A SOCIETAL IMPACTS

As machine learning models are increasingly relied upon in a diverse set of high-impact domains ranging from health-care to financial lending (Esteva et al., 2019; Kose et al., 2021; Doshi-Velez & Kim, 2017; Sheikh et al., 2020; Singh et al., 2021), it is crucial that users of these models can accurately interpret why predictions are made. An understanding of why a model is making a certain prediction is important for users to trust it – for instance a doctor may wish to know if a skin-cancer classifier's high test-set accuracy comes from the leveraging of truly diagnostic features, or a specific imaging device artifact. However, further spurred by the advent of deep learning's increasing popularity (Krizhevsky et al., 2017), many of the models deployed in these high-stakes fields are complex black box's; producing predictions which are non-trivial to explain the reasoning behind. The development of many methods for explaining black-box predictions has arisen from this situation (Ribeiro et al., 2016; Lundberg & Lee, 2017; Covert et al., 2020; Masoomi et al., 2021), but explanations may have varying quality and consistency. Before utilizing explanations in practice, it is essential that users know when, and when not, to trust them. Explanation uncertainty is one proxy for this notion of trust, in which more uncertain explanations may be deemed less trustworthy. In this work, we explore a new way to model explanation uncertainty, in terms of local decision-boundary complexity. In tandem with the careful consideration of domain experts, our methodology may be used to assist in determining when explanations are reliable. Our theoretical results provide new insights towards what explanation uncertainty entails, and open the door for future methods expounding upon our formulation.

B BACKGROUND

B.1 RELATED WORKS: RELIABILITY OF EXPLANATIONS

While feature attribution methods have gained wide popularity, a number of issues relating to the reliability of such methods have been uncovered. Alvarez-Melis & Jaakkola (2018) investigate the notion of robustness and show that many feature attribution methods are sensitive to small changes in input. This has been further investigated in the adversarial setting for perturbation-based methods (Slack et al., 2020) and neural network-based methods. (Ghorbani et al., 2019). Kindermans et al. (2019) show that many feature attribution methods are affected by distribution transformations such as those common in preprocessing. The generated explanations can also be very sensitive to hyperparameter choice Bansal et al. (2020). A number of metrics have been proposed for evaluating explainer reliability, such as with respect to adversarial attack (Hsieh et al., 2021), local perturbations (Alvarez-Melis & Jaakkola, 2018; Visani et al., 2022), black-box smoothness (Khan et al., 2022), fidelity to the black-box model (Yeh et al., 2019), or combinations of these metrics (Bhatt et al., 2020).

B.2 GAUSSIAN PROCESS REVIEW

A single-output Gaussian Process represents a distribution over *functions* $f : \mathcal{X} \to \mathbb{R}$

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \tag{10}$$

Here $m : \mathcal{X} \to \mathbb{R}$ and $k : (\mathcal{X}, \mathcal{X}) \to \mathbb{R}$ are the mean and kernel (or covariance) functions respectively, which are chosen *a priori* to encode the users assumptions about the data. The kernel function $k(x, x')$ reflects a notion of similarity between data points for which predictive distributions over $f(x), f(x')$ respect. The prior $m(x)$ – frequently considered to be less important – is commonly chosen to be the constant $m(x) = 0$.

Specifically, a GP is an infinite collection of R.V's $f(x)$, each indexed by an element $x \in \mathcal{X}$. Importantly, any finite sub-collection of these R.V's

$$f(X_{tr}) = (f(x_1) \ldots, f(x_n)) \in \mathbb{R}^d, \tag{11}$$

corresponding to some index set $X_{tr} = \{x_i\}_{i=1}^n \subset \mathcal{X}$, follows the multivariate normal (MVN) distribution, i.e.

$$f(X_{tr}) \sim \mathcal{N}(m(X_{tr}), K(X_{tr}, X_{tr})). \tag{12}$$

Here the mean vector $m(X_{tr}) = (m(x_1), \ldots, m(x_n)) \in \mathbb{R}^n$ represents the mean function applied on each $x \in X_{tr}$ and the covariance matrix $K \in \mathbb{R}^{n \times n}$, also known as the gram matrix, contains

each pairwise kernel-based similarity value $K_{ij} = k(x_i, x_j)$. Kernel function outputs correspond to dot products in potentially infinite dimensional expanded feature space, which allows for the encoding of nuanced notions of similarity; e.g. the exponential geodesic kernel referenced in this work (Feragen et al., 2015).

Making predictions with a GP is analogous to simply conditioning this normal distribution on our data. Considering a set of input,noise-free label pairs

$$\mathcal{D} = \{(x_i, f(x_i)\}_{i=1}^n \tag{13}$$

we may update our posterior over *any subset* of the R.V's $f(x)$ by considering the joint normal over the subset and $\mathcal{D}$ and conditioning on $\mathcal{D}$. For instance, when choosing a singleton index set $\{x_0\}$, the posterior over $f(x_0)|D$ is another normal distribution which may be written as[2]

$$f(x_0) \sim \mathcal{N}(\bar{f}(x_0), \mathbb{V}[f(x_0)]) \tag{14}$$

where

$$\bar{f}(x_0) = K(x_0, X_{tr})^T K(X_{tr}, X_{tr})^{-1} f(X_{tr}) \tag{15}$$

$$\mathbb{V}[f(x_0)] = k(x_0, x_0) - K(x_0, X_{tr})^T K(X_{tr}, X_{tr})^{-1} K(x_0, X_{tr}) \tag{16}$$

and $K(x_0, X_{tr}) \in \mathbb{R}^d$ is defined element-wise by $K(x_0, X_{tr})_i = k(x_0, x_i)$.

Now we may consider the situation where our labels are noisy:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n,\ y_i = f(x_i) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2),\ \sigma^2 \in \mathbb{R}_+. \tag{17}$$

Here y_i is equal to the quantity we wish to model, $f(x_i)$, with the addition of noise variable ϵ. The conditional is still a MVN, but the mean and variance equations are slightly modified

$$\bar{f}(x_0) = K(x_0, X_{tr})^T (K(X_{tr}, X_{tr}) + \sigma^2 I)^{-1} Y \tag{18}$$

$$\mathbb{V}[f(x_0)] = k(x_0, x_0) - K(x_0, X_{tr})^T (K(X_{tr}, X_{tr}) + \sigma^2 I)^{-1} K(x_0, X_{tr}), \tag{19}$$

where $Y \in \mathbb{R}^n$ has elements $Y_i = y_i$.

