BayLIME: Bayesian Local Interpretable Model-Agnostic Explanations

Xingyu Zhao[1,2]   **Wei Huang**[1]   **Xiaowei Huang**[1]   **Valentin Robu**[2,3,4]   **David Flynn**[2]

[1]Department of Computer Science, University of Liverpool, Liverpool, L69 3BX, U.K.
[2]School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, U.K.
[3]Centrum voor Wiskunde en Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands
[4]Delft University of Technology, Algorithmics Group, EEMCS, 2628 XE Delft, The Netherlands

Abstract

Given the pressing need for assuring algorithmic transparency, Explainable AI (XAI) has emerged as one of the key areas of AI research. In this paper, we develop a novel Bayesian extension to the LIME framework, one of the most widely used approaches in XAI – which we call BayLIME. Compared to LIME, BayLIME exploits prior knowledge and Bayesian reasoning to improve both the consistency in repeated explanations of a single prediction and the robustness to kernel settings. BayLIME also exhibits better explanation fidelity than the state-of-the-art (LIME, SHAP and Grad-CAM) by its ability to integrate prior knowledge from, e.g., a variety of other XAI techniques, as well as verification and validation (V&V) methods. We demonstrate the desirable properties of BayLIME through both theoretical analysis and extensive experiments.

1 INTRODUCTION

A key challenge to wide adoption of AI methods is their perceived lack of transparency. The black-box nature of many AI methods means they do not provide human users with direct explanations of their predictions. This led to growing interest in Explainable AI (XAI) – a research field that aims at improving the trust and transparency of AI. Explainable AI (XAI) methods can be classified by various criteria [Molnar, 2020, Chpt. 2.2], such as model-specific vs model-agnostic or local (instance-wise) vs global (entire model). Readers are referred to [Huang et al., 2020, Adadi and Berrada, 2018] for a comprehensive review.

In this work, we focus on the class of XAI methods using local surrogate models for explaining individual predictions – specifically, we develop a novel Bayesian extension to the most popular method in this category: Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro et al., 2016]. Despite its very considerable success in both research and practice, LIME has several weaknesses, the most significant of which are the lack of *consistency in repeated explanations of a single prediction* and *robustness to kernel settings*. Meanwhile, higher *explanation fidelity* is also expected in many settings. Arguably, these three properties are among the most desirable for an XAI method to have.

The inconsistency of LIME, where different explanations can be generated for the same prediction, has been identified as a critical issue by several prior works [Rehman Zafar and Mefraz Khan, 2019, Shankaranarayana and Runje, 2019]. This is caused by the randomness in generating perturbed samples (of the instance under explanation) that are used for the training of local surrogate models. Plainly stated, a smaller sample size leads to greater uncertainty of such randomness. This inconsistency has been argued to limit its usefulness in critical applications such as the medical domain, where consistency is highly required [Rehman Zafar and Mefraz Khan, 2019]. As an example, in Fig. 1, the pre-trained Convolutional Neural Network (CNN) InceptionV3 [Szegedy et al., 2016] predicts the instance of Fig. 1 (A) as "Bernese mountain dog" (top-1 label). To explain this prediction, we vary the size of perturbed samples (denoted as n) and record the time consumption in Fig. 1 (B). We see that the computational time is *linear* with respect to n. If, say, an application requires LIME to respond in 20s, we have to limit n to around 100 (in our case). Then, we may easily get three *inconsistent* explanations, as shown in Fig. 1 (C)-(E), in three repeated runs of LIME.

The other problem of LIME motivating this work relates to defining the right *locality* – the "neighbourhood" on which a local surrogate is trained to approximate the AI model [Laugel et al., 2018]. As noted by [Molnar, 2020, Chpt. 5.7] as a "very big and unsolved problem", how to choose a kernel setting in LIME to properly define a neighbourhood is challenging. Even worse, explanations can be easily turned around by changing kernel settings (cf. the example in [Molnar, 2020, Chpt. 5.7]). Thus, an enhancement to LIME so
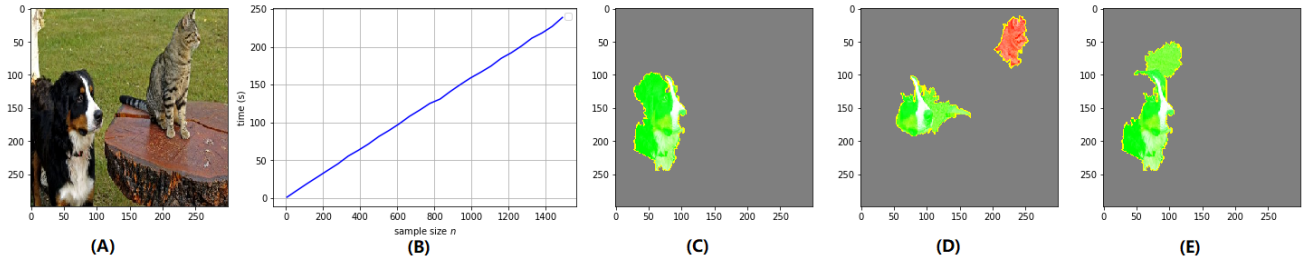

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

Figure 1: (A) The instance with prediction "Bernese mountain dog" under explanation; (B) LIME computational time as a function of perturbed sample size n; (C)-(E) Three repeated and noticeably *inconsistent* LIME explanations when $n = 100$, showing the top 4 features contributing toward and against the prediction (shaded green and red respectively).

that it is robust to the kernel settings is called for.

