Reliable Post hoc Explanations: Modeling Uncertainty in Explainability

Anonymous Author(s)
Affiliation
Address
email

Abstract

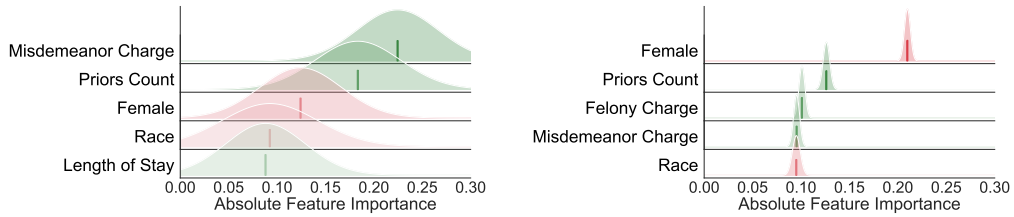
As black box explanations are increasingly being employed to establish model credibility in high stakes settings, it is important to ensure that these explanations are accurate and reliable. However, prior work demonstrates that explanations generated by state-of-the-art techniques are inconsistent, unstable, and provide very little insight into their correctness and reliability. In addition, these methods are also computationally inefficient, and require significant hyper-parameter tuning. In this paper, we address the aforementioned challenges by developing a novel Bayesian framework for generating local explanations along with their associated uncertainty. We instantiate this framework to obtain Bayesian versions of LIME and KernelSHAP which output credible intervals for the feature importances, capturing the associated uncertainty. The resulting explanations not only enable us to make concrete inferences about their quality (e.g., there is a 95% chance that the feature importance lies within the given range), but are also highly consistent and stable. We carry out a detailed theoretical analysis that leverages the aforementioned uncertainty to estimate how many perturbations to sample, and how to sample for faster convergence. This work makes the first attempt at addressing several critical issues with popular explanation methods in one shot, thereby generating consistent, stable, and reliable explanations with guarantees in a computationally efficient manner. Experimental evaluation with multiple real world datasets and user studies demonstrate that the efficacy of the proposed framework.

1 Introduction

As machine learning (ML) models get increasingly deployed in domain such as healthcare and criminal justice, it is important to ensure that decision makers have a clear understanding of the behavior of these models. However, ML models that achieve state-of-the-art accuracy are typically complex *black boxes* that are hard to understand. As a consequence, there has been a surge in post hoc techniques for explaining black box models [1–10]. Most popular among these techniques are local explanation methods which explain complex black box models by constructing interpretable local approximations (e.g., LIME [2], SHAP [4], MAPLE [11], Anchors [1]). Due to their generality, these methods are being leveraged to explain a number of classifiers including deep neural networks and ensemble models in a variety of domains such as law, medicine, and finance [12, 13].

Existing local explanation methods, however, suffer from several drawbacks. Explanations generated using these methods may be unstable [14–18], i.e., negligibly small perturbations to an instance can result in substantially different explanations. These methods are also inconsistent [19] i.e., multiple runs on the same input instance with the same parameter settings may result in vastly different explanations. There are also no reliable metrics to ascertain the quality of the explanations output by these methods. Commonly used metrics such as explanation fidelity rely heavily on the

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



(a) Explanation computed with 100 perturbations

(b) Explanation with 2000 perturbations

Figure 1: **Example explanations** on for an instance from the COMPAS dataset, where vertical lines indicate the feature importance by LIME (red is negative effect, green is positive) and the shaded region visualizes the uncertainty estimated by BayesLIME. While LIME produces very different and contradictory feature importance for different number of perturbations (1a and 1b), BayesLIME provides more context. The overlapping uncertainty intervals in the explanation computed with 100 perturbations (1a) indicate that it is unclear which feature is the most important. However, the tighter uncertainty intervals in the explanation computed with 2K perturbations (1b) clearly indicates that `Female` is the most important.

implementation details of the explanation method (e.g., the perturbation function used in LIME) and do not provide a true picture of the explanation quality [20]. Furthermore, there exists little to no guidance on determining the values of certain hyperparameters that are critical to the quality of the resulting local explanations (e.g., number of perturbations in case of LIME). Local explanation methods are also computationally inefficient i.e., they typically require a large number of black box model queries to construct local approximations [21]. This can be prohibitively slow especially in case of complex neural models.

In this paper, we identify that modeling uncertainty in black box explanations is the key to addressing all the aforementioned challenges. To this end, we propose a novel Bayesian framework for generating local explanations along with their associated uncertainty. We instantiate this framework to obtain Bayesian versions of LIME and KernelSHAP, namely BayesLIME and BayesSHAP, that not only output point-wise estimates of feature importance but also their associated uncertainty in the form of credible intervals (See Figure 1). We derive closed form expressions for the posteriors of the explanations thereby eliminating the need for any additional computational complexity. The credible intervals produced by our framework not only allow us to make concrete inferences about the quality of the resulting explanations but also produce explanations that satisfy user specified levels of uncertainty (e.g., an end user may request for explanations that satisfy a certain 95% confidence level). In addition, the resulting explanations are also highly consistent and stable. *To the best of our knowledge, this work makes the first attempt at addressing several critical challenges in popular explanation methods in one-shots, thereby generating consistent, stable, and reliable explanations with guarantees in a computationally efficient manner*.