Notice how the variance $\sigma^2 I$ is added to $K(X_{tr}, X_{tr})$ in the quadratic form in Eq. 19, resulting in smaller eigenvalues after matrix inversion. Since this quadratic form is subtracted, the decision to model labels as noisy increases the uncertainty (variance) of estimates the GP posterior provides. This agrees with the intuition that noisy labels should result in more uncertain predictions.

While GPs may also be defined over vector valued functions, in this work the independence of each output component is assumed, allowing for modeling with $c \geq 1$ independent GPs. For more details see Ch.2 of Rasmussen & Williams (2005), from which the notation and content of this section were inspired.

C Proof of Theorems and Multiclass Extension

C.1 Theorem 1: Relation to Exponential Geodesic Kernel

$$k(x, y) = \int\int \exp[-\lambda d_{\text{geo}}(m, m')] q(m|x, \rho) q(m'|y, \rho)\ \ \mathrm{d}m'\mathrm{d}m$$
$$s.t. \quad q(m|x, \rho) \propto \exp[-\rho ||x - m||_2^2] p(m)$$

Note that ρ controls how to weight manifold samples close to x, y. We take $\lim_{\rho \to \infty}$:

$$\lim_{\rho \to \infty} q(m|x, \rho) q(m'|y, \rho) = \begin{cases} 1 & x = m \text{ and } y = m' \\ 0 & \text{Otherwise} \end{cases}$$

Therefore the function within the integral of $k(x, y)$ evaluates to zero at all points except $x = m$ and $y = m'$. Since $x, y \in \mathcal{M}$ we can evaluate the integral:

$$k(x, y) = \exp[-\lambda d_{\text{geo}}(x, y)]$$

[2] assuming prior $m(x) = 0$

C.2 THEOREM 2: KERNEL SIMILARITY AND DECISION BOUNDARY COMPLEXITY

From definition 1, given any perturbation $\widetilde{\mathcal{P}}$ on $\mathcal{P}$, there must exist a compact subset $K_i \subset U_i$ s.t. $R|_{\mathcal{P}\setminus \text{int}(K_i)} = \text{id}|_{\mathcal{P}\setminus \text{int}(K_i)}$ and $R|_{\text{int}(K_i)} \neq \text{id}|_{\text{int}(K_i)}$. Furthermore there exists a linear homeomorphism between an open subset $\widetilde{U_i} \subseteq U_i$ with $\mathbb{R}^{d-1}$ which contains K_i.

We parametrize K_i using a smooth function $g : \mathcal{T} \to K_i$ s.t. $g(t) \in \partial K_i \ \forall t \in \partial \mathcal{T}$.

We further define $g_\epsilon(t) = g(t) + \epsilon\eta(t)$, for some perturbation $\epsilon \in \mathbb{R}$ and a smooth function $\eta : \mathcal{T} \to \mathbb{R}^{d-1}$ We also restrict η such that $\eta(t) = \mathbf{0} \ \forall t \in \partial\mathcal{T}$ and $\exists\ t_0 \in \mathcal{T}$ s.t. $\eta(t_0) \neq g(t_0)$. In other words, η is a smooth function where $g_\epsilon(t) = g(t)\ \forall \epsilon > 0, \forall t \in \partial\mathcal{T}$, but is not identical to g for all $t \in \mathcal{T}$. Using $g_\epsilon(t)$, we define the manifold $\mathcal{P}_\epsilon = \{g_\epsilon(t) : t \in \mathcal{T}\}$.

To complete the proof, we want to show that the kernel similarity between any two given points $x, y \in \mathbb{R}^d$ is lower when using the manifold $\mathcal{P}_\epsilon$ for $\epsilon > 0$ as opposed to the manifold $\mathcal{P}_0$. We therefore want to compare the two respective kernels $k_\epsilon(x, y)$ and $k_0(x, y)$. Note that in this proof we consider the local effects of $\mathcal{P}$ on the kernel similarity through $\mathcal{P}_0$ and $\mathcal{P}_\epsilon$ exclusively, ignoring the manifold $\mathcal{P} \setminus U_0$ Using Euler-Lagrange, we can calculate a lower bound for $d_{\text{geo}}(g_\epsilon(t)), g_\epsilon(t')))$. In particular, for any $t, t' \in \mathcal{T}$, $d_{\text{geo}}(g_\epsilon(t)), g_\epsilon(t'))) \geq d_{\text{geo}}(g_0(t), g_0(t'))$.

$$d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t')) \geq d_{\text{geo}}(g_0(t), g_0(t')) \tag{20}$$

$$\exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))] \leq \exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))] \tag{21}$$

$$\int_{\mathcal{T}}\int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))]\ \text{d}t\text{d}t' \leq \int_{\mathcal{T}}\int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))]\ \text{d}t\text{d}t' \tag{22}$$

Note that in Eq. 22 we are integrating over all possible values of t, t', therefore the inequality is tight iff $g_\epsilon(t) = g_0(t)\ \forall t \in \mathcal{T}$; i.e. $\epsilon = 0$ (see proof in C.2.1). The case of $\epsilon = 0$ is trivial; we instead assume $\epsilon > 0$, in which case we can establish the following strict inequality:

$$\int_{\mathcal{T}}\int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))]\ \text{d}t\text{d}t' < \int_{\mathcal{T}}\int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))]\ \text{d}t\text{d}t' \tag{23}$$

Define uniform random variables T, T' over the domain of g, i.e. $T, T' \sim \mathcal{U}_{\mathcal{T}}$. Then we have:

$$\mathbb{E}_{T,T'\sim\mathcal{U}_{[0,1]}}[\exp[-\lambda d_{\text{geo}}(g_\epsilon(T), g_\epsilon(T'))]] < \mathbb{E}_{T,T'\sim\mathcal{U}_{[0,1]}}[\exp[-\lambda d_{\text{geo}}(g_0(T), g_0(T'))]] \tag{24}$$

$$\mathbb{E}_{M,M'\sim p_\epsilon(M)}[\exp[-\lambda d_{\text{geo}}(M, M')]] < \mathbb{E}_{M,M'\sim p_0(M)}[\exp[-\lambda d_{\text{geo}}(M, M')]] \tag{25}$$

We define the random variable $M = g_\epsilon(T)$ with distribution $p_\epsilon(M)$. The distribution $p_\epsilon(M)$ represents the uniform distribution $\mathcal{U}_{\mathcal{T}}$ mapped to the manifold $\mathcal{P}_\epsilon$ using $g_\epsilon(T)$. The step from Eq. 24 to Eq. 25 uses a property of distribution transformations (Eq. 2.2.5 in Casella & Berger (2001)).

