arXiv:2008.05030v2 [cs.LG] 9 Oct 2020

How Much Should I Trust You? Modeling Uncertainty of Black Box Explanations

Dylan Slack, [1] **Sophie Hilgard,** [2] **Sameer Singh,** [1] **Himabindu Lakkaraju** [2]
[1] UC Irvine
[2] Harvard University

Abstract

As local explanations of black box models 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, local explanations generated by existing techniques are often prone to high variance. Further, these techniques are computationally inefficient, require significant hyper-parameter tuning, and provide little insight into the quality of the resulting explanations. We identify lack of uncertainty modeling as a main cause of these challenges and develop a novel set of tools for analyzing explanation uncertainty in a Bayesian framework. In particular, we estimate credible intervals (CIs) that capture the uncertainty associated with each feature importance in local explanations. These credible intervals are tight when we have high confidence in the feature importances of a local explanation. The CIs are also informative both for estimating how many perturbations we need to sample — sampling can proceed until the CIs are sufficiently narrow — and where to sample — sampling in regions with high predictive uncertainty leads to faster convergence. We instantiate this framework to generate Bayesian versions of LIME and KernelSHAP. Experimental evaluation with multiple real world datasets and user studies demonstrate the efficacy of our framework and the resulting explanations.

1 Introduction

As machine learning (ML) models are increasingly deployed in domains such as healthcare and criminal justice, it is important to ensure that decision makers understand these models so that they can diagnose errors and detect model biases correctly. 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 recent surge in post hoc techniques for explaining black box models (Ribeiro, Singh, and Guestrin 2018, 2016; Lakkaraju et al. 2019; Lundberg and Lee 2017a; Simonyan, Vedaldi, and Zisserman 2014; Sundararajan, Taly, and Yan 2017; Selvaraju et al. 2017; Smilkov et al. 2017; Koh and Liang 2017; Bastani, Kim, and Bastani 2017). Several of these techniques explain complex black box models by constructing interpretable local approximations such as linear functions (e.g., LIME (Ribeiro, Singh, and Guestrin 2016), SHAP (Lundberg and Lee 2017a)) or rules (e.g., MAPLE (Plumb, Molitor, and Talwalkar 2018), Anchors (Ribeiro, Singh, and Guestrin 2018)), which are much more readily understood by human users. The intuition behind constructing such local approximations is as follows: while complex black box models typically exhibit highly non-linear decision boundaries globally (and are therefore hard to explain *overall*), the behavior of these models is much less complex locally (e.g., linear decision boundaries), and are therefore more amenable to local explanation.

Existing local explanation methods, however, suffer from serious drawbacks. Explanations generated by methods such as LIME and SHAP are not stable, e.g., explanations can vary significantly even for instances that are nearly identical and have same predictions from the model (Alvarez-Melis and Jaakkola 2018; Ghorbani, Abid, and Zou 2019; Gruber and Molnar 2020; Tan et al. 2019; Kumar et al. Forthcoming 2020). Further, Chen et al. (2019) empirically demonstrate that multiple runs of these methods on the same instance with the same parameters can result in vastly different explanations. Commonly used metrics for assessing the quality of post hoc explanations (e.g., explanation fidelity) rely heavily on the 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 (Tan et al. 2019). In addition, there is little to no guidance on how to pick certain hyperparameters that are critical to the quality of the resulting local explanations, such as the number of perturbations. Lastly, these methods are also computationally inefficient, typically requiring a large number of model queries to construct local approximations, which can be prohibitively slow for complex neural models.

In this paper, we identify uncertainty of local explanations as key to addressing many of these issues, and propose a Bayesian framework for local explanations techniques such as LIME and KernelSHAP. In this framework, we estimate not only the point-wise estimates of feature importances but also capture the uncertainty associated with these feature importances as credible intervals (CIs) (see Figure 1 for an example). We show that uncertainty of the computed explanation is a strong indicator of its instability and can be used to assess the quality of local explanations, i.e., explanations with wider CIs may not be sufficiently trustworthy. Further, we show that key hyperparameters of explanation techniques

Correspondence to dslack@uci.edu

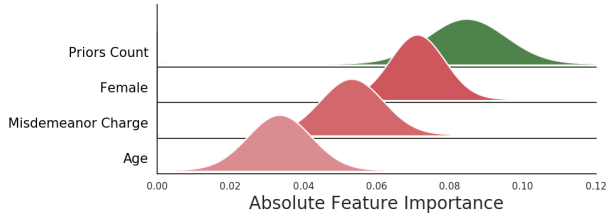


Figure 1: **Example explanation from BayesLIME** on rearrest prediction for an instance from the COMPAS dataset, showing the estimated feature importance and the uncertainty of the estimate (green indicates positive contribution, while red indicates negative). Our explanation suggests that `priors` and `sex` are likely most important features for predicting rearrest. However, overlapping uncertainty indicates one of them is *not* clearly more important than the other.

can be selected in a principled way when modeling uncertainty. We develop methods which leverage CIs to estimate how many perturbations must be sampled to achieve a given level of uncertainty (until the CIs are sufficiently narrow) and where to sample for the greatest reduction in uncertainty, thereby enabling our approaches to be computationally efficient in generating accurate local explanations. We compute closed form expressions for the posteriors of the explanations, requiring only minimal additional computation beyond the original LIME and KernelSHAP methods.

We evaluate the utility of modeling explanation uncertainty on a variety of datasets including COMPAS, German Credit, ImageNet, and MNIST. Our results demonstrate the uncertainty estimates produced by our framework are much more reliable proxies of how well the explantion approximates the black box compared to traditional metrics like fidelity. Our experimental results also confirm we can accurately estimate the number of perturbations needed to generate explanations of a given level of certainty. Furthermore, uncertainty-based sampling speeds up our method by up to a factor of 2 relative to random sampling of perturbations. Lastly, we carry out a user study with 31 humans to evaluate whether explanations generated by our framework focus on important features of the input.

2 Notation & Background

Here we establish the necessary notation, discuss the details of two relevant prior approaches, LIME and SHAP, study the vulnerabilities of these approaches, and illustrate the need for modeling the uncertainty of black box explanations.

Notation Following (Ribeiro, Singh, and Guestrin 2016), let $f : \mathbb{R}^d \to [0,1]$ denote a black box classifier that takes as input a data point x with d features, and returns the probability that x belongs to a certain class $c \in \mathcal{C}$. The goal is to explain the black box classifier f. Let $\phi \in \mathbb{R}^d$ denote an explanation that we intend to learn to explain f for instance x, where ϕ is the coefficients of a linear model. We define $\mathcal{Z}$ as a set of N randomly sampled instances (perturbations) around x. The proximity between x and any $z \in \mathcal{Z}$ is given by $\pi_x(z) \in \mathbb{R}$, with a vector 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 (Ribeiro, Singh, and Guestrin 2016) and **SHAP** (Lundberg and Lee 2017a) are popular *model-agnostic local explanation* approaches that explain individual predictions of any classifier f by learning a linear model ϕ locally around each prediction (i.e. $y \sim \phi^T z$). The coefficients ϕ are treated as the *contribution* of each feature to the corresponding black box prediction. The objective function for both LIME and SHAP is to construct an explanation that approximates the behavior of the black box accurately in the vicinity (neighborhood) of x.

$$\begin{aligned} &\arg\min_{\phi} L(f,\phi,\pi_x) \\ &L(f,\phi,\pi_x) \triangleq \sum_{z\in\mathcal{Z}} [f(z) - \phi^T z)]^2 \pi_x(z). \end{aligned} \tag{1}$$

The main difference between LIME and SHAP lies in how $\pi_x(z)$ is chosen. In LIME, it is defined heuristically: $\pi_x(z)$ is defined using cosine or l_2 distance. KernelSHAP leverages game theoretic principles to assign values to these functions, thereby guaranteeing that the explanations satisfy certain desired properties such local accuracy, missingness, and consistency (Lundberg and Lee 2017a).