The goal of XAI is to provide good explanations. Despite the various definitions on the *goodness* of explanations, fidelity – how truthfully the explanation represents the underlying AI decision – is the most important criterion. Any new XAI method enhancing other properties should preserve the fidelity (if not improving), compared to the state-of-the-art.

In this paper, we propose a novel enhancement of LIME, named as BayLIME, by utilising a *Bayesian* local surrogate model, which we show analytically is a "Bayesian principled weighted sum" of the prior knowledge and the estimates based on new samples (as LIME). The weights consist of parameters with dedicated meanings that can be either automatically fitted from samples (Bayesian model selection) or elicited from application-specific prior knowledge (e.g., V&V methods). Our experiments show that BayLIME significantly improves over LIME with respect to a few objective measures, because the prior knowledge is independent from the causes of aforementioned inconsistency and unrobustness (thus benefits both properties) and also includes additional useful information that improves the fidelity.

The contributions of our work include:

(1) Developing a new XAI method, BayLIME, that provides a Bayesian principled mechanism for the *embedding* of prior knowledge;

(2) Introducing and designing quantitative metrics on both the consistency in repeated explanations and robustness to kernel settings that are generic to all feature-ranking based XAI techniques;

(3) Three typical ways of obtaining prior knowledge and eliciting prior parameters, including explanations on similar instances, other diverse XAI methods and V&V methods, to illustrate potential use cases of BayLIME;

(4) An extensive set of experiments, exhibiting the superior performance of BayLIME over LIME on aforementioned consistency and robustness, and higher fidelity over the state-of-the-art (LIME, SHAP, GradCAM);

(5) A prototype tool with tutorials/experimental-results.

2 CONSISTENCY IN REPEATED EXPLANATIONS

LIME is known to be unstable in terms of generating inconsistent explanations in repeated runs [Rehman Zafar and Mefraz Khan, 2019, Shankaranarayana and Runje, 2019]. This is due to the randomness in the perturbed samples used to train the local surrogate model. While randomness can be reduced by enlarging the sample size, it might be impractical in real-world applications, particularly those applications with strict efficiency constraints (e.g., low response time).

It has been noted by the original LIME paper [Ribeiro et al., 2016] that the time required to produce an explanation is *dominated* by the complexity of the black-box AI model. As shown in Fig. 1 (B), to improve LIME's efficiency, we cannot effectively tune other LIME arguments, rather the best option is to limit the *number of queries made* to the AI model (which, indeed, is very costly when the AI model is deep [He and Sun, 2015]) when generating labels for the n samples (step 3 of the LIME workflow in Appendix C). However, this solution leads to a severer inconsistency issue.

To compare between BayLIME and LIME on their inconsistency, we take Kendall's W [Kendall and Smith, 1939] to measure the agreement among raters (i.e. repeated explanations in our case), which ranges from 0 (no agreement) to 1 (complete agreement). However, Kendall's W only considers the discrete ranks of features, and cannot discriminate explanations with the same ranking of features but different importance vectors. To complement Kendall's W in such corner cases, we also introduce a new metric based on the *index of dispersion* (IoD) of each feature in repeated runs. The new metric weights the IoD of the rank of each feature by its importance, cf. Appendix B for a formal definition.

3 ROBUSTNESS TO KERNEL SETTINGS

Another notorious problem of LIME (or any approach with a notion of localisation) is how to meaningfully define a "neighbourhood" of the instance of interest (as required by the step 4 of LIME depicted in Appendix C). By default,

LIME uses an exponential smoothing kernel to define the neighbourhood with a given kernel width that determines how large the neighbourhood is. While intuitively a smaller kernel width means that an instance needs to be closer to influence the local model (vice versa), no effective way exists to find the best kernel settings. The best strategy for now is to try different kernel settings and see if the explanations make sense, which inevitably is subject to errors/bias. Moreover, an explanation may be easily turned around by changing the kernel settings [Molnar, 2020, Chpt. 5.7].

To compare between BayLIME and LIME, we explore the robustness to the *kernel width* parameter (denoted by l later) in LIME's default exponential kernel function. Other kernel setting parameters[1] can be studied in a similar way.