Next, compare either side of Eq. 25 to our kernel formulation shown below in Eq. 26. The kernel $k_\epsilon(x, y|\rho, \lambda)$ takes an expected value over $q_\epsilon(M|x, \rho)$ and $q_\epsilon(M'|y, \rho)$, which are equivalent to $p_\epsilon(M)$ and $p_\epsilon(M')$ weighted with respect to x, y, and a hyperparameter $\rho \geq 0$.

$$\begin{aligned} k_\epsilon(x, y|\rho, \lambda) = \mathbb{E}_{M\sim q_\epsilon(M|x,\rho), M'\sim q_\epsilon(M'|y,\rho)}[\exp[-\lambda d_{\text{geo}}(M, M')]] \\ s.t. \quad q_\epsilon(M|x, \rho) \propto \exp[-\rho||x - M||_2^2]p_\epsilon(M) \\ s.t. \quad q_\epsilon(M'|y, \rho) \propto \exp[-\rho||y - M'||_2^2]p_\epsilon(M') \end{aligned} \tag{26}$$

Note that when ρ is set to zero, $q(M|x,0) = p(M)$ and $q(M'|y,0) = p(M')$. Therefore Eq. 25 is equivalent to the inequality $k_\epsilon(x,y|0,\lambda) < k_0(x,y|0,\lambda)$.

We next want to prove that the inequality $k_\epsilon(x,y|\rho,\lambda) < k_0(x,y|\rho,\lambda)$ also holds for non-zero values of ρ. For convenience, define

$$f(\rho) = k_0(x,y|\rho,\lambda) - k_\epsilon(x,y|\rho,\lambda) \tag{27}$$

Under this definition, we want to prove there exists $\rho_0 > 0$ such that $f(\rho) > 0 \; \forall \rho < \rho_0$. From Eq. 25, we established that $f(0) > 0$. Assume that

$$\lim_{\rho \to 0} f(\rho) = c \tag{28}$$

It therefore follows that $c > 0$. In addition, note that $f(\rho)$ is continuous with respect to ρ (see proof in section C.2.3). Therefore for any $\epsilon > 0$ there exists $\delta > 0$ s.t. $\rho < \delta$ implies $|f(\rho) - c| < \epsilon$.

We choose $\epsilon = c$ and the define the corresponding δ to be ρ_0. Therefore:

$$\rho < \rho_0 \Rightarrow |f(\rho) - c| < c \tag{29}$$

$$\rho < \rho_0 \Rightarrow 0 < f(\rho) < 2c \tag{30}$$

Since this result holds for any i, it follows that the piecewise linear manifold $\mathcal{P}$ is a local minimum under any perturbation along a specific chart or combination of charts with respect to the kernel similarity $k(x,y) \; \forall x, y \in \mathbb{R}^d$.

C.2.1 PROOF: EQ. 23

We want to prove:

$$\int_{\mathcal{T}} \int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))] \; \mathrm{d}t\mathrm{d}t' = \int_{\mathcal{T}} \int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))] \; \mathrm{d}t\mathrm{d}t' \tag{31}$$
$$\Rightarrow g_\epsilon(t) = g_0(t) \quad \forall t \in \mathcal{T}$$

Consider the LHS of Eq. 31:

$$\int_{\mathcal{T}} \int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))] \; \mathrm{d}t\mathrm{d}t' = \int_{\mathcal{T}} \int_{\mathcal{T}} \exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))] \; \mathrm{d}t\mathrm{d}t' \tag{32}$$

$$\int_{\mathcal{T}} \int_{\mathcal{T}} \underbrace{\exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))] - \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))]}_{h(t,t')} \; \mathrm{d}t\mathrm{d}t' = 0 \tag{33}$$

Define $h(t,t')$ as the function inside the integrals in Eq. 33. From Eq. 21, $h(t,t') \geq 0 \; \forall t, t' \in \mathcal{T}$. Since h is continuous (see proof in C.2.2) and $\int_{\mathcal{T}} \int_{\mathcal{T}} h(t,t') \mathrm{d}t\mathrm{d}t' = 0$, it follows that $h(t,t') = 0$ $\forall t, t' \in \mathcal{T}$ (Ch.6 Rudin (1976)).

It therefore follows that:

$$\exp[-\lambda d_{\text{geo}}(g_0(t), g_0(t'))] = \exp[-\lambda d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))] \quad \forall t, t' \in \mathcal{T} \tag{34}$$

From the definition of $\eta(t)$ in $g_\epsilon(t) = g(t) + \epsilon\eta(t)$, there must exist $t \in \mathcal{T}$ s.t. $\eta(t) \neq 0$. Therefore ϵ must be zero for Eq. 34 to hold. It follows that $g_\epsilon(t) = g_0(t) \; \forall t \in \mathcal{T}$.

C.2.2 PROOF: CONTINUITY OF $h(t, t')$

We prove that $h(t, t')$ is continuous with respect to t, t'. First note that by definition, $g_\epsilon(t)$ is a continuous parametrization of the manifold $\mathcal{P}_\epsilon$. From Burago et al. (2001), it follows that for any two points $g_\epsilon(t), g_\epsilon(t') \in \mathcal{P}_\epsilon$, $d_{\text{geo}}(g_\epsilon(t), g_\epsilon(t'))$ is continuous. Since the exponential functional preserves continuity and the sum of continuous functions are also continuous, it follows that $h(t, t')$ is continuous.