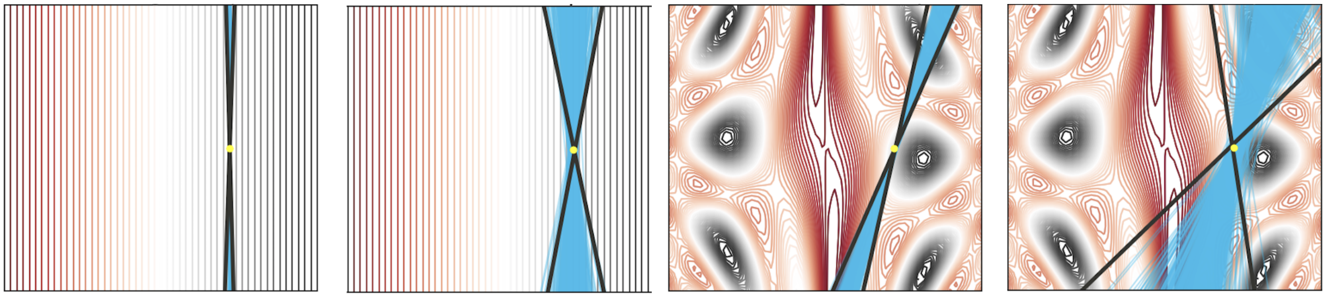
Limitations One of the biggest drawbacks of LIME and KernelSHAP is that the resulting explanations are prone to high variance. We illustrate this phenomenon using a toy 2-dimensional example. We run LIME to explain an instance as we vary the underlying decision surface and number of perturbations, in Figure 2. With a large number of samples and a linear surface in Fig 2a, we see that LIME explanations produced on multiple runs are nearly identical (blue lines), however, reducing the number of samples to 25 results in very different explanations (Fig 2b). For a classifier that violates the linearity assumption, as is common in practice, there is significant variance even with a large number of samples (Fig 2c) and an even wider discrepancy for fewer samples (Fig 2d). For real-world generation of explanations, as we do not know the underlying decision surface and it is not practical to run LIME thousands of times to estimate variance, it is difficult to estimate whether any generated explanation (a single blue line) is representative of the model.

3 Bayesian Local Explanations

In this section, we define our Bayesian framework which captures 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 discuss how to efficiently construct highly accurate explanations with uncertainty guarantees using our framework.

3.1 Constructing Bayesian Local Explanations

To model the variance of a given local explanation (of a black box model f in the vicinity of a data point x), we



(a) linear, many samples

(b) linear, fewer samples

(c) nonlinear, many samples

(d) nonlinear, fewer samples

Figure 2: **Rerunning LIME local explanations** 1000 **times and BayesLIME *once*** for linear and non-linear toy surfaces using few (25) and many (250) perturbations. The linear surface is given as $p(y) \propto x_1$ and the non linear surface is defined as $p(y) \propto \sin(x_1/2) * 10 + \cos(10 + (x_1 * x_2)/2) * \cos(x_1)$. We plot each run of LIME in blue and the BayesLIME 95% credible region in black. We see that there is high variance in LIME local explanations and that BayesLIME captures this variance well.

need to first model the uncertainty associated with the explanation. To this end, we define a Bayesian framework for fitting local linear explanations. Our framework is designed to capture two sources of explanation uncertainty: 1) uncertainty associated with the estimates of the feature importance scores ϕ and 2) uncertainty associated with the estimate of how well ϕ captures the local decision surface of the underlying black box model f. We model local explanations using the following generative process:

$$y|z, \phi, \sigma^2 \quad \sim \quad \phi^T z + \underbrace{\mathcal{N}(0, \frac{\sigma^2}{\pi_x(z)})}_{\epsilon}, \qquad \forall z \in \mathcal{Z} \tag{2}$$

$$\phi|\sigma^2 \sim \mathcal{N}(\phi_0, \sigma^2 \Sigma_0) \qquad\qquad \sigma^2 \sim \text{Inv-}\chi^2(n_0, \sigma_0^2). \tag{3}$$

Similar to LIME and SHAP, Eqn (2) models the linear relationship between the data points $z \in \mathcal{Z}$ and their corresponding black box predictions $y = f(z) \in Y$. However, we also introduce an error term $\epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{\pi_x(z)})$ that models the noise in this linear relationship: 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. The proximity function $\pi_x(z)$ in this error term ensures that perturbations that are in close proximity to the data point x are modeled accurately while allowing more room for error for perturbations that are farther away. The conjugate priors on ϕ and σ^2 are shown in Eqn (3). In practice, we set the hyperparameters $\phi_0 = 0$ and $\Sigma_0 = \text{Diag}(1, .., 1)$ in order to induce sparsity on ϕ. Additionally, we set n_0 and σ_0^2 to small values (10^{-6}) so that the prior is nearly uninformative.

Inference The posterior distributions on ϕ and σ^2 are normal and scaled Inv-χ^2 distributions respectively due to the corresponding conjugate priors (Moore 1995):

$$\begin{aligned} \sigma^2|\mathcal{Z}, Y &\sim \text{Scaled-Inv-}\chi^2(N, s^2) \\ \phi|\sigma^2, \mathcal{Z}, Y &\sim \text{Normal}(\hat{\phi}, V_\phi \sigma^2) \end{aligned} \tag{4}$$

We compute ϕ, V_ϕ, and s^2 in closed form:

$$\begin{aligned} \hat{\phi} &= V_\phi \mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z})) Y \\ V_\phi &= (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z})) \mathcal{Z} + I)^{-1} \end{aligned} \tag{5}$$

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

Details of the inference procedure including derivations of Eqns. (4-6) are included in appendix B.

Estimating Sources of Uncertainty Recall our generative process captures two sources of explanation uncertainty: uncertainty associated with the feature importance scores ϕ, and the uncertainty over the error term ϵ. We assess the former by repeatedly drawing from the posterior distribution of ϕ (Eq (4)). Then, we use these instances to estimate the 95% *credible intervals* of each feature in ϕ. In practice, this term is computed using the 95% density about $\hat{\phi}$ on 10,000 instances. We illustrate how these computed intervals capture the variance in the explanations in Figure 2.

The error term captures how accurate the explanation is to the underlying model. Therefore, we use it as a proxy for explanation quality. We calculate the marginal posterior distribution of ϵ by integrating out σ^2 which results in a three parameter Student's t distribution (derivation in appendix B):

$$\epsilon|\mathcal{Z}, Y \sim t_{(\mathcal{V}=N)}(0, s^2). \tag{7}$$

To compute our proxy for explanation quality, we evaluate the probability density function (PDF) of this posterior at 0, i.e., $P(\epsilon = 0)$, dropping dependence on $\mathcal{Z}$ and Y for conciseness, to estimate our confidence that there is no error in the explanation. This is computed in closed form using Student's t and s^2, which is directly computed from the data. This expression gives us the probability density that our explanation has no error. Thus, it describes whether ϕ perfectly captures local decision surface of the underlying black box. If the classifier is locally nonlinear, we expect this value to be low because is it unlikely the explanation describes the black box well.

Proposition 3.1. *As the number of perturbations sampled around x goes to ∞ i.e., $N \to \infty$:*
(1) the estimate of ϕ converges to the true importance scores, and its uncertainty converges to 0. *(2) uncertainty of the error term ϵ converges to the bias of the local linear model ϕ. [Details in Appendix C]*

BayesLIME and BayesSHAP We use this framework to generate a 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 function such as cosine or l_2 distance. We can then obtain the previously described uncertainty measures for LIME feature importances. This enables the derivation of tools to efficiently manage and mitigate variance in explanations using this popular method. Similarly, we instantiate Bayesian version of KernelSHAP by setting $\pi_x(z) = \frac{d-1}{(d \text{ choose } |z|)|z|(d-|z|)}$ where $|z|$ denotes the number of 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 (Lundberg and Lee 2017a). These Shapley values in turn represent the contribution of each of the features to the black box prediction i.e., $f(x) = \phi_0 + \sum \phi_i$. Therefore, the resulting estimates of uncertainty output by BayesSHAP represent how poorly defined the variable contributions are given the current sample of perturbations.