Definition 1 *To explain a given instance i, we denote $h_i : (0, +\infty) \rightarrow \mathbb{R}^m$ as the importance vector of m features taking l as the kernel width setting. For any pair of kernel width parameters l_1 and l_2, there exists a global Lipschitz value $\mathcal{L} \in \mathbb{R}$ such that $||h_i(l_1) - h_i(l_2)|| \leq \mathcal{L}||l_1 - l_2||$.*

Intuitively, this global Lipschitz value quantifies the robustness of an explainer to the change of kernel width. The computation of $\mathcal{L}$ can be seen as an optimisation problem:

$$\mathcal{L} = \underset{l_1, l_2 \in (0, +\infty)}{\arg\max} \frac{||h_i(l_1) - h_i(l_2)||}{||l_1 - l_2||} \tag{1}$$

which, unfortunately, is very challenging to solve analytically or to estimate numerically, due to the high complexity and non-linearity of the function h_i. Similar difficulties are also in, e.g. [Alvarez-Melis and Jaakkola, 2018], when studying the robustness to perturbed input instances.

To bypass the difficulty and still provide insights on the robustness defined earlier, we instead introduce a weaker empirical notion of the robustness to kernel settings:

Definition 2 *Assume L_1 and L_2 are both random variables of kernel width settings within $[l_{lo}, l_{up}]$. Then, R is the median[2] (denoted as $\mathbb{M}(\cdot)$) of the ratio between the perturbed distances of h_is and the pair (L_1, L_2):*

$$R = \mathbb{M}\left(\frac{||h_i(L_1) - h_i(L_2)||}{||L_1 - L_2||} \, \middle| \, l_{lo} \leq L_1 \leq l_{up}, l_{lo} \leq L_2 \leq l_{up} \right) \tag{2}$$

which represents the average robustness to the kernel settings when explaining the instance i.

The kernel width cannot be too large nor too small (cf. the general discussions on variance-bias trade-off in selecting hyper-parameters [Cawley and Talbot, 2010]). In practice,

---

[1] E.g., the distance parameter in a periodic kernel.

[2] Although other statistics may also suffice, we choose the median value to cope with the possible extreme outliers.

after a few trials, e.g., by cross-validation, it is not hard to figure out empirically a bound $[l_{lo}, l_{up}]$ as the range of all possible kernel settings in which we may perturb the kernel width to compute the robustness. In Eq. (2), we focus on the median value which is a much easier quantity to estimate than the optimised solution in Eq. (1), yet it provides insights on the general robustness.

4 EXPLANATION FIDELITY

Among many evaluation criteria on the goodness of explanations, fidelity – how truthfully the explanation represents the unknown behaviour of the underlying AI model – is of great importance. In other words, we want the XAI method to explain the *true cause* of the underlying model's prediction [Petsiuk et al., 2018, Sun et al., 2020]. In this paper, we adopt the *actual causality* as an indicator for the explanation fidelity in the following two different ways.

First, we reuse the two causal metrics proposed in [Petsiuk et al., 2018] – *deletion* and *insertion*. The intuition behind deletion is that the removal of the "cause" (important features) will force the underlying AI model to change its decision. To be exact, starting with a complete instance and then gradually removing top-important features (according to the ranked list obtained from XAI methods), the deletion metric measures the decrease in the probability of the predicted label. A sharp drop and thus a low Area Under the probability Curve (AUC), as a function of the fraction of removed features, suggests a good explanation that captures the real causality. Similarly, but in a complementary way, the insertion metric measures the increase in the probability as more and more important features are introduced, with higher AUC indicating a better explanation. The two metrics not only alleviate the need for large-scale human annotation effort, but also are better at assessing causal explanations by avoiding human bias [Petsiuk et al., 2018].

Second, we evaluate explanation fidelity through neural backdoors, inspired by [Lin et al., 2020, Sun et al., 2020]. The major difficulty of evaluating fidelity is the lack of ground truth behaviour of the underlying AI model. As the most important features that cause deliberate misclassification, backdoor triggers provide such ground truth and therefore should be highlighted by a good XAI method. Both [Lin et al., 2020] and [Sun et al., 2020] use the metric Intersection over Union (IoU) to measure the success in highlighting the trigger. Given a bounding box around the true trigger area B_T and the highlighted area by XAI methods B'_T, the IoU is their overlapped area divided by the area of their union, i.e., $(B_T \cap B'_T)/(B_T \cup B'_T)$. IoU is an established metric originally designed for object detection that ranges from 0 to 1, a higher IoU is better. It only considers the overlapping of the highlighted area and the ground truth, ignoring how geometrically closed they are when there is no overlapping. To complement this, we introduce a secondary

metric in Appendix A.2 to further confirm our conclusions.