C.2.3 PROOF: CONTINUITY OF $k(x, y)$ WITH RESPECT TO ρ

We prove that $k(x, y)$ is continuous with respect to ρ.

$$k(x, y) = \int \int \exp[-\lambda d_{\text{geo}}(m, m')]\ q(m|x, \rho)\ q(m'|y, \rho)\ \mathrm{d}m\mathrm{d}m' \tag{35}$$

$$= \frac{1}{Z_m(\rho) Z_{m'}(\rho)} \int \int \mathcal{A} \underbrace{\exp[-\rho(||x - m||_2^2 + ||y - m'||_2^2)]}_{Z(\rho)}\ \mathrm{d}m\mathrm{d}m' \tag{36}$$

$$s.t. \quad Z_m(\rho) = \int \exp[-\rho||x - m||_2^2] p(m)\ \mathrm{d}m$$

$$Z_{m'}(\rho) = \int \exp[-\rho||y - m'||_2^2] p(m')\ \mathrm{d}m'$$

$$\mathcal{A} = \exp[-\lambda d_{\text{geo}}(m, m')] p(m) p(m')$$

Define $h(\rho) = \rho\mathcal{B}$, where $\mathcal{B}$ is a constant. Consider $h(\rho) - h(\rho_0)$, where ρ_0 is a fixed positive constant:

$$|h(\rho) - h(\rho_0)| = |\rho\mathcal{B} - \rho_0\mathcal{B}| \tag{37}$$

$$= |(\rho - \rho_0)\mathcal{B}| < \delta|\mathcal{B}| \tag{38}$$

It follows that $\forall\ \epsilon > 0, \exists\ \delta = \frac{\epsilon}{|\mathcal{B}|} > 0$ such that $|\rho - \rho_0| < \delta \ \Rightarrow\ |h(\rho) - h(\rho_0)| < \epsilon$. Therefore h is continuous for all $\rho \in \mathbb{R}^+$.

We set $\mathcal{B}$ to be $||x - m||_2^2$, $||y - m'||_2^2$, and $||x - m||_2^2 + ||y - m'||_2^2$, which shows that $Z_m(\rho)$, $Z_{m'}(\rho)$, and $Z(\rho)$ are also continuous, respectively. It then follows that the entirety of Eq. 36 is continuous.

C.3 EXTENDING TO MULTICLASS CLASSIFIERS

In the multiclass case we define a black-box prediction model $F : \mathcal{X} \to \mathbb{R}^c$. We consider the one-vs-all DB for every class $y \in \mathcal{Y} = \{1, \ldots, c\}$, defined as $\mathcal{M}_y = \{x \in \mathbb{R}^d : F_y(x) = \max_{i \in \mathcal{Y}} F_i(x) = \max_{j \neq y \in \mathcal{Y}} F_j(x)\}$, where F_k indicates the model output for class k. We then apply the GPEC framework separately to each class using the respective DB. The uncertainty estimate of the GP model would be of dimension $d \times s$.

D IMPLEMENTATION DETAILS

D.1 ALGORITHM

The GPEC training algorithm is outlined in Alg. 1. GPEC is parametrized using a multi-output Gaussian Process Regression model using the explanations as labels. Once the explanations L, explanation uncertainty U, and WEG kernel matrix K are generated from Alg. 1, we can directly use these values to update the GP posterior and calculate the prediction variance for new test samples (Eq. 19).

D.2 ADVERSARIAL SAMPLE FILTERING MULTI-CLASS MODELS

We elect to sample from multi-class neural network decision boundaries by running binary search on train-point adversarial example pairs. Specifically, given a test-point $x_0 \in \mathbb{R}^d$ and model prediction $y = \text{argmax}_{k \in \mathcal{Y}} F(x_0)$, decision boundary points may be generated by the following procedure:

First, for each class $v \in \mathcal{Y}$ a set of M_v points is randomly sampled from the set of train points on which the model predicts class v:

$$\mathcal{X}_v \subseteq \{x : \text{argmax}_{k \in \mathcal{Y}} F(x) = v,\ x \in \mathcal{X}_{tr}\},\ |\mathcal{X}_v| = M_v \tag{39}$$

$\forall v \in \mathcal{Y}$. An untargeted adversarial attack using some l_p norm and radius ϵ is run on each point in $\mathcal{X}_y$, the set of points with the same class prediction as x_0. Each attack output $Attack_{un}(x, \epsilon) \in \mathbb{R}^d$ is paired with its corresponding input, resulting in the set

$$\mathcal{X}_{y'} = \{(x, Attack_{Un}(x, \epsilon)) : x \in \mathcal{X}_y\}, \tag{40}$$

where for an element $(a, b) \in \mathcal{X}_{y'}$ we have $\text{argmax}_{k \in \mathcal{Y}} F(a) = y, \text{argmax}_{k \in \mathcal{Y}} F(b) = v \neq y$, where v is an unspecified class.

Likewise a targeted adversarial attack, with target class y, is run on each point in each of the sets of points that are not predicted as class y. Each attack output $Attack_y(x, \epsilon) \in \mathbb{R}^d$ may be paired with its input x resulting in sets

$$\mathcal{X}_{v'} = \{(x, Attack_y(x, \epsilon)) : x \in \mathcal{X}_v\} \tag{41}$$

$\forall v \neq y \in \mathcal{Y}$. Here, for an element $(a, b) \in \mathcal{X}_{v'}$ we have $\text{argmax}_{k \in \mathcal{Y}} F(a) = v, \text{argmax}_{k \in \mathcal{Y}} F(b) = y$.

Thus, we have generated a diverse set of $\sum_{v \in \mathcal{Y}} M_v$ pairs of points that lie on opposite sides of the decision boundary for class y. The segment between any pair from a given set $\mathcal{X}_{v'}$ $v \neq y$ will necessarily contain a point on the class v v.s. class y decision boundary. Likewise, in the interest of further diversity, segments between any pair from the set $\mathcal{X}_{y'}$ will contain a point on the class v v.s. class y decision boundary, where $v \neq y \in \mathcal{Y}$ is unspecified. A binary search may be run on each pair to find the boundary point in the middle.