Generating Sparse Explanations In order to provide sparsity for BayesLIME and BayesSHAP, we can employ the same dimensionality reduction techniques used in the original methods (Lundberg and Lee 2017a; Ribeiro, Singh, and Guestrin 2016; Sokol et al. 2019). To generate sparse versions of either BayesLIME or BayesSHAP, users can run a preliminary locally-weighted lasso regression on the perturbations before computing BayesLIME or BayesSHAP using only the nonzero features.

3.2 Required Number of Perturbations

We use the uncertainty estimates from this framework to address one of the major failings of previous approaches (LIME and KernelSHAP): there is little guidance on how to pick the appropriate number of perturbations, a key factor for both the running time and the quality of the explanations. To this end, we develop *perturbations-to-go* (PTG) a measure 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 CI (denoted as α), and the *maximum* width of the CI (represented as W), e.g. "width of 95% CI interval should be less than 0.1" corresponds to $\alpha = 0.95$ and $W = 0.1$. To estimate PTG for the local explanation of a data point x, we first generate N perturbations around x (where N is small and chosen by the user) and fit a local linear model using our method.[1] This provides an initial estimates of various parameters shown in Eqns (4)-(6) which can then be used to compute PTG.

Theorem 3.2. *Given N seed perturbations, the number of additional perturbations required (PTG) 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:*

$$PTG(W, \alpha, x) = \frac{4s_N^2}{\bar{\pi}_N \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} - N \tag{8}$$

where $\bar{\pi}_N$ is the average proximity $\pi_x(z)$ for the N perturbations, s_N^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 PTG, we first relate W and α to Var(ϕ_i), variance of the marginal importance distribution for any feature i, obtained by integrating out σ^2.[2] Because Student's t can be approximated by a Normal distribution for large degrees of freedom (here, N should be large enough), we use the inverse normal CDF to calculate CI 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 S samples is the sample SSE s_N^2 scaled by $\approx \frac{4}{\bar{\pi}_N S}$ (derivation in appendix C). If we assume SSE scales linearly with N, we can take this to be a reasonable estimate of s_S^2 at any S. We can then estimate PTG as

$$\begin{aligned}
\left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2 &= \text{Var}(\phi_i) = \frac{4s_N^2}{\bar{\pi}_N \times S} \\
S &= \frac{4s_N^2}{\bar{\pi}_N \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} \\
PTG &= \frac{4s_N^2}{\bar{\pi} \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} - N.
\end{aligned} \tag{9}$$

□

3.3 Efficient Construction of Local Explanations

PTG enables users to understand how many samples are required to achieve reliable explanations, but for high PTG, LIME and SHAP users still face the computational burden

[1] PTG assumes sampling in a simplified feature space, in which features are either present or absent according to Bernoulli(.5). As in Ribeiro, Singh, and Guestrin (2016), these *interpretable* features are flexible and can be user-defined to appropriately encode what is important to the end user of the explanation.

[2] Note that the value of Var(ϕ_i), the marginal variance of the feature importance for feature i obtained by integrating out σ^2, is similar for all features. This is due to the fact that this variance captures random error which is a function of the number of perturbations and is therefore common across all features.

of querying the black-box model for its predictions on the perturbations (Denton et al. 2014; Jaderberg, Vedaldi, and Zisserman 2014). To reduce this cost, we further develop an alternative sampling procedure which uses uncertainty estimates to target the black box queries to regions with high uncertainty, reducing the computational cost of generating reliable explanations. Inspired by active learning (Settles 2010), we introduce a batch-sampling procedure called *uncertainty sampling* that strategically prioritizes perturbations whose predictions the explanation is most uncertain about. Specifically, we compute the posterior predictive distribution for any new instance z (derivation in Appendix B):

$$\hat{y}(z)|\mathcal{Z}, Y \sim t_{(\mathcal{V}=N)}(\hat{\phi}^T z, (z^T V_\phi z + 1)s^2) \tag{10}$$

The variance of the aforementioned three parameter student's t distribution is $((z^T V_\phi z + 1)s^2)(N/(N-2))$ (which we refer to as *predictive variance*) and captures how uncertain ϕ is about the black box label for an instance z. We prioritize querying the black box model with perturbations that have high predictive variance.

Our uncertainty sampling procedure proceeds as follows: (1) We first randomly sample N instances (perturbations) around the data point x. Let us call this set $\mathcal{Q}$. (2) We then compute $\exp(z^T V_\phi z + 1)s^2 / \sum_{z \in \mathcal{Q}} \exp(z^T V_\phi z + 1)s^2$ for each perturbation $z \in \mathcal{Q}$ thereby generating a distribution over all the points in $\mathcal{Q}$. Let us call this resulting distribution $\mathcal{Q}_{\text{dist}}$. (3) We then draw B perturbations from $\mathcal{Q}_{dist}$ and query the black box model for labels of these B instances. (4) Using the newly sampled B instances as well as the N initial perturbations and their corresponding black box labels, we fit a local linear model ϕ. (5) If the resulting explanation satisfies the desired level of certainty, terminate early and return the local explanation, otherwise, repeat steps (1) - (5). In practice, we observe that this procedure allows us to obtain explanations with desired levels of certainty with far fewer than PTG number of queries to the black box. Because we prioritize instances with uncertain predictions, we introduce some bias into the sampling procedure. Pseudocode for this procedure is provided in Appendix F.

4 Experiments

We evaluate the proposed framework by first analyzing the quality of the uncertainty estimates output by our framework for both feature importances and error. Next, we assess the correctness of our estimates of required perturbations (PTG), and evaluate the computational efficiency of uncertainty based sampling. Last, we describe a user study with 31 human 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 and credit scoring) as well as modalities (e.g., structured data, images). Our first structured dataset is **COMPAS** (Angwin et al. 2016), 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 (Dua and Graff 2017) 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** (LeCun, Cortes, and Burges 2010) 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** (Deng et al. 2009), we use the off-the-shelf VGG16 model (Simonyan and Zisserman 2015) 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 (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017b). For images, we construct super pixels as described in Ribeiro, Singh, and Guestrin (2016) and use them as the features to use in the explanation (number of super pixels is fixed to 20 per image). For our framework, we set perturbation sample 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.

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

Figure 3: **Assessing BayesLIME and BayesSHAP 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.

Quality of Uncertainty Estimates The key component of our explanation tools is the estimation of uncertainty (CIs) associated with the feature importances. 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 evaluate the quality of the CI estimates by running our methods with 100 perturbations to estimate feature importances and corresponding 95% CIs for each test instance. We compute what fraction of the true feature importances fall within our 95% CI estimates. Since we do not have access to the true feature importances of the complex black box models, following Prop 3.1, we use feature important computed from a large value of N ($N = 10,000$), and treat the resulting estimates as ground truth. Results for BayesLIME in Table 3 indicate that the true feature importances fall within estimated CIs about 94.8% to 97.2% of the time, confirming that these un-

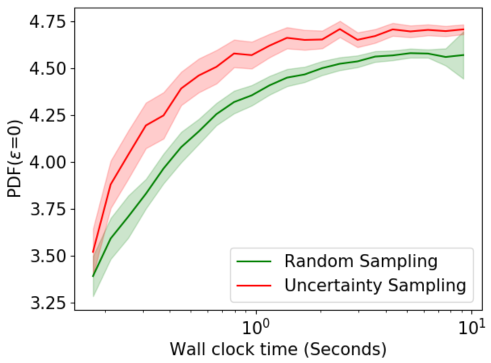


Figure 4: **Comparing random and uncertainty sampling** for 100 Imagenet images. We provide mean and standard error of the explanation quality (y-axis) over time. Uncertainty sampling converges much quicker than random sampling.