5 A BAYESIAN RETROFIT OF LIME

Denote the input set with n samples as $\boldsymbol{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_n}\}$ where $\boldsymbol{x_i}$ is the instance i with m features (i.e. $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n \times m}$), and let the corresponding n target values be a column vector $\boldsymbol{y} = [y_1, ..., y_n]^T$. In Bayesian linear regression, a response y_i is assumed to be Gaussian distributed around $\boldsymbol{x_i}\boldsymbol{\beta}$:

$$Pr(y_i \mid \boldsymbol{\beta}, \boldsymbol{x_i}, \alpha) = \mathcal{N}(y_i \mid \boldsymbol{x_i}\boldsymbol{\beta}, \alpha^{-1}) \quad (3)$$

where $\boldsymbol{\beta}$ is the coefficient vector of m features, and α is the precision parameter (reciprocal of the variance) representing noise in this linear assumption. Then, we may write down the likelihood function:

$$Pr(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X}, \alpha) = \prod_{i=1}^{n} \mathcal{N}(y_i \mid \boldsymbol{x_i}\boldsymbol{\beta}, \alpha^{-1}). \quad (4)$$

For computational convenience, we choose a conjugate prior distribution for $\boldsymbol{\beta}$ – another (multivariate) Gaussian:

$$Pr(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}) = \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}) \quad (5)$$

where $\boldsymbol{\mu_0}$ and $\boldsymbol{S_0}$ are the prior mean vector and covariance matrix, respectively. Then, thanks to the conjugacy, the posterior $\boldsymbol{\beta}$ is also a Gaussian:

$$\begin{aligned} Pr(\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X}, \alpha, \boldsymbol{\mu_0}, \boldsymbol{S_0}) &\propto Pr(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{X}, \alpha) Pr(\boldsymbol{\beta} \mid \boldsymbol{\mu_0}, \boldsymbol{S_0}) \\ &= \mathcal{N}(\boldsymbol{\beta} \mid \boldsymbol{\mu_n}, \boldsymbol{S_n}) \end{aligned} \quad (6)$$

with the mean vector $\boldsymbol{\mu_n}$ and covariance matrix $\boldsymbol{S_n}$:

$$\boldsymbol{\mu_n} = \boldsymbol{S_n}(\boldsymbol{S_0}^{-1}\boldsymbol{\mu_0} + \alpha\boldsymbol{X}^T\boldsymbol{y}), \ \boldsymbol{S_n}^{-1} = \boldsymbol{S_0}^{-1} + \alpha\boldsymbol{X}^T\boldsymbol{X} \quad (7)$$

For simplicity and without loss of generality, we assume the Gaussian prior is governed by a single precision parameter λ, i.e. $\boldsymbol{S_0} = \lambda^{-1}\boldsymbol{I}_m$ where $\boldsymbol{I}_m$ is a $m \times m$ identity matrix.

5.1 EMBEDDING PRIOR KNOWLEDGE IN LIME

The local surrogate model is trained on *weighted* samples perturbed around the instance of interest, with weights representing their respective proximity to the instance. Therefore, the posterior estimate on $\boldsymbol{\beta}$ is now as follows.

$$\boldsymbol{\mu_n} = \boldsymbol{S_n}(\lambda\boldsymbol{I}_m\boldsymbol{\mu_0} + \alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}), \ \boldsymbol{S_n}^{-1} = \lambda\boldsymbol{I}_m + \alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} \quad (8)$$

where the weights $\boldsymbol{W} = diag(w_1, \ldots, w_n)$ is a diagonal matrix calculated by a kernel function according to the new samples' proximity to the original instance. The first equation of (8) can be rewritten as:

$$\begin{aligned} \boldsymbol{\mu_n} = &(\lambda\boldsymbol{I}_m + \alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\lambda\boldsymbol{I}_m\boldsymbol{\mu_0} \\ &+ (\lambda\boldsymbol{I}_m + \alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}\boldsymbol{\beta}_{MLE} \end{aligned} \quad (9)$$

where $\boldsymbol{\beta}_{MLE} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}$ is the Maximum Likelihood Estimates (MLE) for the linear regression model [Bishop, 2006] on the weighted samples.

The Eq. (9) is essentially a *weighted* sum of $\boldsymbol{\mu_0}$ and $\boldsymbol{\beta}_{MLE}$ – A *Bayesian combination* of prior knowledge and the new observations. The weights are proportional to: (i) $\lambda\boldsymbol{I}_m$, the "pseudo-count" of prior sample size based on which we form our prior estimates $\boldsymbol{\mu_0}$; (ii) $\alpha\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$, the "accurate-actual-count" of observation sample size, i.e. the actual observation of the n new samples $\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$ scaled by the precision α.

To see the above insight clearer, we present the special case of a single feature instance ($m = 1$) with a simplified kernel function that returns a constant weight w_c (i.e. $w_i = w_c, \forall i = 1, \ldots, n$), then μ_n ($\boldsymbol{\mu_n}$ with 1 feature) becomes:

$$\frac{\lambda}{\lambda + \alpha w_c \sum_{i=1}^{n} x_i^2}\mu_0 + \frac{\alpha w_c \sum_{i=1}^{n} x_i^2}{\lambda + \alpha w_c \sum_{i=1}^{n} x_i^2}\beta_{MLE} \quad (10)$$

where $\beta_{MLE} = \frac{\alpha w_c \sum_{i=1}^{n} x_i y_i}{\alpha w_c \sum_{i=1}^{n} x_i^2} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$. Now, denote X as the r.v. representing x_i, we know:

$$\sum_{i=1}^{n} x_i^2 \approx n\mathbb{E}(X^2) = n\left(Var(X) + \mathbb{E}(X)^2\right). \quad (11)$$