We carry out theoretical analysis that leverages the measures of uncertainty (credible intervals) produced by our framework to estimate the values of critical hyperparameters. More specifically, we derive a closed form expression for the number of perturbations required to generate explanations that satisfy desired levels of confidence. We also propose a novel sampling technique called *focused sampling* that leverages uncertainty to determine how to sample perturbations for faster convergence, thereby enabling our framework to generate explanations in a computationally efficient manner.

We evaluate the efficacy of the proposed framework on a variety of datasets including COMPAS, German Credit, ImageNet, and MNIST. Our results demonstrate that the explanations output by our framework are not only highly reliable, but also very consistent and stable (53% more stable than LIME/SHAP on an average). Our experimental results also confirm that we can accurately estimate the number of perturbations needed to generate explanations with a desired level of uncertainty, and that our uncertainty sampling technique speeds up the process of generating explanations by up to a factor of 2 relative to random sampling of perturbations. Lastly, we carry out a user study with 31 human subjects to evaluate the quality of the explanations generated by our framework, demonstrating that our explanations accurately capture the importance of the most influential features.

2 Notation & Background

Here we introduce notation and discuss two relevant prior approaches, LIME and KernelSHAP.

Notation Let $f : \mathbb{R}^d \rightarrow [0, 1]$ denote a black box classifier that takes a data point x with d features, and returns the *probability* that x belongs to a certain class. Our goal is to explain individual predictions of f. Let $\phi \in \mathbb{R}^d$ denote the explanation in terms of feature importances for the prediction $f(x)$, i.e. coefficients ϕ are treated as the feature *contributions* to the black box prediction. Note that ϕ captures the coefficients of a linear model. Let $\mathcal{Z}$ be a set of N randomly sampled instances (perturbations) around x. The proximity between x and any $z \in \mathcal{Z}$ is given by $\pi_x(z) \in \mathbb{R}$. We denote the vector of these distances over the N perturbations in $\mathcal{Z}$ as $\Pi_x(\mathcal{Z}) \in \mathbb{R}^N$. Let $Y \in [0, 1]$ be the vector of the black box predictions $f(z)$ corresponding to each of the N instances in $\mathcal{Z}$.

LIME [2] and **KernelSHAP** [4] are popular *model-agnostic local explanation* approaches that explain predictions of a classifier f by learning a linear model ϕ locally around each prediction (i.e. $y \sim \phi^T z$). The objective function for both LIME and KernelSHAP constructs an explanation that approximates the behavior of the black box accurately in the vicinity (neighborhood) of x.

$$\arg\min_{\phi} \sum_{z \in \mathcal{Z}} [f(z) - \phi^T z]^2 \pi_x(z). \tag{1}$$

The above objective function has the following closed form solution:

$$\hat{\phi} = (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + \mathbb{I})^{-1}(\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))Y) \tag{2}$$

The main difference between LIME and KernelSHAP lies in how $\pi_x(z)$ is chosen. In LIME, it is chosen heuristically: $\pi_x(z)$ is computed as the cosine or l_2 distance. KernelSHAP leverages game theoretic principles to compute $\pi_x(z)$, guaranteeing that explanations satisfy certain properties.

3 Our Framework: Bayesian Local Explanations

In this section, we introduce our Bayesian framework which is designed to capture the uncertainty associated with local explanations of black box models. First, we discuss the generative process and inference procedure for the framework. Then, we highlight how our framework can be instantiated to obtain Bayesian versions of LIME and SHAP. Lastly, we present detailed theoretical analysis for estimating the values of critical hyperparameters, and discuss how to efficiently construct highly accurate explanations with uncertainty guarantees using our framework.

3.1 Constructing Bayesian Local Explanations

Our goal here is to explain the behavior of a given black box model f in the vicinity of an instance x while also capturing the uncertainty associated with the explanation. To this end, we propose a Bayesian framework for constructing local linear model based explanations and capturing their associated uncertainty. We model the black box prediction of each perturbation z as a linear combination of the corresponding feature values ($\phi^T z$) plus an error term (ϵ) as shown in Eqn (3). While the weights of the linear combination ϕ capture the feature importances and thereby constitute our explanation, ϵ captures the error that arises due to the mismatch between our explanation ϕ and the local decision surface of the black box model f. Our complete generative process is shown below:

$$y|z, \phi, \epsilon \sim \phi^T z + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{\pi_x(z)}) \qquad \phi|\sigma^2 \sim \mathcal{N}(0, \sigma^2 \mathbb{I}) \qquad \sigma^2 \sim \text{Inv-}\chi^2(n_0, \sigma_0^2). \tag{3}$$

The error term is modeled as a Gaussian whose variance relies on the proximity function $\pi_x(z)$ i.e., $\epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{\pi_x(z)})$. This proximity function ensures that perturbations closer to the data point x are modeled accurately, while allowing more room for error in case of perturbations that are farther away. $\pi_x(z)$ can be computed using cosine or l_2 distance or other game theoretic principles similar to that of LIME and KernelSHAP (see Section 2). The conjugate priors on ϕ and σ^2 are shown in Eqn (3).

Thus, our generative process corresponds to the Bayesian version of the weighted least squares formulation of LIME and KernelSHAP outlined in Eqn. (1), with additional terms to model uncertainty.