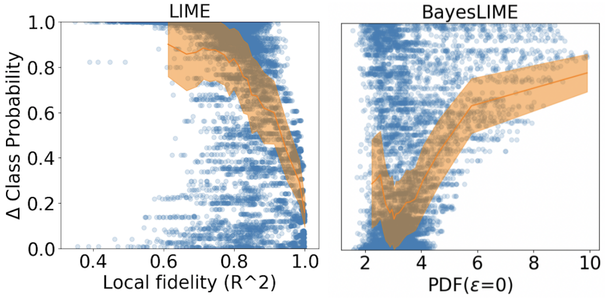


Figure 5: **Assessing explanation quality** (as measured by Δ class probability) when top 5% of super pixels using LIME fidelity and BayesLIME PDF($\epsilon = 0$) are masked, with different perturbation sizes & $1,000$ images (mean and std deviation in orange). PDF($\epsilon = 0$) has a *positive relationship* with the explanation quality while fidelity has *negative*; both significant as per Pearson's r test ($p < 10^{-20}$).

certainty estimates are very well calibrated. While the estimates by BayesSHAP are somewhat less calibrated (true feature importances fall within our estimated 95% CIs about 89.6 to 90.1% of the time), this may be due to the previously known drawback that the Shapley kernel produces extremely small proximity scores for perturbations (Lundberg and Lee 2017a), leading to instabilities in uncertainty estimation. All in all, these results confirm that the CIs learned by our methods are well calibrated and therefore highly reliable in capturing the uncertainty of the feature importances.

Explanation Error as Metric of Explanation Quality Recall that $P(\epsilon = 0)$ (Eqn. 7) gives us the probability density that an explanation perfectly captures the local decision surface of the underlying black box, and can therefore be used to evaluate the quality of explanations. Here, we assess if this notion of explanation quality is more reliable than another commonly used metric – locally weighted R^2 (local fidelity) (Tan et al. 2019; Ribeiro, Singh, and Guestrin 2016).

To compare which metric better correlates with explanation quality, we use Δ class-probability (Iwana, Kuroki, and Uchida 2019; Gu, Yang, and Tresp 2019), which is defined as $f(x) - f(x')$ and corresponds to the change in the probability output by the black box model f as we go from an instance x to another instance x', as a proxy for explanation quality. For two local linear models ϕ and ϕ' that approximate f around an image x, if ϕ is a better explanation than ϕ', removing the most important features highlighted by ϕ from image x should result in a larger change in the Δ class-probability than removing the most important features highlighted by ϕ'. We compare how explanations ranked using PDF($\epsilon = 0$) and fidelity compare to Δ class-probability, with higher correlation indicating a better metric for explanation quality. We take every image x in the MNIST test set, identify its top 5% most important features using our estimates of $\hat{\phi}$, mask these features to obtain a new image x', and measure the corresponding Δ class-probability, as well as the evaluation metrics PDF($\epsilon = 0$) for BayesLIME/BayesSHAP and local fidelity in case of LIME/SHAP. We repeat this varying number of perturbations, and show scatter plots of evaluation metrics vs. Δ class-probability in Figure 5 (results for SHAP and BayesSHAP are in Appendix). Δclass-probability has a *negative* relationship with local fidelity while it has a clear *positive* relationship with our metric P($\epsilon = 0$). This may be because local fidelity does not account for perturbation sample size i.e., an explanation with a weighted R^2 of 0.9 learned using 10 perturbations is considered better than another explanation with a weighted R^2 of 0.85 learned using 10,000 perturbations. These results clearly demonstrate that our metric P($\epsilon = 0$) is much more reliable than local fidelity in evaluating the quality of explanations.

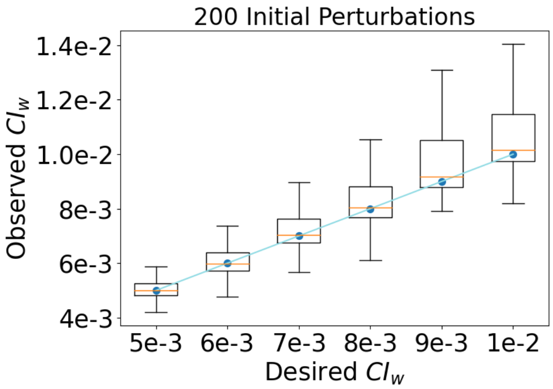


Figure 6: **Assessing PTG** by running for PTG perturbations computed using the *desired* CI_w (x-axis), and comparing it to the *observed* CI_w (y-axis) (blue line indicates ideal calibration). Results are averaged over 100 MNIST images, and PTG is estimated from $N = 200$ seed samples, varying between 200 and $20,000$ across images.

Correctness of Estimated Number of Perturbations We assess whether our estimate of *perturbations-to-go* (PTG) 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 and use $N = 200$ as the initial number of perturbations to obtain a preliminary explanation and its associated uncertainty estimates. We then leverage these estimates to compute PTG for 6 different certainty levels. For each image and certainty level, we run our method for the estimated number of perturbations (PTG) to determine if the resulting certainty levels (observed CI_w) match the desired levels of certainty (desired CI_w). Results in Figure 6 show that the observed and desired levels of certainty are well calibrated demonstrating that PTG estimates are reliable approximations of the additional number of perturbations needed. We also observed significant differences in PTG estimates across instances (details in appendix D) i.e. number of perturbations needed to obtain explanations with 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.

Efficiency of Uncertainty Sampling Recall that our *uncertainty sampling* procedure uses the *predictive variance* to strategically choose perturbations which will reduce uncertainty in order to be labeled by the black box. Here, we evaluate whether uncertainty sampling produces higher quality explanations (measured by $P(\epsilon = 0)$) more efficiently than random sampling. We experiment with BayesLIME on Imagenet data to carry out this analysis. This setting replicates real-world scenarios where LIME is not preferred as it is computationally expensive to query complex black boxes (e.g., VGG16). We run each sampling strategy for 10 seconds and plot wall clock time (computed on a machine with an Intel Core i9-9900 CPU) vs. explanation quality ($P(\epsilon = 0)$). Results in Figure 4 show that uncertainty sampling results in faster convergence to high quality explanations compared to random sampling; uncertainty sampling also stabilizes within a few seconds where as random sampling takes closer to 10 seconds. Results with number of model queries vs. explanation quality are included in Figure 11 in the Appendix. These results clearly demonstrate that uncertainty sampling can significantly speed up the process of generating high quality local explanations.

User Study We perform a user study to compare BayesLIME and LIME explanations on MNIST. We evaluate the following: are more confident explanations more meaningful for humans? To evaluate this question, we mask the most important features selected by BayesLIME and LIME, and ask users to determine the class. The better the explanation, the more difficult it should be for the users to guess the right digit. We randomly select 15 correctly predicted images from the MNIST test set, generate explanations by sweeping over the same perturbation sizes as Figure 5, and choose the *top* explanation for each image by either fidelity (for LIME) or $P(\epsilon = 0)$ (for BayesLIME). We find that the explanations identified by our method focus on more useful parts of the image for humans, since hiding them makes it difficult to guess the digit.

We sent the user study out to computer science students and colleagues through email and Slack channels. We provided instructions for the task using the text provided in Figure 13 in the Appendix. Users had an error rate for LIME of 25.7%, 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 measure of explanation confidence defined in BayesLIME selects better explanations that fidelity in LIME.

5 Related Work

Some prior work has attempted to tackle the problem of instability of explanations by averaging over several explanations (Yeh et al. 2019; Lee et al. 2019) or adopting a Bayesian non-parametric approach (Guo et al. 2018). However, these methods tend to be expensive in terms of computation. Further, these methods do not focus on modeling the uncertainty of explanations. In particular, (Guo et al. 2018) provides a similar approach which uses a Bayesian nonparametric mixture regression to generate explanations. Fixing their model to a single component, they achieve individual explanations. Our approach differs in a number of ways. Our focus is on assessing the uncertainty of our model while (Guo et al. 2018) do not assess uncertainty. Further, (Guo et al. 2018) does not use local weighting for individual explanations and thus cannot be used with LIME and SHAP. Also, our method is much more efficient (to corroborate these claims, we provide benchmarking tests in appendix H) Last, our method additionally provides useful information regarding where and how much to sample to reduce explanation uncertainty efficiently.

Additional methods quantify the uncertainty in causal explanations using bootstrapping (Schwab and Karlen 2019). Other works related to creating more trustworthy explanations include development of sanity checks for explainers (Camburu et al. 2019; Adebayo et al. 2018; Yang and Kim 2019). These techniques represent an important step to improved usability given experimental evidence that humans are often too eager to accept inaccurate machine explanations (Kaur et al. 2020; Hohman et al. 2019; Poursabzi-Sangdeh et al. 2018; Lakkaraju and Bastani 2020).