In practice the entire procedure may be amortized for each class, and ran for all classes as a single post-processing step immediately after training. This results in a dictionary of boundary points which may be efficiently queried on demand via the model predicted class of any given test point.

Each adversarial attack is attempted multiple times, once using each radius value ϵ in the list: $[0.0, 2e^{-4}, 5e^{-4}, 8e^{-4}, 1e^{-3}, 1e^{-3}, 1.5e^{-3}, 2e^{-3}, 3e^{-3}, 1e^{-2}, 1e^{-1}, 3e^{-1}, 5e^{-1}, 1.0]$. For a given input, the output of the successful attack with smallest ϵ is used. If no attack is successful at any radius, the input is discarded from further consideration.

In this work the Foolbox Rauber et al. (2017; 2020) implementation of the Projected Gradient Descent (PGD) Madry et al. (2018) attack with the l_∞ norm was used for both targeted and untargeted attacks. The M_c values used for the relevant datasets are indicated below in Appendix E.1.

E EXPERIMENT SETUP

E.1 DATASETS AND MODELS

Census. The UCI Census dataset consists of 32,561 samples from the 1994 census dataset. Each sample is a single person's response to the census questionaire. An XGBoost model is trained using the 12 features to predict whether the individual has income $\geq$ \$50k.

Online Shopper. The UCI Online Shoppers dataset consists of clickstream data from 12,330 web sessions. Each session is generated from a different individual and specifies whether a revenue-generating transaction takes place. There are 17 other features including device information, types of pages accessed during the session, and date information. An XGBoost model is trained to predict whether a purchase occurs.

German Credit. The German Credit dataset consists of 1,000 samples; each sample represents an individual who takes credit from a bank. The classification task is to predict whether an individual is

	Census	Online Shoppers	German Credit	MNIST	f-MNIST
GPEC-WEG	0.11	0.37	0.07	12.90	18.15
GPEC-RBF	0.00	0.00	0.02	8.95	7.41
CXPlain	0.05	0.06	0.04	9.76	18.18
BayesSHAP	140.40	54.56	4.86	42,467	42,361
BayesLIME	91.29	54.60	4.83	41,832	41,992

Table 2: Execution time comparison for estimating the uncertainty for all features for 100 samples (in seconds). For MNIST and f-MNIST datasets, results represent execution time for calculating uncertainty estimates with respect to all ten classes. For GPEC-WEG, GPEC-RBF, and CXPlain methods, the results show inference times.

considered a good or bad risk. Features include demographic information, credit history, and information about existing loans. Categorical features are converted using a one-hot encoding, resulting in 24 total features.

MNIST. The MNIST dataset (LeCun & Cortes, 2010) consists of 70k grayscale images of dimension 28x28. Each image has a single handwritten numeral, from 0-9. A fully connected network with layer sizes 784-700-400-200-100-10 and ReLU activation functions was trained and validated on on $50,000$ and $10,000$ image label pairs, respectively. Training lasted for 30 epochs with initial learning rate of 2 and a learning rate decay of $\gamma = 0.5$ when training loss is plateaued. During adversarial example generation we used $M_y = 500$ and $M_c = 50\ \forall c \neq y$.

Fashion MNIST The Fashion MNIST dataset (Xiao et al., 2017) contains 70k grayscale images of dimension 28x28. There are 10 classes, each indicating a different article of clothing. We train a MLP model with the same architecture used for the MNIST dataset, however we increase training to 100 epochs and increase the initial learning rate to 3. During adversarial example generation we used $M_y = 500$ and $M_c = 50\ \forall c \neq y$.

E.2 Competitor Implementation Details

BayesLIME and BayesSHAP. Slack et al. (2021) extend the methods LIME and KernelSHAP to use a Bayesian Framework. BayesLIME and BayesSHAP are fit using Bayesian linear regression models on perturbed outputs of the black-box model. The posterior distribution of the model weights are taken as the feature attributions instead of the frequentist estimate that characterizes LIME and KernelSHAP. We take the expected value of the posterior distribution as the point estimate for feature attributions, and the 95% credible interval as the estimate of uncertainty. To implement BayesLIME and BayesSHAP we use the public implementation[3]. We set the number of samples to 200, disable discretization for continuous variables, and calculate the explanations over all features. Otherwise, we use the default parameters for the implementation.

CXPlain. Schwab & Karlen (2019) introduces the explanation method CXPlain, which trains a surrogate explanation model based on a causal loss function. After training the surrogate model, the authors propose using a bootstrap resampling technique to estimate the variance of the predictions. In our experiments we implement the publicly available code[4]. We use the default parameters, which include using a 2-layer UNet model Ronneberger et al. (2015) for the image datasets and a 2-layer MLP model for the tabular datasets. We take a 95% confidence interval from the bootstrapped results as the estimate of uncertainty.

F Additional Results

F.1 Execution Time Comparison

In Table 2 we include an execution time comparison between the methods implemented in this paper. Results are averaged over 100 test samples. For MNIST and f-MNIST datasets, results evaluate the time to calculate uncertainty estimates with respect to all classes. All experiments were run on an internal cluster using AMD EPYC 7302 16-Core processors. We observe from the results that

[3]https://github.com/dylan-slack/Modeling-Uncertainty-Local-Explainability

[4]https://github.com/d909b/cxplain

the methods that amortization methods (GPEC-WEG, GPEC-RBF, CXPlain) are significantly faster than perturbation methods BayesLIME and BayesSHAP.

F.2 Toy Example: Evaluating GPEC on a Linear Classifier

In order to understand how the GPEC-WEG uncertainty estimate behaves for linear models, we use a toy example shown in Fig. F.2. In the top row, we visualize the GPEC-WEG uncertainty; we see that the uncertainty estimate is a small, constant value for the linear model (left) whereas uncertainty increases for the nonlinear model (right). Intuitively, GPEC derives its uncertainty estimate by evaluating the distribution of explanations with respect to the black-box model. Since it is generally infeasible to evaluate the space of all possible explanations, GPEC will typically estimate a "baseline" amount of uncertainty for every explanation, which depends on the sampling of the training distribution and is minimized for a linear model.