As in Eqns. (3), the process captures two sources of uncertainty in local explanations: 1) ***feature importance uncertainty***: the uncertainty associated with the feature importances ϕ, and (2) ***error uncertainty***: the uncertainty associated with the error term ϵ which captures how well our explanation ϕ models the local decision surface of the underlying black box.

Inference Our inference process involves estimating the values of two key parameters: ϕ and σ^2. By doing so, we can compute the local explanation as well as the uncertainties associated with feature importances and the error term. Posterior distributions on ϕ and σ^2 are normal and scaled Inv-χ^2, respectively, due to the corresponding conjugate priors [22]:

$$\sigma^2|\mathcal{Z}, Y \sim \text{Scaled-Inv-}\chi^2\left(n_0 + N, \frac{n_0\sigma_0^2 + Ns^2}{n_0 + N}\right) \qquad \phi|\sigma^2, \mathcal{Z}, Y \sim \text{Normal}(\hat{\phi}, V_\phi\sigma^2) \tag{4}$$

Further, $\hat{\phi}$, V_ϕ, and s^2 can be directly computed:

$$\hat{\phi} = V_\phi(\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))Y) \qquad V_\phi = \left(\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + \mathbb{I}\right)^{-1} \tag{5}$$

$$s^2 = \frac{1}{N}\left[(Y - \mathcal{Z}\hat{\phi})^T \text{diag}(\Pi_x(\mathcal{Z}))(Y - \mathcal{Z}\hat{\phi}) + \hat{\phi}^T\hat{\phi}\right] \tag{6}$$

Details of the complete inference procedure including derivations of Eqns. (4-6) are provided in the Appendix A. Note that our estimate of the posterior mean feature importances $\hat{\phi}$ (Eqn. (5)) is the same as that of the feature importances computed in case of LIME and KernelSHAP (Eqn. (2)).

Remark 3.1. *If we use the same proximity function $\pi_x(z)$ in our framework as in LIME or KernelSHAP, the posterior mean of the feature importance $\hat{\phi}$ output by our framework (Eq* (5)*) will be equivalent to the feature importances output by LIME or KernelSHAP, respectively.*

Feature Importance Uncertainty To obtain the local feature importances and their associated uncertainty, we first compute the posterior mean of the local feature importances $\hat{\phi}$ using the closed form expression in Eqn. (6). We then estimate the credible interval (measure of uncertainty) around the mean feature importances by repeatedly sampling from the posterior distribution of ϕ (Eq (4)).

Error Uncertainty The error term ϵ can serve as a proxy for explanation quality because it captures the mismatch between the constructed explanation and the local decision surface of the underlying black box. To compute the uncertainty associated with this error term (error uncertainty), we need to estimate the probability that the error term becomes 0 i.e., $P(\epsilon = 0)$. To this end, we first calculate the marginal posterior distribution of ϵ by leveraging Eqn (3) and integrating out σ^2. This results in a three parameter Student's t distribution (derivation in appendix A):

$$\epsilon|\mathcal{Z}, Y \sim t_{(\mathcal{V}=n_0+N)}(0, \frac{n_0\sigma_0^2 + Ns^2}{n_0 + N}). \tag{7}$$

We then evaluate the probability density function (PDF) of the above posterior at 0, i.e., $P(\epsilon = 0)$ by substituting the value of s^2 computed using Eqn. (6) into the Student's t distribution above (Eqn. (7)). The resulting expression gives us the probability density that the explanation output by our framework perfectly captures the local decision surface underlying the black box. This operation is performed in constant time, adding minimal overhead to non-Bayesian LIME and SHAP. We illustrate how these computed intervals capture the variance in the explanations in Figure 10.

Proposition 3.2. *As the number of perturbations around x goes to ∞ i.e., $N \to \infty$: (1) the estimate of ϕ converges to the true feature importance scores, and its uncertainty to 0. (2) uncertainty of the error term ϵ converges to the bias of the local linear model ϕ. [Details in Appendix B]*

BayesLIME and BayesSHAP Our framework can be instantiated to obtain the Bayesian version of LIME by setting the proximity function to $\pi_x(z) = \exp(-D(x, z)^2/\sigma^2)$ where D is a distance metric (e.g. cosine or l_2 distance), and n_0 and σ_0^2 to small values (10^{-6}) so that the prior is uninformative. We compute feature importance uncertainty and error uncertainty for LIME's feature importances.

Our framework can also be instantiated to obtain the Bayesian version of KernelSHAP by setting uninformative prior on σ^2 and $\pi_x(z) = \frac{d-1}{(d \text{ choose } |z|)|z|(d-|z|)}$ where $|z|$ denotes the number of the variables in the variable combination represented by the data point z i.e., the number of non-zero

valued features in the vector representation of z. Note that the original SHAP method views the problem of constructing a local linear model as estimating the Shapley values corresponding to each of the features [4]. These Shapley values represent the contribution of each of the features to the black box prediction i.e., $f(x) = \phi_0 + \sum \phi_i$. Therefore, the measures of uncertainty output by our method BayesSHAP capture the reliability of the estimated variable contributions.

To encourage BayesLIME and BayesSHAP explanations to be sparse, we can use dimensionality reduction or feature selection techniques as used by LIME and SHAP to obtain the top K features [2, 4, 23]. We can then construct our explanations using the data corresponding to these top K features.