6 Conclusion

We propose a novel Bayesian framework that models the uncertainty associated with local explanations. The uncertainty estimates in the form of credible intervals (CIs) output by our framework are not only informative in assessing the quality of local explanations, but also in estimating critical hyperparameters and making the process of learning local explanations highly efficient. Although our Bayesian framework provides good uncertainty estimates, one potential downside is setting strong priors. Strong priors could influence the explanations and may produce misleading results downstream. All in all, our contribution paves the way for future work measuring other sources of uncertainty in explanations and exploring how this uncertainty quantification can reduce errors of algorithmic overconfidence in domains such as healthcare, criminal justice, and business.

Ethics Statement

Interpretability of complex black box models in machine learning is a quickly growing area of research with immediate societal considerations. Our work addresses issues of explanation methods unreliability through better expressing notions of explanation uncertainty. This method could better allow users to understand whether they have generated reliable, replicable model explanations. In particular, it could provide guidelines for when *not to* trust any given explanation. Such endeavors could have positive downstream societal outcomes through mitigating the effects of faulty model explanations, and open up potential applications domains for interpretable machine learning.

Though this methods presents potential societal upsides, there are potential negative outcomes considering the context and use case of the method. The primary concern we see is that users could potentially conflate low sampling uncertainty with unrelated sources of uncertainty, in particular model uncertainty. This may exacerbate the effect of explanations leading to justification and overtrust of inaccurate model predictions (Kaur et al. 2020; Hohman et al. 2019; Poursabzi-Sangdeh et al. 2018). Furthermore, our method identifies only *sample uncertainty* for a fixed perturbation distribution and so does not protect against adversarial attacks on explanation methods which leverage gaps between the data distribution and the perturbation distribution (Slack et al. 2020). Mitigating the above concerns will require broader initiatives to educate users around the specific use cases and failings of explanation methods (Sokol and Flach 2020) as well as interdisciplinary efforts with social psychologists and HCI practitioners to make the potential errors of explanation methods more salient.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 9505–9515.

Alvarez-Melis, D.; and Jaakkola, T. S. 2018. On the Robustness of Interpretability Methods. *ICML Workshop on Human Interpretability in Machine Learning* .

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine bias. In *ProPublica*.

Bastani, O.; Kim, C.; and Bastani, H. 2017. Interpretability via model extraction. *FAT/ML Workshop 2017* .

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Camburu, O.-M.; Giunchiglia, E.; Foerster, J.; Lukasiewicz, T.; and Blunsom, P. 2019. Can I Trust the Explainer? Verifying Post-hoc Explanatory Methods. *arXiv preprint arXiv:1910.02065* .

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2019. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *International Conference on Learning Representations*.

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

Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R. 2014. Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation. In *NIPS*.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Fahrmeir, L.; Kneib, T.; and Lang, S. 2007. *Regression*. Statistik und ihre Anwendungen. Berlin [u.a.]: Springer. ISBN 978-3-540-33932-8. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+510939260&sourceid=fbw_bibsonomy.

Ghorbani, A.; Abid, A.; and Zou, J. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3681–3688.

Gruber, S.; and Molnar, C. 2020. LIME and Sampling. URL https://compstat-lmu.github.io/iml_methods_limitations/lime-sample.html.

Gu, J.; Yang, Y.; and Tresp, V. 2019. Understanding Individual Decisions of CNNs via Contrastive Backpropagation. In *Asian Conference on Computer Vision*, 119–134. ISBN 978-3-030-20892-9. doi:10.1007/978-3-030-20893-6_8.

Guo, W.; Huang, S.; Tao, Y.; Xing, X.; and Lin, L. 2018. Explaining Deep Learning Models – A Bayesian Non-parametric Approach. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 4514–4524. Curran Associates, Inc. URL http://papers.nips.cc/paper/7703-explaining-deep-learning-models-a-bayesian-non-parametric-approach.pdf.

Hohman, F.; Head, A.; Caruana, R.; DeLine, R.; and Drucker, S. M. 2019. 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*, 1–13.

Iwana, B.; Kuroki, R.; and Uchida, S. 2019. Explaining Convolutional Neural Networks using Softmax Gradient Layer-wise Relevance Propagation. In *IEEE/CVF International Conference on Computer Vision Workshop*, 4176–4185.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Speeding up Convolutional Neural Networks with Low Rank Expansions. *BMVC 2014 - Proceedings of the British Machine Vision Conference 2014* doi:10.5244/C.28.88.

Kaur, H.; Nori, H.; Jenkins, S.; Caruana, R.; Wallach, H.; and Wortman Vaughan, J. 2020. Interpreting Interpretability: Understanding Data Scientists' Use of Interpretability Tools for Machine Learning. In *CHI 2020*.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1885–1894. JMLR. org.

Kumar, I. E.; Venkatasubramanian, S.; Scheidegger, C.; and Friedler, S. Forthcoming 2020. Problems with Shapley-value-based explanations as feature importance measures. *International Conference on Machine Learning* .

Lakkaraju, H.; and Bastani, O. 2020. " 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*, 79–85.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Leskovec, J. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 131–138. ACM.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* 2.

Lee, E.; Braines, D.; Stiffler, M.; Hudler, A.; and Harborne, D. 2019. Developing the sensitivity of LIME for better machine learning explanation. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, 1100610. International Society for Optics and Photonics.

Lundberg, S. M.; and Lee, S.-I. 2017a. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 4765–4774.

Lundberg, S. M.; and Lee, S.-I. 2017b. A Unified Approach to Interpreting Model Predictions. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 4765–4774. Curran Associates, Inc. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Moore, A. 1995. Locally Weighted Bayesian Regression.

Plumb, G.; Molitor, D.; and Talwalkar, A. S. 2018. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, 2515–2524.

Poursabzi-Sangdeh, F.; Goldstein, D. G.; Hofman, J. M.; Vaughan, J. W.; and Wallach, H. 2018. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810* .

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Schwab, P.; and Karlen, W. 2019. CXPlain: Causal Explanations for Model Interpretation under Uncertainty. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 10220–10230. Curran Associates, Inc. URL http://papers.nips.cc/paper/9211-cxplain-causal-explanations-for-model-interpretation-under-uncertainty.pdf.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Settles, B. 2010. Active learning literature survey.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop at International Conference on Learning Representations*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; and Lakkaraju, H. 2020. Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods. *AAAI/ACM Conference on Artificial Intelligence, Ethics, and Society (AIES)* .

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *Workshop on Visualization for Deep Learning, ICML* .

Sokol, K.; and Flach, P. 2020. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 56–67.

Sokol, K.; Hepburn, A.; Santos-Rodríguez, R.; and Flach, P. A. 2019. bLIMEy: Surrogate Prediction Explanations Beyond LIME. *NeurIPS HCML Workshop* .

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3319–3328. JMLR. org.

Tan, H. F.; Song, K.; Udell, M.; Sun, Y.; and Zhang, Y. 2019. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. In *ICML Workshop on AI for Social Good*.

Yang, M.; and Kim, B. 2019. BIM: Towards quantitative evaluation of interpretability methods with ground truth. *arXiv preprint arXiv:1907.09701* .

Yeh, C.-K.; Hsieh, C.-Y.; Suggala, A.; Inouye, D. I.; and Ravikumar, P. K. 2019. On the (In) fidelity and Sensitivity of Explanations. In *Advances in Neural Information Processing Systems*, 10965–10976.

A BayesLIME Case Study

We consider two explanations output by our method BayesLIME to explain a random forest classifier (black box) trained on the COMPAS dataset. One of these explanations is generated using fewer perturbations (100), and the other is generated using larger number of perturbations (5000). Figures 7 & 8 show the top 5 features corresponding to each of these explanations, the associated feature importance distributions. In case of few perturbations, BayesLIME suggests that the true feature importance values could vary significantly from the mean estimates. With several perturbations, the feature importances are much more certain. Additionally, the importance ranking amongst the features has changed significantly. `sex` turns out to be the most important feature while `priors count` has dropped in relative importance. Though the feature importances change significantly with additional perturbations, BayesLIME correctly captures this uncertainty even with fewer perturbations.