In the bottom two figures we visualize the magnitude of the gradient for the two respective models, which is a noiseless estimate of feature importance. We see that even using this deterministic feature importance method, the estimates can fail to be robust due to the nonlinearity, i.e. nearby samples within the same class can have very different explanations.

F.3 Visualizing effects of Explainer Uncertainty in GPEC Estimate

In section 5.4 we evaluate GPEC's ability to combine uncertainty from the black-box decision boundary and the uncertainty estimate from BayesSHAP and SSV explainers. In Figure 7 we extend this experiment to evaluate how well GPEC can capture the explainer uncertainty. We calculate the combined GPEC+explainer estimate using different numbers of approximation samples.

Both BayesSHAP and SSV depend on sampling to generate their explanations; having fewer samples increases the variance of their estimates. As we decrease the number of samples from 200 (Row A) to 5 (Row B) we would expect that the explainer uncertainty, and consequently the combined GPEC uncertainty, would increase. We see in Row Δ that the results follow our intuition; uncertainty increases for most of the plotted test points and uncertainty does not decrease for any points.

F.4 Additional Results for Uncertainty Visualization Experiment

Results for Y-Axis Feature In Figure 8 we visualize the estimated explanation uncertainty as a heatmap for a grid of explanations. The generated plots only visualize the uncertainty for the feature on the x-axis. Due to space constraints, we list the results for the y-axis feature in the appendix, in Figure 8. We can see that the results are in line with those from the x-axis figure.

Black-box model output For reference, in Figure 9 we plot the probability output of the XGBoost models used in the visualization experiment (Figure 4).

F.5 Sensitivity Analysis of WEG Kernel Parameters

The WEG kernel formulation uses two parameters, ρ and λ. The parameter ρ controls the weighting between each datapoint and the manifold samples. As ρ increases, the WEG kernel places more weight on manifold samples close in ℓ_2 distance to the given datapoint. The parameter λ acts as a bandwidth parameter for the exponential geodesic kernel. Increasing λ increases the effect of the geodesic distance along the manifold. Therefore decision boundaries with higher complexity will have an increased effect on the WEG kernel similarity. In Figures 10, 11, and 12 we plot heatmaps for various combinations of ρ and λ parameters to evaluate the change in the uncertainty estimate. The black line is the decision boundary and the red points are the samples used for training GPEC. Please note that the heatmap scales are not necessarily the same for each plot.

Algorithm 1 GPEC Model Training

Input : GPEC Training Samples $X \in \mathbb{R}^{M \times d}$. Explainer E. Hyperparameters P (# DB Samples), J (*optional*: # Samples for Functional Approximation Uncertainty estimate)
Output : Explanations $L \in \mathbb{R}^{M \times S}$, Explanation Uncertainty $U \in \mathbb{R}^{M \times S}$, WEG Kernel $K \in [0,1]^{M \times M}$

\\ Calculate Function Approximation Uncertainty
Initialize $L \in \mathbb{R}^{M \times S}$, $U \in \mathbb{R}^{M \times S}$
if *Explainer returns uncertainty estimate* **then**
  for *i = 1,2,...M* **do**
    $L_{i,:}, U_{i,:} \leftarrow E(X_{i,:})$ \\ Get explanations ($L_{i,:}$) and explanation uncertainty ($U_{i,:}$) from Explainer
  end
end
else if *Explainer is stochastic* **and** $J > 1$ **then**
  for *i = 1,2,...M* **do**
    Initialize $Q \in \mathbb{R}^{J \times S}$
    for *j = 1,2,...J* **do**
      $Q_{j,:} \leftarrow E(X_{i,:})$ \\ Draw stochastic explanations for same data sample
    end
    $L_{i,:} \leftarrow \frac{1}{J} \sum_{j=1}^{J} Q_{j,:}$
    $U_{i,:} \leftarrow \frac{1}{J} \sum_{j=1}^{J} (Q_{j,:} - L_{i,:})^2$ \\ Empirical uncertainty estimate (Eq. 2)
  end
end
else if *Explainer is deterministic* **then**
  for *i = 1,2,...M* **do**
    $L_{i,:} \leftarrow E(X_{i,:})$
  end
  $U \leftarrow \mathbf{0}$ \\ Set functional approximation uncertainty to zero
end

\\ Calculate EG Kernel Matrix
$B \leftarrow$ *Decision_Boundary_Sampler*(F, P) \\ Draw P DB samples of dimension d. $B \in \mathbb{R}^{P \times d}$
Initialize $G \in [0,1]^{P \times P}$
for *i = 1,2,...P* **do**
  for *j = 1,2,...P* **do**
    $G_{i,j} \leftarrow \exp(-\lambda d_{geo}(B_{i,:}, B_{j,:}))$ \\ Eq. 4
  end
end

\\ Calculate WEG Kernel Matrix
Initialize $W \in [0,1]^{M \times P}$ \\ Initialize weighting matrix
for *i = 1,2,...M* **do**
  for *j = 1,2,...P* **do**
    $W_{i,j} \leftarrow \exp(-\rho ||X_{i,:} - B_{j,:}||_2^2)$
  end
  $W_{i,:} \leftarrow \frac{W_{i,:}}{\sum_{j=1}^{P} W_{i,j}}$ \\ Normalize weighting distribution (Eq. 9)
end
$K \leftarrow WGW^{\intercal}$ \\ Apply weighting to EG Kernel Matrix