3.2 Estimating the Number of Perturbations

One of the major drawbacks of approaches such as LIME and KernelSHAP is that they do not provide any guidance on how to choose the number of perturbations, a key factor in obtaining reliable explanations in an efficient manner. To address this, we leverage the uncertainty estimates output by our framework to compute *perturbations-to-go* (G), an estimate of how many *more* perturbations are required to obtain explanations that satisfy a desired level of certainty. This estimate thus *predicts* the computational cost of generating an explanation with a desired level of certainty and can help determine whether it is even worthwhile to do so. The user specifies the confidence level of the credible interval (denoted as α) and the *maximum* width of the credible interval (W), e.g. "width of 95% credible interval should be less than 0.1" corresponds to $\alpha = 0.95$ and $W = 0.1$. To estimate G for the local explanation of a data point x, we first generate S perturbations around x (where S is small and chosen by the user) and fit a local linear model using our method[1]. This provides initial estimates of various parameters shown in Eqns (4)-(6) which can then be used to compute G.

Theorem 3.3. *Given S seed perturbations, the number of additional perturbations required (G) to achieve a credible interval width W of feature importance for a data point x at user-specified confidence level α can be computed as:*

$$G(W, \alpha, x) = \frac{4s_S^2}{\bar{\pi}_S \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} - S \tag{8}$$

where $\bar{\pi}_S$ is the average proximity $\pi_x(z)$ for the S perturbations, s_S^2 is the empirical sum of squared errors (SSE) between the black box and local linear model predictions, weighted by $\pi_x(z)$, as in (6)*, and $\Phi^{-1}(\alpha)$ is the two-tailed inverse normal CDF at confidence level α.*

Proof (Sketch). To estimate G, we first relate W and α to $\text{Var}(\phi_i)$, the marginal variance of the feature importance[2] for any feature i, obtained by integrating out σ^2. Because Student's t can be approximated by a Normal distribution for large degrees of freedom (here, S should be large enough), we use the inverse normal CDF to calculate credible interval width at level α. We compute V_ϕ from (5) using $\mathcal{Z}$, treating its entries as Bernoulli distributed with probability 0.5. Due to the covariance structure of this sampling procedure, the resulting variance estimate after N samples is the sample SSE s_S^2 scaled by $\approx \frac{4}{\bar{\pi}_S N}$ (derivation in appendix B). If we assume SSE scales linearly with S, we can take this to be a reasonable estimate of s_N^2 at any N. We can then estimate G as

$$\left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2 = \text{Var}(\phi_i) = \frac{4s_S^2}{\bar{\pi}_S \times (G + S)} \implies G = \frac{4s_S^2}{\bar{\pi}_S \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} - S. \tag{9}$$

□

3.3 Focused Sampling of Perturbations

Perturbations-to-go (G) provides us with an estimate of how many samples are required to achieve reliable explanations. However, if G is large, querying the black-box model for its predictions on a large number of perturbations can be computationally expensive for larger models [24, 25]. To reduce this cost, we develop an alternative sampling procedure called *focused sampling* which leverages

[1] We assume a simplified feature space where features are present or absent according to Bernoulli(.5). As in Ribeiro et al. [2], these *interpretable* features are flexible and can encode what is important to the end user.

[2] Since the error depends primarily on the number of perturbations, $\text{Var}(\phi_i)$ is similar across features.

Data set	BayesLIME	BayesSHAP
ImageNet	94.8	89.9
MNIST	97.2	90.1
COMPAS	95.5	87.9
German Credit	96.9	89.6

Table 1: **Evaluating Credible Intervals.** We report the % of time the 95% credible intervals with 100 perturbations include their true values (estimated on $10,000$ perturbations). Closer to 95.0 is better. Both BayesLIME and BayesSHAP are well calibrated.

uncertainty estimates to query the black box in a more targeted fashion (instead of querying randomly), thereby reducing the computational cost associated with generating reliable explanations. Inspired by active learning [26], focused sampling strategically prioritizes perturbations whose predictions the explanation is most uncertain about, when querying the black box. This enables the focused sampling procedure to query the black box only for the predictions of the most informative perturbations and thereby learn an accurate explanation with far fewer queries to the black box.

To determine how uncertain our explanation ϕ is about the black box label for any given instance z, we first compute the posterior predictive distribution for z (derivation in Appendix A), given as $\hat{y}(z)|\mathcal{Z}, Y \sim t_{(\mathcal{V}=N)}(\hat{\phi}^T z, (z^T V_\phi z + 1)s^2)$. The variance of this three parameter student's t distribution is $((z^T V_\phi z + 1)s^2)(N/(N-2))$. We refer to this variance as the *predictive variance* and it captures how uncertain our explanation ϕ is about the black box label for an instance z.

The focus sampling procedure first fits the explanation with an initial S perturbations (where S is a small number). We then iterate the following procedure until the desired explanation certainty level is reached. We draw a batch of A candidate perturbations, compute their predictive variance with the Bayesian explanation, and induce a distribution over the perturbations by running softmax on the variances. We draw a batch of B perturbations from this distribution and query the black box model for their labels. Finally, we refit the Bayesian explanation on all the labeled perturbations collected so far. We provide pseudocode for the uncertainty sampling procedure in appendix Algorithm 1.

4 Experiments

We evaluate the proposed framework by first analyzing the quality of our uncertainty estimates i.e., feature importance uncertainty and error uncertainty. We also assess our estimates of required perturbations (G), and evaluate the computational efficiency of focused sampling. Last, we describe a user study with 31 subjects to assess the informativeness of the explanations output by our framework.