B Model derivation

From the assumption that σ^2 is uninformative, namely that the prior parameters are about 0, we write the joint posterior as

$$\phi, \sigma^2 | Y, \mathcal{Z} \propto \rho(Y|X, \beta, \sigma^2)\rho(\beta|\sigma^2)\rho(\sigma^2) \tag{11}$$

$$\propto (\sigma^2)^{-N/2}\exp(-\frac{1}{2\sigma^2}(Y - \mathcal{Z}\phi)^T \text{diag}(\Pi_x(\mathcal{Z}))\cdot (Y - \mathcal{Z}\phi))(\sigma^2)^{-1}\exp(-\frac{1}{2\sigma^2}\phi^T\phi) \tag{12}$$

Letting $\hat{\phi} = (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1}\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))Y$, we group terms in the exponentials according to ϕ. The intermediate steps can be found in (Fahrmeir, Kneib, and Lang 2007).

$$= (Y - X\phi)^T \text{diag}(\Pi_x(\mathcal{Z}))(Y - X\phi) + \phi^T\phi \tag{13}$$

$$= [\phi - \hat{\phi}]^T(\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)[\phi - \hat{\phi}] \tag{14}$$

Using what we've derived so far, we can write down the conditional posterior of ϕ as

$$\phi|\sigma^2, Y, \mathcal{Z} \propto \exp(\frac{1}{2}\sigma^{-2}[\phi - \hat{\phi}]^T(\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)[\phi - \hat{\phi}]) \tag{15}$$

So, we can see that our estimates for the mean and variance of $\rho(\phi|\sigma^2, Y, \mathcal{Z})$ are $\hat{\phi}$ and $\sigma^2(\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1}$.

Next, we derive the conditional posterior for σ^2. We identify the form of the scaled inverse-χ^2 distribution in the joint posterior as in (Moore 1995) and write

$$\sigma^2|\mathcal{Z}, \hat{\phi}, Y \sim \text{Inv-}\chi^2(N, s^2) \tag{16}$$

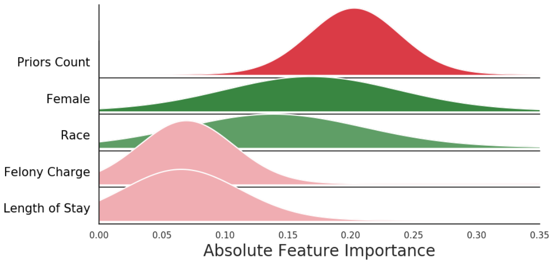


Figure 7: Explanations learned using 100 perturbations. Top 5 features (priors count being the most important) and their corresponding feature importance distributions.

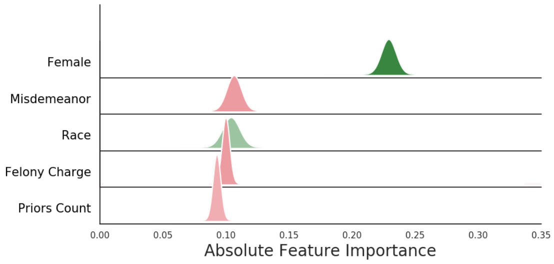


Figure 8: Explanations learned using 5000 perturbations. Top 5 features (sex being the most important) and their corresponding feature importance distributions.

where

$$s^2 = \frac{(y - \mathcal{Z}\hat{\phi})^T\text{diag}(\Pi_x(\mathcal{Z}))(y - \mathcal{Z}\hat{\phi}) + \hat{\phi}^T\hat{\phi}}{N} \tag{17}$$

Derivation of equation 7 We establish the identity (Moore 1995):

$$\sigma^2 \sim \text{Inv-}\chi^2(a, b) \text{ and } z|\sigma^2 \sim \mathcal{N}(\mu, \lambda\sigma^2) \iff z \sim t_{(\mathcal{V}=a)}(\mu, \lambda b) \tag{18}$$

We have, $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 \sim \text{Inv-}\chi^2(N, s^2)$. Then, it's the case that $\epsilon \sim t_{(\mathcal{V}=N)}(0, s^2)$.

Derivation of equation 10 We apply the identity from equation 18 to derive this posterior. We have $\hat{y} \sim \hat{\phi}^T z + \epsilon$ for some z. Thus, $\hat{y} \sim \mathcal{N}(\hat{\phi}^T z, z^T V_\phi z\sigma^2) + \mathcal{N}(0, \sigma^2)$, where $\sigma^2 \sim \text{Inv-}\chi^2(N, s^2)$. So, we have $\hat{y} \sim t_{(\mathcal{V}=N)}(\hat{\phi}^T z, (z^T V_\phi z + 1)s^2)$.

C Proof of Theorems

In this appendix, we prove proposition 3.1 and theorem 3.2.

In these derivations, the perturbation matrices $\mathcal{Z}$ have elements $\mathcal{Z}_{ij} \in \{0, 1\}$ where each $\mathcal{Z}_{ij} \sim \text{Bernoulli}(0.5)$. This convention is typically used to denote features being "included" (1) or "excluded" (0) (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017b).

We initially prove theorem 3.2 because the results contained in this proof are useful for the propositions.

C.1 Proof of Theorem 3.2

We use three assumptions stated as follows. First, $\frac{\bar{\pi}S}{2}$ is sufficiently large such at $\frac{\bar{\pi}S}{2} + 1$ is equivalent to $\frac{\bar{\pi}S}{2}$. Second, S is sufficiently large such that $S+1$ is equivalent to S and $\frac{S}{S-2}$ is equivalent to 1. Third, the product of $\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z}$ within V_ϕ can be taken at its expected value.

Note that we use S to denote the *total* perturbations while we use N to denote the perturabtions collected *so far*.

First, we state the marginal distribution over feature importance ϕ_i where i is an arbitrary feature importance $i \in d$. This given as

$$\phi_i|\mathcal{Z}, Y \sim t_{\mathcal{V}=N}(\hat{\phi_i}, V_{\phi_{ii}} s^2) \tag{19}$$

where $V_\phi = (\mathcal{Z}^T\text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1}$. Recalling each element of $\mathcal{Z}$, i.e. $\mathcal{Z}_{ij}$, is given $\sim$ Bern$(.5)$ we use the third assumption to write V_ϕ

$$V_\phi = \begin{bmatrix} \frac{\bar{\pi}S}{2} + 1 & \frac{\bar{\pi}S}{4} & \cdots \\ \frac{\bar{\pi}S}{4} & \ddots & \\ \vdots & & \frac{\bar{\pi}S}{2} + 1 \end{bmatrix}^{-1} \tag{20}$$

We can see this is the case considering that each element in $\mathcal{Z}$ is a Bern$(.5)$ draw. So, the diagonals are scaled by $S/2$ because these are the expected value of the dot product of each row with itself. The off-diagonals are scaled by $S/4$ considering this is expected value of the dot product of a row with a row besides itself. Dropping $1's$ due the first assumption

$$V_\phi = \begin{bmatrix} \frac{\bar{\pi}S}{2} & \frac{\bar{\pi}S}{4} & \cdots \\ \frac{\bar{\pi}S}{4} & \ddots & \\ \vdots & & \frac{\bar{\pi}S}{2} \end{bmatrix}^{-1} \tag{21}$$

where $\frac{\bar{\pi}S}{2}$ defines the diagonal and $\frac{\bar{\pi}S}{4}$ defines the off diagonal elements. Through the Sherman-Morrison formula, we can write the inverse of this matrix as

$$V_\phi = \left[\begin{bmatrix} \frac{\bar{\pi}S}{2} - \frac{\bar{\pi}S}{4} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & \frac{\bar{\pi}S}{2} - \frac{\bar{\pi}S}{4} \end{bmatrix} + \frac{\bar{\pi}S}{4} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}^T \right]^{-1} \tag{22}$$