As implemented by LIME, x_is are n random samples from some distribution, depending on the type of the feature, e.g.: (i) For superpixels of images, $x_i \in \{0, 1\}$ are random samples from an uniform 0-1 two-point distribution; (ii) For numerical features of a tabular dataset, x_i are random samples from a standard Gaussian $\mathcal{N}(0, 1)$. Thus, say, for numerical features from tabular data, $\sum_{i=1}^{n} x_i^2 \approx n(1 + 0^2) = n$ via Eq. (11). Then, Eq. (10) can be simplified as:

$$\frac{\lambda}{\lambda + \alpha w_c n}\mu_0 + \frac{\alpha w_c n}{\lambda + \alpha w_c n}\beta_{MLE} \quad (12)$$

which neatly shows how the prior knowledge is embedded in our BayLIME in an intuitive story – "based on λ data-points prior to our new experiment, we form our prior estimate of μ_0. Now, in the experiments, we collect n samples. After considering the precision (α) and weights (w_c) of the new samples, we form a MLE estimate β_{MLE}. Then, we combine the two estimates – μ_0 and β_{MLE} – according to their proportions of the *effective* samples size used, i.e., λ and $\alpha w_c n$, respectively. Finally, the confidence in our new posterior estimate is captured by all effective samples used, i.e. $\lambda + \alpha w_c n$ (the posterior precision)". Now it is easy to see via Eq. (12) (also holds for Eq. (9)):

Remark 1 *Smaller λ means lower confidence in the prior knowledge, thus the posteriors are mostly dominated by the new observations, and vice versa. That is, in extreme cases: 1) When $\lambda \simeq 0$ the result (9) (and its simplified case (12)) reduces to MLE, i.e. "let the data speak for themselves". 2) On the other hand, if $n \simeq 0$ (or equivalently $\alpha \simeq 0$, $w_c \simeq 0$), then the β_{MLE} estimate from the new data is negligible and the prior knowledge dominates the posteriors.*

This kind of intuitive decomposition is generic to Bayesian inference in the linearly updated canonical-exponential families [Bernardo and Smith, 1994], which is found to be useful in applications that exploit prior knowledge, e.g., [Filieri et al., 2012, Zhao et al., 2019, Walter and Augustin, 2009, Zhao et al., 2020].

Remark 2 *Via the "weighted sum" Eq.* (9) *(and its simplified case* (12)*), the general reasons why integrating prior knowledge improve the three aforementioned properties are:*
1) $\boldsymbol{\beta}_{MLE}$ *is a function of the* n *randomly perturbed samples that causes inconsistency, while* $\boldsymbol{\mu_0}$ *is independent from the cause. A weighted sum of both may "average out" the randomness, thus improving the consistency.*
2) $\boldsymbol{\beta}_{MLE}$ *is a function of* $\boldsymbol{W}$ *that depends on the choice of kernel settings, while* $\boldsymbol{\mu_0}$ *is independent from kernel settings. A weighted sum of both may "average out" the effect from kernels, thus improving the robustness.*
3) $\boldsymbol{\mu_0}$ *normally contains added diverse information to* $\boldsymbol{\beta}_{MLE}$ *(black-box queries) that benefits the explanation fidelity.*

5.2 THE BAYLIME FRAMEWORK

BayLIME uses Bayesian linear regressors as local surrogate models, and implements three options as shown in Table 1.

Table 1: The BayLIME framework with three options

BayLIME Opt. \ prior para.	$\boldsymbol{\mu_0}$	λ	α
non-informative priors	zero vector	fitted	fitted
partial informative priors	known	known	fitted
full informative priors	known	known	known

BayLIME with non-informative priors. When no prior knowledge is available, we assume a zero mean vector for $\boldsymbol{\mu_0}$ and do *Bayesian model selection* [Bishop, 2006, Chpt. 3.4] for λ and α. Specifically, BayLIME reuses the established algorithm to fit α and λ from data [MacKay, 1992, Tipping, 2001]. The solutions for λ and α are *implicit*, since they are obtained by starting with initial values and then iterating over some interval equations until convergence.

BayLIME with partial informative priors. We assume a known complete prior distribution of the feature coefficients with mean vector $\boldsymbol{\mu_0}$ and precision λ. We call this option in BayLIME "with partial priors" in the sense that we still do not know the parameter α, for which we fit from data. Similarly as before, we modify the Bayesian model selection algorithm by iterating α (but with fixed λ in this case) to maximise the log marginal likelihood until convergence.

BayLIME with full informative priors. By assuming an ideal scenario in which we have full prior knowledge of all the $\boldsymbol{\mu_0}$, λ and α parameters, BayLIME may directly implement the closed-form results of Eq. (8).