Setup We experiment with a variety of real world datasets spanning multiple applications (e.g., criminal justice, credit scoring) as well as modalities (e.g., structured data, images). Our first structured dataset is **COMPAS** [27], containing criminal history, jail and prison time, and demographic attributes of 6172 defendants, with class labels that represent whether each defendant was rearrested within 2 years of release. The second structured dataset is the **German Credit** dataset from the UCI repository [28] containing financial and demographic information (including account information, credit history, employment, gender) for 1000 loan applications, each labeled as a "good" or "bad" customer. We create 80/20 train/test splits for these two datasets, and train a random forest classifier (sklearn implementation with 100 estimators) as *black box* models for each (test accuracy of 62.5% and 64.0%, respectively). We also include popular image datasets–MNIST and Imagenet. For the **MNIST** [29] handwritten digits dataset, we train a 2-layer CNN to predict the digits (test accuracy of 99.2%) and use the prediction of digit "4" as the target class. For **Imagenet** [30], we use the off-the-shelf VGG16 model [31] as the black box, and select a sample of 100 "French bulldog" images as our test set and explanation target (the model predicts French bulldog on 88% of these images). For generating explanations, we use standard implementations of the baselines LIME and KernelSHAP with default settings [2, 4]. For images, we construct super pixels as described in [2] and use them as features (number of super pixels is fixed to 20 per image). For our framework, we set perturbation size $N = 50$, batch size $B = 10$, the desired level of certainty is expressed as the width of the 95% credible interval, and use all the features.

Quality of Uncertainty Estimates A critical component of our explanations is the feature importance uncertainty. To evaluate the correctness of these estimates, we compute how often *true* feature importances lie within the 95% credible intervals estimated by BayesLIME and BayesSHAP. We

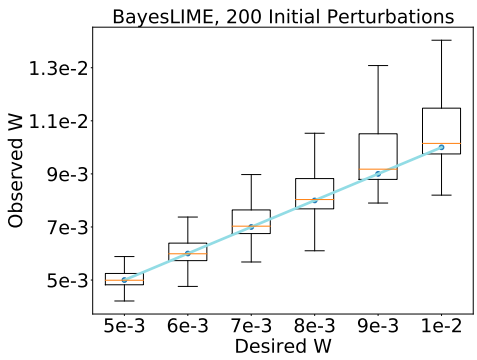


Figure 2: **Perturbations-to-go** (G). We generate explanation with G perturbations, where G is computed using the *desired* credible interval width (x-axis), and compare desired levels to the *observed* credible interval width (y-axis) (blue line indicates ideal calibration). Results are averaged over 100 MNIST images, and G, estimated from $S = 200$ seed samples, varies between 200 and $20,000$ across images. We see that G provides a good approximation of the additional perturbations needed.

evaluate the quality of our credible interval estimates by running our methods with 100 perturbations to estimate feature importances and taking the corresponding 95% credible intervals for each test instance. We compute what fraction of the true feature importances fall within our 95% credible intervals. Note, because there are no methods to provide uncertainty estimates for LIME and SHAP, we do not provide further baselines. Since we do not have access to the true feature importances of the complex black box models, following Prop 3.2, we use feature importances computed using a large value of N ($N = 10,000$), and treat the resulting estimates as ground truth.

Results for BayesLIME in Table 1 indicate that the true feature importances are close to ideal and indicate the estimates are well calibrated. While the estimates by BayesSHAP are somewhat less calibrated (true feature importances fall within our estimated 95% credible intervals about 89.6 to 90.1% of the time), they still are quite close to ideal. All in all, these results confirm that the credible intervals learned by our methods are well calibrated and therefore highly reliable in capturing the uncertainty of the feature importances. Lastly, though we set our priors to be uninformative in general, we also investigate how sensitive our uncertainty estimates are to hyperparameter choices in Figure 5 in the Appendix. We find that the explanation uncertainty becomes uncalibrated with strong priors. However, our explanations seem to be robust to hyperparameter choices in general.

Correctness of Estimated Number of Perturbations We assess whether our estimate of *perturbations-to-go* (G; Section 3.2) is an accurate estimate of the *additional* number of perturbations needed to reach a desired level of feature importance certainty. We carry out this experiment on MNIST data (additional datasets explored in Appendix C) and use $S = 200$ as the initial number of perturbations to obtain a preliminary explanation and its associated uncertainty estimates. We then leverage these estimates to compute G for 6 different certainty levels. First, we observe significant differences in G estimates across instances (details in appendix C) i.e. number of perturbations needed to obtain a particular level of certainty varied significantly across instances–ranging from 200-$5,000$ for the lowest level of certainty to 200-$20,000$ for higher levels of certainty. Next, for each image and certainty level, we run our method for the estimated number of perturbations (G) to determine if the observed estimates of uncertainty (observed credible interval width W) match the desired levels of uncertainty (desired credible interval width W). Results in Figure 2 show that the observed and desired levels of certainty are well calibrated, demonstrating that G estimates are reliable approximations of the additional number of perturbations needed.