Let $k = \frac{\bar{\pi}S}{2}$. It follows directly from Sherman Morrison that the i-th and j-th entries of V_ϕ are given as

$$(V_\phi)_{ij} = \begin{cases} \frac{2}{k} - \frac{2}{k(S+1)} & i = j \\ -\frac{2}{k(S+1)} & i \neq j \end{cases} \tag{23}$$

$$(V_\phi)_{ii} = \frac{4}{\bar{\pi}(S+1)} \tag{24}$$

We see that the diagonals are the same. Thus, we take the PTG estimate in terms of a single marginal ϕ_i. Substituting in the s^2 estimate s_N^2 and using the second assumption, we write the variance of marginal ϕ_i as

$$\text{Var}(\phi_i) = \frac{4s_N^2}{\bar{\pi}(S+1)} \frac{S}{S-2} \tag{25}$$

$$= \frac{4s_N^2}{\bar{\pi} \times S} \tag{26}$$

Thus, the *total* number of samples needed is

$$S = \frac{4s_N^2}{\bar{\pi} \times \text{Var}(\phi_i)} \tag{27}$$

Because our notion of feature importance uncertainty is in the form of a credible interval, we use the normal approximation of Var(ϕ_i) and write

$$S = \frac{4s_N^2}{\bar{\pi} \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} \tag{28}$$

where W is the desired width, α is the desired confidence level, and $\Phi^{-1}(\alpha)$ is the two-tailed inverse normal CDF. Finally, we subtract the initial N samples. Thus, PTG is given as

$$PTG(W, \alpha, x) = \frac{4s_N^2}{\bar{\pi}_N \times \left[\frac{W}{\Phi^{-1}(\alpha)}\right]^2} - N \tag{29}$$

□

C.2 Discussion of assumptions from Theorem 3.2

Binary z's We refer readers to footnote 1 in section 3.
SSE scaling linearly with N Though the rate at which SSE increases may vary, the assumption that SSE scales linearly results in empirically good estimates of PTG (figure 6).

C.3 Proposition 3.1

Before providing a proof for proposition 3.1, we note to readers that the claims are related to well known results in bayesian inference (e.g. similar results are proved in (Bishop 2006)). We provide the proofs here to lend formal clarity to the properties of our explanations.

We outline three claims in the proposition. Namely, (1) Var$(\phi) \to 0$ as $N \to \infty$ (2) the mean of ϕ is consistent and (3) Var(ϵ) converges to the bias of the local model as $N \to \infty$

Convergence of Var(ϕ) Recall the posterior distribution of ϕ given in equation 4. In equation 22, we see the on and off-diagonal elements of V_ϕ are given as $\frac{4}{\pi(N+1)}$ and $-\frac{4}{\pi N(N+1)}$ respectively (here replacing S with N to stay consistent with equation 4). Because we have $N \to \infty$, these values define V_ϕ due to the law of large numbers. Thus, as $N \to \infty$, V_ϕ goes to the null matrix and so does the uncertainty over ϕ.

Consistency of $\hat{\phi}$ Recall the mean of ϕ, denoted $\hat{\phi}$ given in equation 5. To establish consistency, we must show that $\hat{\phi}$ converges in probability to the true $\hat{\phi}$ as $n \to \infty$. To avoid confusing true $\hat{\phi}$ with the distribution over ϕ, we denote the true $\hat{\phi}$ as ϕ^*. Thus, we must show $\hat{\phi} \to_p \phi^*$ as $n \to \infty$. We write

$$\hat{\phi} = (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1} \mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))Y \tag{30}$$

$$= (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1} \mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))(\mathcal{Z}\phi^* + \epsilon) \tag{31}$$

Considering the mean of ϵ is 0

$$= (\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I)^{-1} \mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z}\phi^* \tag{32}$$

Through the law of large numbers

$$\begin{aligned} &= (n^{-1}[\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z} + I])^{-1} n^{-1} \times \\ &\mathcal{Z}^T \text{diag}(\Pi_x(\mathcal{Z}))\mathcal{Z}\phi^* \end{aligned} \tag{33}$$

$$= \phi^* \tag{34}$$

which establishes the claim.

Convergence of Var(ϵ) Assume we have $N \to \infty$ so $\hat{\phi}$ converges to ϕ^*. The uncertainty over the additive error term is given as the variance of the distribution in equation 7. The variance of this generalized student's t distribution is given as

$$s^2 \frac{N}{N-2} \tag{35}$$

which for large N is s^2. Recalling the definition of s^2 (see equation 17), s^2 reduces to the local error of the model as $N \to \infty$, namely

$$s^2 = \frac{(y - \mathcal{Z}\hat{\phi})^T \text{diag}(\Pi_x(\mathcal{Z}))(y - \mathcal{Z}\hat{\phi})}{N} \tag{36}$$

which is equivalent to the squared bias of the local model, considering that there is no uncertainty over ϕ.

D Detailed Results

Quality of Uncertainty Estimates In section 4, we assessed whether our uncertainty estimates are well calibrated and presented results for BayesLIME and BayesSHAP. Here, we demonstrate that the BayesLIME uncertainty estimates capture the uncertainty within the original LIME framework. In Table 1, we show the results when we rerun LIME for each image using $10,000$ perturbations and find the estimates to be similarly within the BayesLIME CI estimates. We cannot compute this for SHAP because the KernelSHAP implementation from (Lundberg and Lee 2017a) does not randomly select perturbations.

Data set	Model	LIME % Within CI
ImageNet	VGG16	94.8
MNIST	CNN	97.2
COMPAS	Random Forest	97.6
German Credit	Random Forest	96.9

Table 1: We assess the credible interval estimates of BayesLIME by computing how often the true feature importance values for LIME falls within our 95% CI estimate. Though we cannot get a true feature importance estimate, we compute the feature importance at a very high sampling size ($10,000$ perturbations) and assess how often these values fall within the CIs of our explanations computed at 100 perturbations.

Explanation Error as a Metric of Explanation Quality In section 4, we evaluated whether $P(\epsilon = 0)$ is a better metric for explanation quality than locally weighted R^2 and showed results for BayesLIME. Here, we demonstrate similar results for BayesSHAP. SHAP does not support locally weighted R^2, so we compare BayesSHAP ranked by $P(\epsilon = 0)$ with BayesSHAP ranked by local fidelity. We evaluate each image in the MNIST testing set, sweeping over a range of perturbation amounts ($[100, 150, 200, 250, 300, 350, 400, 450, 500]$) and assess Δclass-probability. The results in Figure 9 show a *negative* relationship between fidelity and Δclass-probability, and demonstrate a *postive* relationship between our metric and Δclass-probability. This may be because local fidelity does not account for perturbation sample size.

Correctness of the Estimated Number of Perturbations In section 4, we assessed if PTG produces good estimates of the number of additional samples needed to reach the desired level of feature importance certainty. In figure 10, we show the desired level of certainty (desired width of credible interval CI_w) versus the actual PTG estimate (i.e. the estimated number of perturbations) for figure 6 in the main paper. We see the estimated number of perturbations is highly variable depending on desired CI_w. For the lowest levels of certainty, PTG ranges from 200 to 5000 perturbations. For the highest levels of certainty considered, PTG ranges from 200 to 20000.

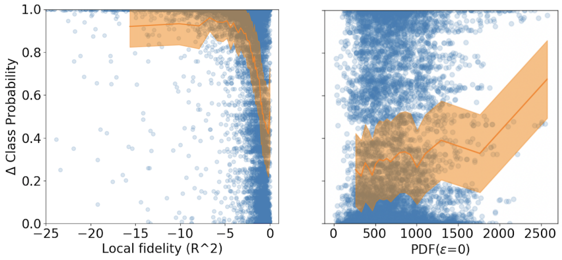


Figure 9: Δ class probability experiment from figure 5 repeated for BayesSHAP. The line provided is the mean and standard deviation of binned points. We see that our PDF($\epsilon = 0$) metric has a *positive relationship* with explanation quality as a measured by Δ class probability while fidelity has a *negative relationship*. The relationships are significant according to a Pearson's correlation coefficient test ($p < 1e - 20$ in both cases). The local fidelity term for SHAP is very low because the resulting Shapley values tend to be extremely small or large, which can lead to poor locally weighted R^2.