5.3 WAYS OF OBTAINING PRIOR KNOWLEDGE

How to elicit accurate priors for BayLIME varies case by case (depending on the application-specific context), and is indeed challenging. However, we believe such challenge is neither unique to BayLIME, but generic to any approach with a Bayesian flavour, nor clueless in practice. That said, we implemented 3 ways of getting priors in our experiments (but not limited to, cf. Sec. 8 for more discussions):

a) Validation and Verification (V&V) methods that directly analyse the behaviour of the underlying AI model may indicate the importance of certain features, yielding priors required by BayLIME. E.g., when explaining a prediction made by an infected model, (imperfect) detection tools may provide prior knowledge on possible backdoor triggers. We implemented this scenario by using the results of the popular NeuralCleanse [Wang et al., 2019] as the priors.

b) There are emerging XAI techniques (cf. [Huang et al., 2020, Molnar, 2020] for a review). Explanations by some other *diverse* XAI explainers based on fundamentally different theories to LIME (e.g., gradient-based vs perturbation-based, global vs local) may provide useful prior knowledge. Presumably, the drawbacks of individual XAI methods can be mitigated by the "hybrid" solution provided by BayLIME. We implemented and confirmed this conjecture by using GradCAM [Selvaraju et al., 2017] results as the priors.

c) We also use explanations of *a set of similar instances* (to the instance under explanation) to form our prior knowledge. Although it is still not rigorous enough (e.g., how to decide what is a "similar" instance), we believe this way of obtaining priors serves as a first illustrative example.

6 EVALUATION

We evaluate our BayLIME by performing extensive experiments to address the following research questions.

RQ1: How effectively BayLIME improves the consistency of LIME in different scenarios of available prior knowledge?

RQ2: How effectively BayLIME improves the robustness to kernel settings of LIME in different scenarios of available prior knowledge?

RQ3: Will BayLIME improve the explanation fidelity compared to some state-of-the-art XAI methods in different scenarios of available prior knowledge?

Evaluation methods. LIME is one of the few XAI techniques that work for all tabular data, text and images [Molnar, 2020]. Our experiments were also conducted on diverse types of datasets including the Boston house-price dataset, the breast cancer Wisconsin dataset (with both numerical and categorical features), and a set of CNNs pretrained on the ImageNet and GTSRB [Stallkamp et al., 2012]. While
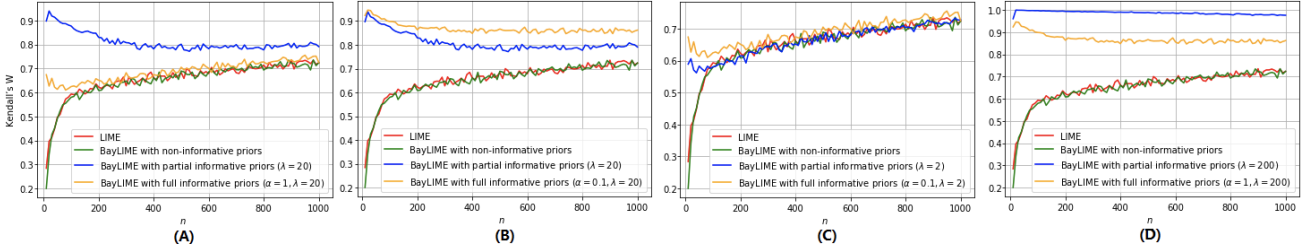
Figure 2: Kendall's W in $k = 200$ repeated explanations by LIME/BayLIME on random Boston house-price instances. Each set shows an illustrative combination of α and λ. Same patterns are observed on images, cf. Appendix B for more.

only typical results are presented in the main paper, cf. the appendices and our BayLIME project website[3] for more.

For RQ1, in addition to LIME, we select a set of BayLIME explainers with different options and prior parameters. For each explainer, we iterate the explanation of the given instance k times, and quantify the inconsistency according to Kendall's W. We treat the sample size n as a variable to confirm our earlier observation that inconsistency is an even severer problem when n has to be limited for efficiency.

To answer RQ2, we firstly define an interval $[l_{lo}, l_{up}]$ as the empirical bounds of all possible kernel width settings for the given application. We randomly sample from that interval 5000 pairs of kernel width parameters. Then, for each pair, we can calculate the "distance" of the two explanations[4]. Finally, we obtain a sample set of ratios between the two distances of explanations and the kernel width pair, on which statistics provides insights on the general robustness, e.g., the median value as used by Eq. (2).

The prior knowledge used in RQ1 and RQ2 is obtained from previous LIME explanations on a set of *similar* instances. Specifically, the average importance of each feature in that set collectively forms our prior mean vector $\boldsymbol{\mu_0}$, and the number of similar instances implies λ, leaving the α either as unknown (then fitted from data) or assigned empirically (based on previous fitting of α on similar instances).

For RQ3, we implemented 2 scenarios in which the prior knowledge is obtained from other diverse XAI and V&V methods respectively:

RQ3.a) We first use the gradient-based XAI approach Grad-CAM to obtain "grey-box" information of the underlying CNN as the prior, and then combines such extra knowledge with the "black-box" evidence as utilised by LIME. We compare the explanation fidelity of BayLIME with LIME, GradCAM and SHAP, via the deletion and insertion metrics.