Return L, U, K

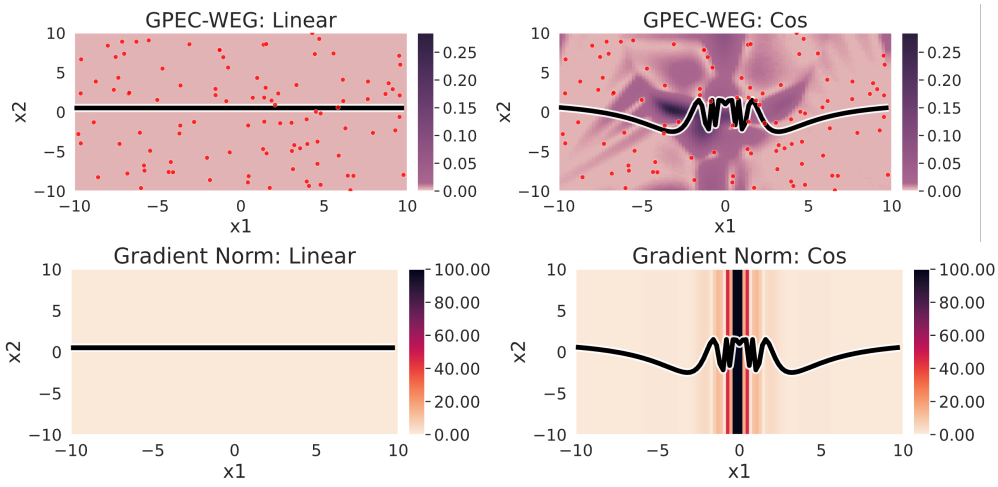

Figure 6: Comparison of a GPEC uncertainty estimates on linear and nonlinear toy models: $f_{\text{linear}}(x_1, x_2) = x_2$ and $f_{\cos}(x_1, x_2) = 2\cos(\frac{10}{x_1}) - x_2$. TOP: Uncertainty estimate from GPEC. Applying GPEC on a linear model results in a small, relatively constant variance for the test explanations. As the DB becomes more complex, as in $f_{\cos}$, the uncertainty estimate increases around the nonlinearities in the DB. BOTTOM: Gradient norm, which is an estimate of feature importance. The feature importance estimate becomes unstable (i.e. nearby samples of the same class have can have very different explanations) due to the nonlinearity of $f_{\cos}$.

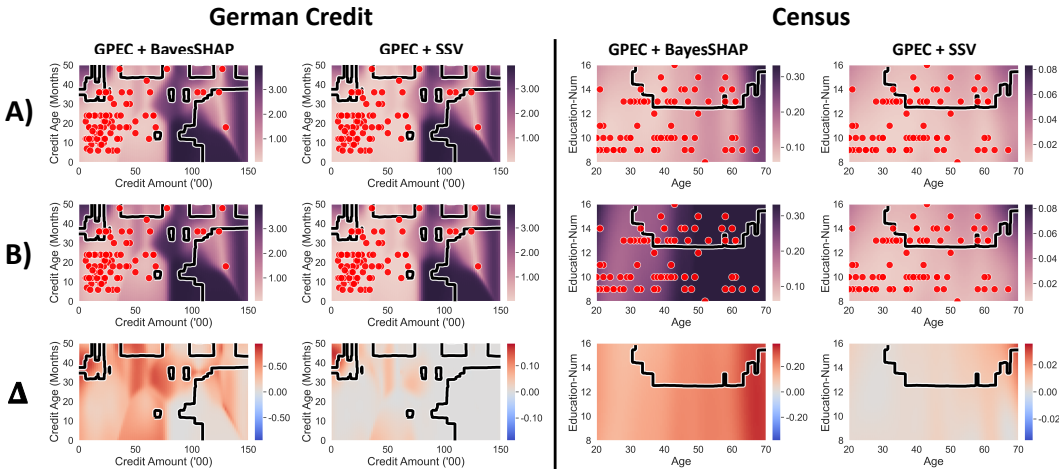

Figure 7: Comparison of the change in quantified uncertainty of explanations as we change the number of samples for BayesSHAP and SSV. Row (A) visualizes the combined uncertainty estimate using GPEC and either BayesSHAP or SSV, using 200 samples for approximating the BayesSHAP / SSV explanation. In Row (B) we decrease the number of samples to 5 and recalculate the estimated uncertainty. Row (Δ) represents the change in uncertainty estimate between (A) and (B). We see that the average uncertainty changes as we decrease the number of samples, which indicates that GPEC is able to capture the uncertainty arising from BayesSHAP / SSV approximation.

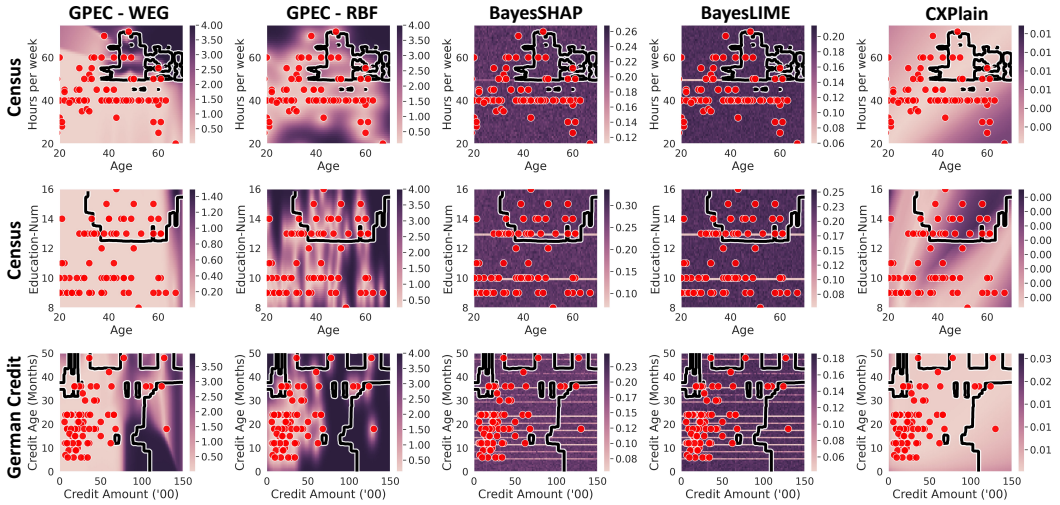

Figure 8: Complement to Figure 4. Visualization of estimated explanation uncertainty where the heatmap represents level of uncertainty for the feature on the **y-axis**.