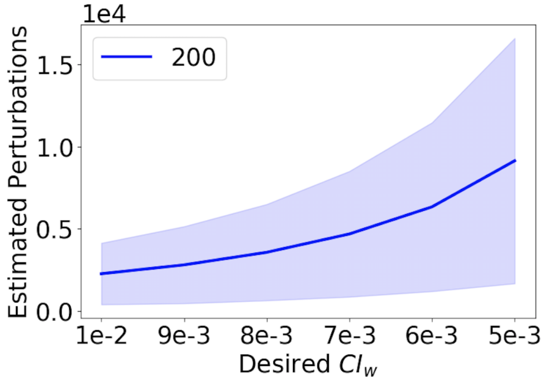
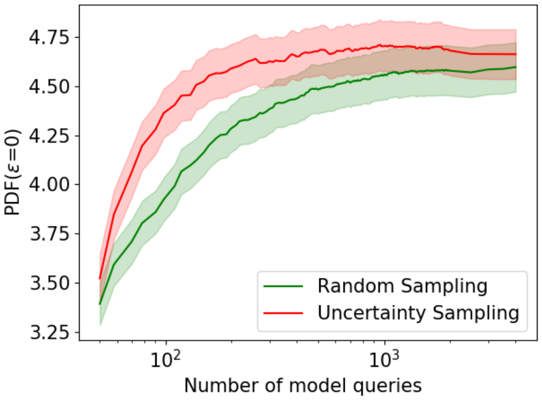


Figure 10: Desired CI_w versus the actual number of perturbations estimated by PTG in figure 6 of the main paper. We plot mean and standard deviation of PTG.

Efficiency of Uncertainty Sampling In section 4, we evaluated whether uncertainty sampling produces quicker convergence to high quality explanations and presented results plotting wall clock time versus $P(\epsilon = 0)$. In figure 11, we plot the number of model queries versus $P(\epsilon = 0)$. This experiment is analogous to figure 4 in the main paper, but here we use the number of model queries instead of time on the x-axis. We see that uncertainty sampling is more query efficient than random sampling for BayesLIME.

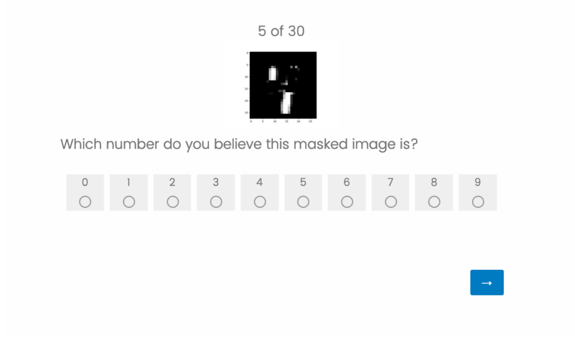
E User study

In this appendix, we give an example screen shot from the user study in figure 12. We also provide the instructions given to user study participants in figure 13.



(a) BayesLIME model queries.

Figure 11: Assessment of the number of model queries needed to converge to a high quality explanation (analogous to figure 4 in the paper). We use both random sampling and uncertainty sampling over 100 Imagenet images. We provide the mean and standard error for binned estimates of these values.



5 of 30

Which number do you believe this masked image is?

0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8 ○ 9 ○

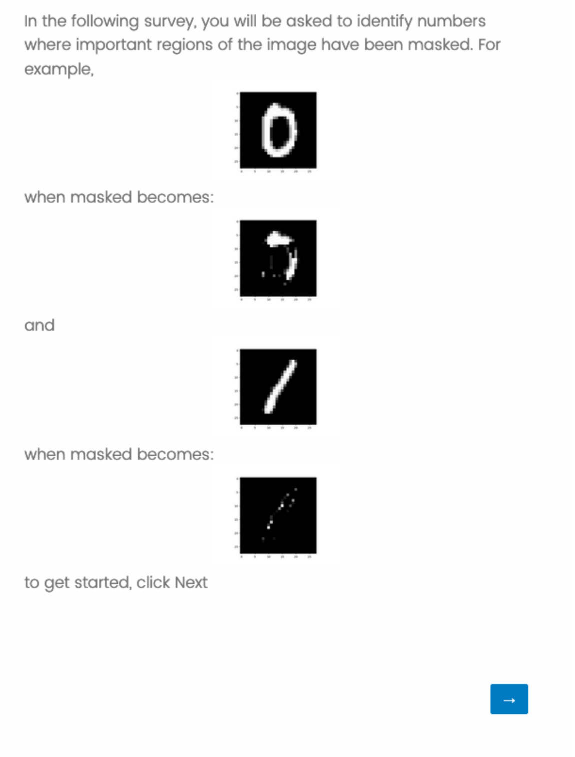
Figure 12: Screen shot from user study (correct answer 4).

F Uncertainty sampling algorithm

Here, we provide pseudo code for the uncertainty sampling procedure in algorithm 1.

G Additional BayesSHAP details

The BayesSHAP implementation is different from the BayesLIME implementation in the use of the weights — i.e. using the Shapley kernel instead of the LIME Kernel. In addition, BayesSHAP uses the sample perturbation strategy from the original KernelSHAP implementation in (Lundberg and Lee 2017a). This means we always treat all the variables as binary $\{0, 1\}$ variables where 1 represents the variable being included in the perturbation and 0 meaning its excluded. When the variable is excluded, a value from a background distribution is used. We always set this background distribution to the set of training data. For example when using BayesSHAP on images, we include the superpixel if the value is 1; if it is 0, the value 0 is substituted into the superpixel. We proceed to get the prediction using the



In the following survey, you will be asked to identify numbers where important regions of the image have been masked. For example,

when masked becomes:

and

when masked becomes:

to get started, click Next

Figure 13: The instructions provided to user study participants.

perturbed instance. Then we regress on a binary $\{0, 1\}$ regression problem. With tabular data, we do the same. For instances, if the range of values for a feature is $\{2.1, 3, 5, 10\}$, we randomly draw one of these values and substitute it into the instance, collecting the label of the perturbed data instance. We again perform regression on binary $\{0, 1\}$ data.

Our method only differs slightly from the implementation of KernelSHAP in (Lundberg and Lee 2017a) when it comes to how we select the perturbations. (Lundberg and Lee 2017a) generates perturbations through sampling coalitions in order of their importance according to the Shapley kernel. To generate good explanations while ensuring an adequate level of stochasticity for generating well calibrated uncertainty estimates, we first enumerate all the singleton coalitions. Then, we proceed to randomly sample coalitions.

H Benchmarking

We benchmark BayesLIME and BayesSHAP against a related Bayesian explanation method (Guo et al. 2018) to demonstrate their efficiency. The technique provided in (Guo et al. 2018) 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, the authors find that their method takes 139.2 seconds. Under the same conditions, our method takes 20.3 seconds for BayesLIME and 21.1 seconds for BayesSHAP. We perform this experiment using an Intel Core i9-9900 CPU and GeForce RTX 2080 Ti GPU. These results clearly demonstrate that our closed form solution to produce Bayesian local explanations is very efficient.

Algorithm 1 Uncertainty sampling for local explanations

```
Require: Perturbation size $S$, Preliminary perturbation size $N$, Batch size $B$, Model $f$, Data instance $X$, Explanation Model $\phi$ with Predictive Variance $\phi_g$, Candidate perturbation batch size $\mathcal{A}$
function UNCERTAINTY SAMPLE
    Initialize data set $\mathcal{D}$ and add $N$ initial perturbations, $(\mathcal{Z}, f(\mathcal{Z}))$.
    Fit $\phi$ on $\mathcal{D}$
    for $i \leftarrow 1$ to $S - N$ do
        if $i$ mod. $B = 0$ then
            Generate set of candidate perturbations $\mathcal{Q}$ of size $\mathcal{A}$
            Draw $B$ perturbations into $\mathcal{Q}_{\text{new}}$ from $\mathcal{Q}_{\text{dist}} \sim \exp(\phi_g(\mathcal{Q}))_{j \in |\mathcal{Q}|} / \sum \exp(\phi_g(\mathcal{Q}))$
            $\mathcal{D} \leftarrow \mathcal{D} \cup (\mathcal{Q}_{\text{new}}, f(\mathcal{Q}_{\text{new}}))$; Fit $\phi$ on $\mathcal{D}$
        end if
    end for
    return $\phi$
end function
```