RQ3.b) In this scenario, we aim to explain the behaviour of infected CNNs (BadNet [Gu et al., 2019] and TrojanAttack

[Liu et al., 2018] both trained on GTSRB with backdoors). The backdoor detector NeuralCleanse is used to *approximate* the location of triggers first. Then, BayLIME considers such knowledge as priors to provide a better explanation on the attacking behaviour, compared to NeuralCleanse and LIME applied solely. For the two example scenarios, we choose the relatively more practical option – BayLIME with partial priors, i.e., α is fitted, and the readers are referred to Appendix A for the formulae with intuitive rationales on eliciting $\boldsymbol{\mu_0}$ and λ.

Results and Discussion. RQ1. The red curves in Fig. 2 present the Kendall's W measurements as a function of the perturbed sample size n. Although it increases quickly, we observe very low consistency when n is relatively small (e.g., $n < 200$). These results support our earlier conjecture on the inconsistency issue of LIME, especially when n has to be limited by an upper-bound due to efficiency considerations. Non-informative BayLIME is indistinguishable from LIME, since both of them are *only* exploiting the information from the n samples that generated randomly – the presence of randomness means that the results of sampling cannot be duplicated if the process were repeated. Naturally, the more sparse the samples are, the greater randomness presents in the dataset. Thus, LIME and non-informative BayLIME show *monotonic trends* as n increases.

By contrast, BayLIME with partial/full informative priors can "average out" the sampling noise in the new generated data by combining the information from the priors, in a Bayesian principled way of Eq. (9). To closely inspect how effectively BayLIME with different priors affects the consistency, we first need the auxiliary of the factor λ/α. It is a known result that λ/α can be treated as a *regularization coefficient* in Bayesian linear regressors [Bishop, 2006, Chpt. 3.3], meaning a larger λ/α penalises more on the training data (to control over-fitting). Indeed, this aligns well with our Remark 1: (i) when $\alpha \simeq 0$, the factor $\lambda/\alpha \to +\infty$ meaning a huge penalty on the data, thus the prior knowledge effectively dominates the posteriors; (ii) when $\lambda \simeq 0$, $\lambda/\alpha \to 0$ meaning no penalty on the data, thus the posteriors is dominated by the new observations.

Both the plots of BayLIME with full informative priors (yellow curves) in Fig. 2 (A) and (C) have a regularization

---

[3] https://github.com/x-y-zhao/BayLime

[4] In RQ2, we fix the perturbed sample size to a sufficiently large number, e.g., $n = 1000$ (as indicated by the results of RQ1), to minimise the impact of inconsistency.

factor $\lambda/\alpha = 20$, and are basically identical. This is because, once $\lambda/\alpha = 20$ is fixed, the "proportion" of contributions to the posteriors by the priors and the new data is fixed. In other words, given n samples, the ability of "averaging out" sampling noise by the priors is fixed. When λ/α increases to 200, as shown by the yellow curves in Fig. 2 (B) and (D), such ability of averaging out sampling noise is even stronger, which explains why Kendall's W measurements in this case are higher than the case of $\lambda/\alpha = 20$. For BayLIME with partial informative priors (blue curves in Fig. 2), we observe that smaller λ results in worse consistency, e.g., Fig. 2 (C) vs (D). Again, Remark 1 applies here – smaller λ implies less contributions from the priors to the posteriors, meaning with less ability to average out the randomness in the samples.

Starting from a non-zero small number, as n increases, we can see BayLIME with partial/full informative priors may exhibit an uni-modal pattern, e.g., the Fig. 2 (C)[5] with a minimum point. This represents a tension between the *perfectly consistent prior knowledge* (does not change at all in repeated explanations) and *quite consistent MLE based on large samples*. There must be a "balance-point" in-between that compromises both ends of the tension, yielding a minimised consistency. Finally, when $n \to +\infty$, it is trivial to see (e.g., by taking the limit of Eq. (12) as a function of n), all plots will eventually converge (to the measurement based on MLE using infinitely large samples).

RQ2. Fig. 3 are box-and-whisker charts providing insights on the general robustness of eight AI explainers to kernel width settings, in which the median values defined by Eq. (2) are marked, as usual, by horizontal bars inside the boxes.
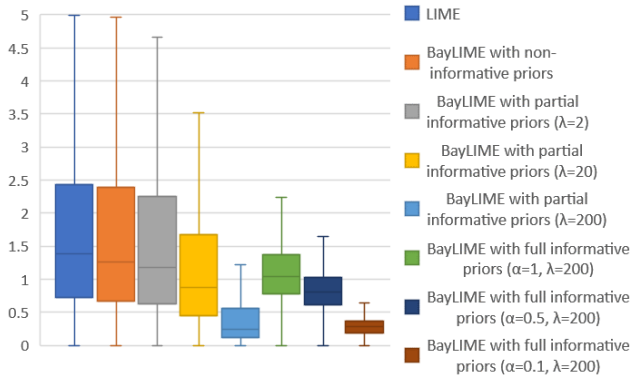

Figure 3: The general (un-)robustness of eight AI explainers to kernel settings (box-and-whisker plots without outliers).