Efficiency of Focused Sampling *Focused sampling* uses the *predictive variance* to strategically choose perturbations that will reduce uncertainty in order to be labeled by the black box (section 3.3). Here, we will evaluate the efficiency of the focused sampling procedure. First, we assess whether focused sampling produces reliable explanations (as measured by error uncertainty ($P(\epsilon = 0)$)) more efficiently than random sampling. To this end, we experiment with BayesLIME on Imagenet data to carry out this analysis. This setting replicates scenarios where LIME is applied to a computationally expensive black box model, making it highly desirable to limit the number of perturbations to reduce total running time. We run each sampling strategy for 2,000 perturbations and plot the number of model queries versus error uncertainty. The results in Figure 3 show that focused sampling results in faster convergence to reliable and high quality explanations; focused sampling stabilizes within a couple hundred model queries while random sampling takes over 1,000. Note, as the inefficiency of querying the black box model increases, the advantages of focused sampling decreasing total running

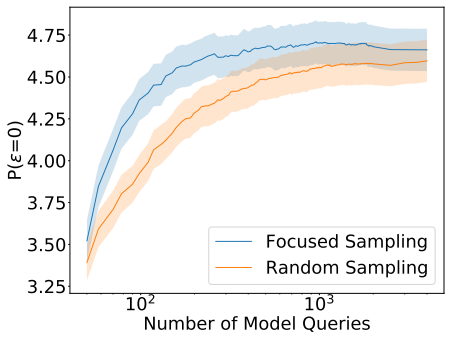


Figure 3: **Efficiency of focused sampling** for 100 Imagenet images, with random sampling as a baseline. We provide mean and standard error. We assess the efficiency of focused sampling by comparing *error uncertainty* (y-axis) over model queries, and show much quicker convergence than random sampling.

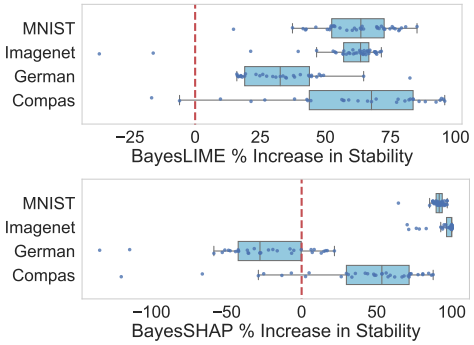


Figure 4: **Assessing the % increase in stability** of BayesLIME and BayesSHAP over LIME and SHAP respectively. Our Bayesian methods are significant more stable ($\rho <$ 1e-2 according to Wilcoxon signed-rank test) except for BayesSHAP on German Credit, where there is not a significant difference between the methods ($\rho > 0.05$).

time of the explanations will only become more pronounced. These results clearly demonstrate that focused sampling can significantly speed up the process of generating high quality local explanations.

We also benchmark the efficiency of BayesLIME and BayesSHAP against Guo et al. [32], a related Bayesian explanation method that uses a Bayesian non parametric mixture regression and MCMC for parameter inference. Fixing their mixture regression to a single component results in a similar model to ours and thus is a useful point of comparison. To explain a single instance on ImageNet using VGG16, their approach takes 139.2 seconds, while BayesLIME and BayesSHAP take 20.3 seconds and 21.1 seconds respectively, under the same conditions, demonstrating that the closed form solution is very efficient. Additionally, in Appendix C, we also check if focused sampling causes any bias (due to sampling based on uncertainty estimates) that results in convergence to a different/wrong explanation, however our results clearly indicate that this is not the case.

Stability of BayesLIME & BayesSHAP Recall that LIME & SHAP are not stable: small changes to instances can produce substantially different explanations. We consider whether BayesLIME & BayesSHAP produce more stable explanations than their LIME & SHAP counterparts. To perform this analysis, we use the local Lipschitz metric for explanation stability [18]:

$$\hat{L}(x_i) = \underset{x_j \in N_\epsilon(x_i)}{\text{argmax}} \frac{||\phi_i - \phi_j||_2}{||x_i - x_j||_2} \tag{10}$$

where x_i refers to an instance, $N_\epsilon(x_i)$ is the ϵ-ball centered at x_i, and ϕ_i and ϕ_j are the explanation parameters for x_i and x_j. Lower values of the above metric indicate more stable explanations. We follow the setup outline by Alvarez-Melis and Jaakkola [18] and compute the local Lipschitz values, comparing both LIME & BayesLIME and SHAP & BayesSHAP across Compas, German Credit, MNIST, and Imagenet. We perform the comparison using the default number of perturbations in both LIME & SHAP, and use this same number in the respective Bayesian variants. We use focused sampling for BayesLIME and BayesSHAP, and report the % increase in stability of these approaches over LIME and SHAP, respectively, for 40 points in the test data. The results given in Figure 4 show a clear improvement (on average 53%) in stability for both BayesLIME and BayesSHAP in all cases except German Credit for BayesSHAP. Further, we run a Wilcoxon signed-rank test and find our results are statistically significant in all cases ($\rho <$ 1e-2) except for BayesSHAP for German Credit, where there is not a significant difference between the methods ($\rho > 0.05$). These results demonstrate BayesLIME and BayesSHAP are more stable than previous methods.