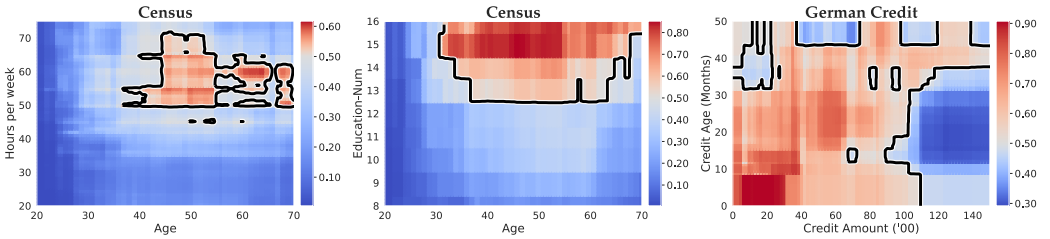

Figure 9: Output of binary classifiers used in the experiments. The color indicates the predicted probability of each point. Points with probability ≥ 0.5 are classified as class 1, and points with probability < 0.5 are classified as class 0. The decision boundary, which is the set of points $\{m : m \in \mathbb{R}^2, f(m) = 0.5\}$ where f is the classifier, is represented by the black line.

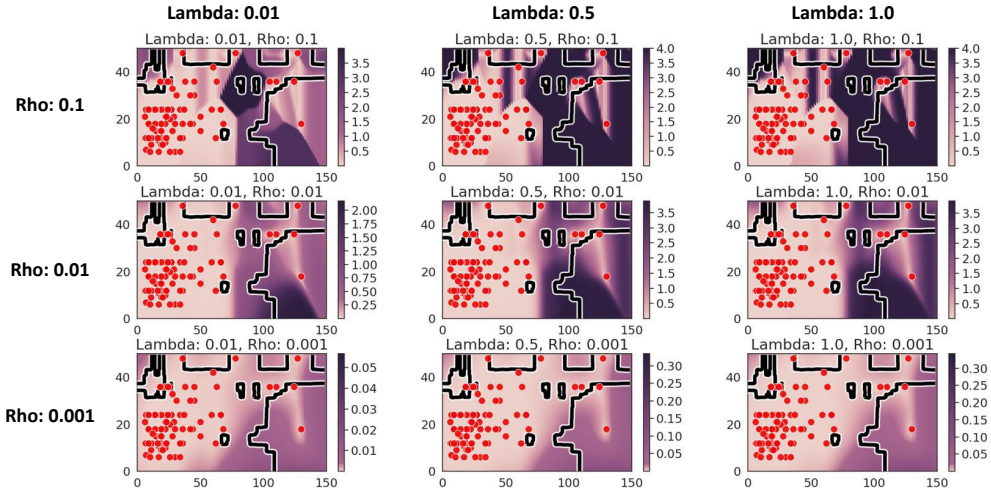

Figure 10: Hyperparameter sensitivity analysis for the German Credit Dataset. Heatmap of estimated uncertainty for the x-axis variable under different ρ and λ parameter choices.

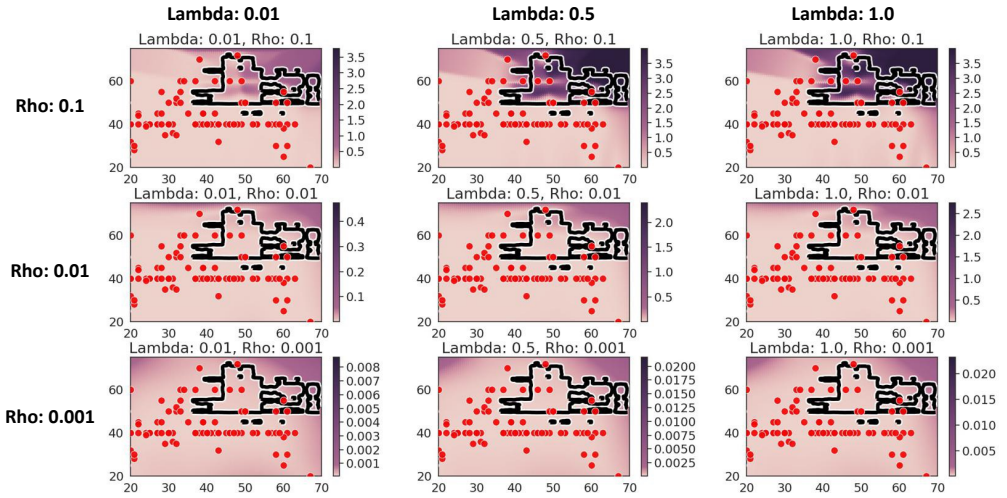

Figure 11: Hyperparameter sensitivity analysis for the Census Dataset. Heatmap of estimated uncertainty for the x-axis variable under different ρ and λ parameter choices.

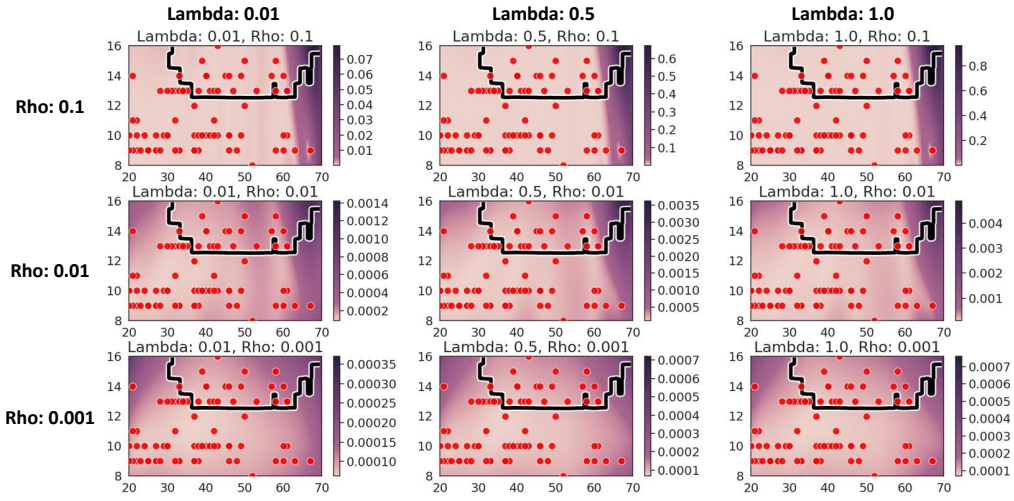

Figure 12: Hyperparameter sensitivity analysis for the Census Dataset. Heatmap of estimated uncertainty for the x-axis variable under different ρ and λ parameter choices.