANs. *IEEE Access* 7 (2019), 46411–46421.
- [42] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2011. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 804–813.

Algorithm 3 Benjamini–Hochberg (BH) correction

Require: α , empirical p-values $\hat{p}_1, \dots, \hat{p}_K$

- 1: Sort the \hat{p}_i in ascending order, yielding $\hat{p}^{(1)}, \dots, \hat{p}^{(K)}$
 - 2: Compute the largest i such that $\hat{p}^{(i)} \leq \frac{i}{K}\alpha$
 - 3: **Return** $\tau := \hat{p}^{(i)}$
-

A BENJAMINI–HOCHBERG CORRECTION PROCEDURE

First, for the sake of completeness, in Algorithm 3 we provide the Benjamini–Hochberg [4] correction procedure that we use as MHT-Correct for the IRT in all experiments.

B IMAGE EXPERIMENT DETAILS

For automatic bounding box selection of a given image, we first use the YOLOv3 object detector [26] pre-trained on the COCO dataset. This yields a set of bounding boxes and corresponding probabilities. We sorted the bounding boxes in descending order of probability and included each one if it has area that was between 10% and 50% of the area of the entire image and doesn't overlap with any bounding boxes added so far.

Next, we add additional random bounding boxes by repeating the following 100 times: choose the width of the bounding box uniformly at random between a quarter and half the width of the image (and similarly for height) then choose the location of the bounding box to be uniformly at random in the image such that it fits entirely within the dimensions of the image. Finally, keep it if and only if it does not overlap with any added bounding boxes so far.

C LANGUAGE EXPERIMENT DETAILS

We tokenize reviews into WordPieces [37], the sub-word level inputs to the BERT model, and test the significance of each WordPiece. To fit the reviews in memory, we restrict the training set to the 13k reviews that are under 256 WordPieces in length. We tune a pre-trained BERT model to perform sequence classification on this task, achieving 93.1% accuracy at test time. The 1000 sampled reviews from the test set to interpret via OSFT are chosen from among those under 256 WordPieces in length. For all pretrained BERT models, we tune from BERT-Base-Cased and use the framework provided by <https://github.com/huggingface/pytorch-pretrained-BERT> to train both the classification and conditional models.