User Study We perform a user study with 31 subjects to compare BayesLIME and LIME explanations on MNIST. We evaluate the following: are explanations with low levels of uncertainty (i.e., most confident explanations) more meaningful to humans? To answer this question, we mask the most important features selected by BayesLIME and LIME, and ask users to guess the digit of the masked images. The better the explanation, the more difficult it should be for the users to get it right. We

randomly select 15 correctly predicted test images, generate explanations by sweeping over a range of perturbation amounts $[10^{.5}, ..., 10^{3.5}]$ incremented by 0.5. We choose the *top* explanation for each image based on either fidelity (for LIME) or $P(\epsilon = 0)$ (for BayesLIME). We sent the user study out to students and researchers with background in computer science. A screen shot of the task is show in Figure 8 in the Appendix. We find that the explanations output by our methods focus on more informative parts of the image, since hiding them makes it difficult for humans to guess the digit. Users had an error rate of 25.7% for LIME, while it was 30.7% for BayesLIME, both with standard error 0.003 ($\rho = 0.028$ through a one-tailed two sample t-test). This result indicates that our method BayesLIME and the associated measure of explanation uncertainty result in more high quality and reliable explanations compared to LIME and its associated fidelity metric.

5 Related Work

Interpretability Methods A variety of interpretability methods have been proposed. Some methods that are inherently interpretable include additive models [33, 34], decision lists and sets [35, 36], and instance-based explanations [37]. However, black-box models are often more flexible, accurate, and easier to use; thus, there has been a lot of interest in constructing post hoc explanations[38]. These include LIME [2] and SHAP [4], which are among the most popular due to their broad applicability and code availability, but saliency maps [5–8], permutation feature importance [39], and partial dependency plots [40] also follow this paradigm. Other approaches to post hoc explanations focus on rule-based models [1, 3], counterfactuals [41, 42], and influence functions [9].

Vulnerabilities of Post hoc Explanations Recent work has shed light on the downsides of post hoc explanation techniques. These methods are often highly sensitive to small changes in inputs [14], are susceptible to manipulation [15, 16, 43, 44], and are not faithful to the underlying black boxes [45]. Perturbation-based explanation methods such as LIME and SHAP are subject to additional criticisms: results vary between runs of the algorithms [18–20, 46, 21], and hyperparameters used to select the perturbations can greatly influence the resulting explanation [20]. Prior work has attempted to tackle the problem of instability in perturbation-based explanations by averaging over several explanations [47, 19], however, this is computationally expensive. Other works related to creating more trustworthy explanations include development of sanity checks for explainers [48, 17, 49]. These techniques represent an important step towards improved usability, given experimental evidence that humans are often too eager to accept inaccurate machine explanations [50–53]. Recent works theoretically analyze the sources of non-robustness in black box explanations [54–56].

Bayesian Methods in Explainable ML Few recent works have adopted Bayesian formulations to explain black box models [32, 57]. Guo et al. [32] introduce a Bayesian non-parametric approach to fit a *global* surrogate model. Their formulation seeks to fit a mixture of generalizable explanations across instances. Zhao et al. [57] draw on Bayesian frameworks from a preprint of this work to study whether incorporating informative priors improves the stability of the resulting explanations. However, neither of these works focus on modeling the uncertainty of local explanations. Further, these approaches also do not tackle the critical problems of estimating key hyperparameters or improving efficiency of computing explanations.

6 Conclusion

We developed a Bayesian framework for generating local explanations along with their associated uncertainty. We instantiated this framework to obtain Bayesian versions of LIME and SHAP that output pointwise estimates of feature importances as well as their associated credible intervals. These intervals enabled us to infer the quality of the explanations and output explanations that satisfied user specified levels of uncertainty. We carried out theoretical analysis that leverages these uncertainty measures (credible intervals) to estimate the values of critical hyperparameters (e.g., the number of perturbations). We also proposed a novel sampling technique called focused sampling that leverages uncertainty estimates to determine how to sample perturbations for faster convergence. One potential negative outcome of our work is that if the credible intervals output by our framework are incorrect, it could lead to overreliance of faulty explanations. It would be interesting to extend our framework to produce global explanations with uncertainty guarantees and explore how uncertainty quantification can help calibrate user trust in model explanations.

References

[1] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[2] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why Should I Trust You? explaining the predictions of any classifier. In *Knowledge Discovery and Data mining (KDD)*, 2016.

[3] 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. ACM, 2019.

[4] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[5] 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.

[6] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

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

[8] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smooth-grad: removing noise by adding noise. *Workshop on Visualization for Deep Learning, ICML*, 2017.

[9] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org, 2017.

[10] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpretability via model extraction. *FAT/ML Workshop 2017*, 2017.

[11] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Neural Information Processing Systems*, 2018.

[12] Radwa Elshawi, Mouaz H Al-Mallah, and Sherif Sakr. On the interpretability of machine learning-based model for predicting hypertension. *BMC medical informatics and decision making*, 19(1):146, 2019.

[13] Leanne S Whitmore, Anthe George, and Corey M Hudson. Mapping chemical performance on molecular structures using locally interpretable explanations. *arXiv preprint arXiv:1611.07443*, 2016.

[14] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.

[15] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. *Conference on Artificial Intelligence, Ethics, and Society (AIES)*, 2020.

[16] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. *arXiv preprint arXiv:1906.07983*, 2019.

[17] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[18] David Alvarez-Melis and Tommi S. Jaakkola. On the robustness of interpretability methods. *ICML Workshop on Human Interpretability in Machine Learning*, 2018.

[19] Eunjin Lee, David Braines, Mitchell Stiffler, Adam Hudler, and Daniel Harborne. Developing the sensitivity of lime for better machine learning explanation. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100610. International Society for Optics and Photonics, 2019.

[20] Hui Fen Tan, Kuangyan Song, Madeilene Udell, Yiming Sun, and Yujia Zhang. "why should you trust my explanation?" understanding uncertainty in lime explanations. In *ICML Workshop on AI for Social Good*, 2019.

[21] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2019.

[22] Andrew Moore. Locally weighted bayesian regression, January 1995.

[23] Kacper Sokol, Alexander Hepburn, Raúl Santos-Rodríguez, and Peter A. Flach. blimey: Surrogate prediction explanations beyond lime. *NeurIPS HCML Workshop*, 2019.

[24] Emily L. Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.

[25] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014*, 05 2014.

[26] Burr Settles. Active learning literature survey. 2010.

[27] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *ProPublica*, 2016.

[28] Dheeru Dua and Casey Graff. Uci machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[29] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[32] Wenbo Guo, Sui Huang, Yunzhe Tao, Xinyu Xing, and Lin Lin. Explaining deep learning models – a bayesian non-parametric approach. In *Neural Information Processing Systems (NeurIPS)*. 2018.

[33] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *KDD*, 2013.

[34] Berk Ustun, Stefano Traca, and Cynthia Rudin. Supersparse linear integer models for interpretable classification. *arXiv preprint arXiv:1306.6677*, 2013.

[35] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Knowledge Discovery and Data mining (KDD)*, 2016.

[36] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *arXiv preprint arXiv:1704.01701*, 2017.

[37] Been Kim, Cynthia Rudin, and Julie Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *arXiv preprint arXiv:1503.01161*, 2015.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Model-Agnostic Interpretability of Machine Learning. In *ICML Workshop on Human Interpretability in Machine Learning (WHI)*, June 2016.

[39] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[40] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[41] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 10–19, 2019. ISBN 978-1-4503-6125-5.

[42] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[43] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. In *Advances in Neural Information Processing Systems 32*, pages 2921–2932. 2019.

[44] Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. Gradient-based Analysis of NLP Models is Manipulable. In *Findings of the Association for Computational Linguistics: EMNLP (EMNLP Findings)*, page 247–258, 2020.

[45] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206, 2019.

[46] Muhammad Rehman Zafar and Naimul Mefraz Khan. Dlime: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019.

[47] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. In *Advances in Neural Information Processing Systems*, pages 10965–10976, 2019.

[48] Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. Can i trust the explainer? verifying post-hoc explanatory methods. *arXiv preprint arXiv:1910.02065*, 2019.

[49] Mengjiao Yang and Been Kim. Bim: Towards quantitative evaluation of interpretability methods with ground truth. *arXiv:1907.09701*, 2019.

[50] Harmanpreet Kaur, Harsha Nori, Samuel Jenkins, Rich Caruana, Hanna Wallach, and Jennifer Wortman Vaughan. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *CHI*, April 2020.

[51] Fred Hohman, Andrew Head, Rich Caruana, Robert DeLine, and Steven M Drucker. Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.

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

[53] Himabindu Lakkaraju and Osbert Bastani. " how do i fool you?" manipulating user trust via misleading black box explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 79–85, 2020.

[54] Damien Garreau and Ulrike von Luxburg. Looking deeper into lime. *arXiv preprint arXiv:2008.11092*, 2020.

[55] Prasad Chalasani, Jiefeng Chen, Amrita Roy Chowdhury, Xi Wu, and Somesh Jha. Concise explanations of neural networks using adversarial training. In *International Conference on Machine Learning*, pages 1383–1391. PMLR, 2020.

[56] Alexander Levine, Sahil Singla, and Soheil Feizi. Certifiably robust interpretation in deep learning. *arXiv preprint arXiv:1905.12105*, 2019.

[57] Xingyu Zhao, Xiaowei Huang, Valentin Robu, and David Flynn. Baylime: Bayesian local interpretable model-agnostic explanations. *arXiv preprint arXiv:2012.03058*, 2020.

[58] Ludwig Fahrmeir, Thomas Kneib, and Stefan Lang. *Regression*. Statistik und ihre Anwendungen. Springer, 2007. ISBN 978-3-540-33932-8.

[59] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

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] We introduce our Bayesian explanation framework and describe that it leads to improved explanations. By describing in the main text each way it does this (e.g., focused sampling, PTG) and validating these experimentally, we validate the claims made in the introduction.
   (b) Did you describe the limitations of your work? [Yes] We describe the limitations of our work in a number of places. For one, we describe where our Bayesian explanations are not successful experimentally (e.g., German credit in BayesSHAP stability in figure 4). We also indicate the need to study how humans perceive the uncertainty estimates in the Conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss potential impacts in the Conclusion.
   (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? [Yes] The assumptions for our main theorem are clearly stated in the theorem body.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide proofs in the appendix.
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] Provided in supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide details in the experiment section, Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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] We provide discussion in section D in the Appendix.
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 use public datasets and cite the creators.
   (b) Did you mention the license of the assets? [Yes] We mention licenses in Section E
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include code in the supplement.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The datasets used are commonly used public datasets. For our user study, we provided a page where we asked participants whether they were willing to participate.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The datasets used are commonly used public datasets. For our user study, we did not ask for personal information.

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? [Yes] Provided in Appendix.
   (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]