Again, LIME and BayLIME with non-informative priors exhibit similar robustness, since there is no prior knowledge being considered, rather the data solely determines the explanations of both. In stark contrast, when either partial or full prior knowledge is taken into account, we observe an obvious improvement on the robustness to kernel settings.

---

[5]Other plots may not show the pattern clearly due to the range/scale of the axes and inevitable noise in the experiments.
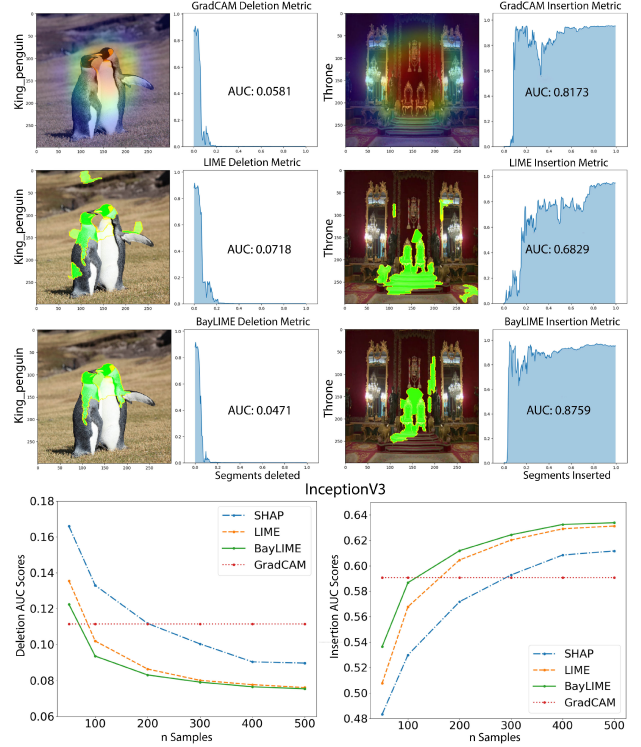

Figure 4: Two sets of examples ($n = 300$), and the average AUC (based on 1000 images per value of n) of the deletion (smaller is better) and insertion (bigger is better) metrics of BayLIME comparing with other XAI methods.

The regularisation factor λ/α and Remark 1 are still handy here in the discussions on how varying the λ and α affects the robustness – it all boils down to how much contribution from the priors (that is independent from kernel setting), compared with the contribution from the new data (that is sensitive to kernel settings), to the posteriors.

RQ3.a. GradCAM provides fundamentally different grey-box information as priors to the balck-box LIME. BayLIME, as expected, performs better than both GradCAM (the prior) and LIME (new observations), thanks to its unique advantage of incorporating diverse information. Fig. 4 first shows two sets of such examples via the AUC scores of the deletion and insertion metrics, respectively. Then statistics on the average scores (varying n) are shown in the last row of Fig. 4. We observe: (i) BayLIME performs better than SHAP and LIME, while there is a converging trend when n increases. This aligns with the second point in Remark 1. (ii) GradCAM is not a function of n (thus showing horizontal lines) and only performs better in the corner cases when n is extremely small (even smaller than the number of features). We conclude that, compared to the other 3 XAI methods, BayLIME has better fidelity in the middle and most practical range of n (e.g., 100∼400 in our case). For more examples/statistics on other CNNs cf. Appendix A.1

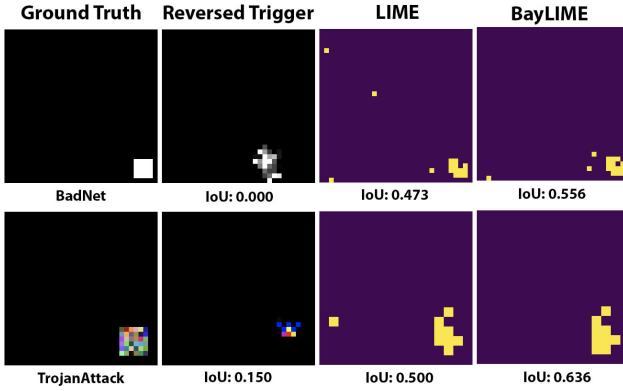**RQ3.b.** The V&V tool NeuralCleanse yields reversed trig-

Figure 5: Two examples and their IoU measurements of BayLIME using V&V results as priors (reversed triggers by NeuralCleanse), compared to the results of NeuralCleanse and LIME applied individually. LIME and BayLIME use the top-few features (where each feature is a fixed size square superpixel, cf. Appendix A.2 for details) such that the total number of pixels is the same as the ground truth. NB: We omit the background image to highlight the triggers only.

gers as the approximation of backdoor, which are far from perfect. E.g., in the BadNet example of Fig. 5 (1st row), the reversed trigger completely missed the ground truth trigger (thus IoU = 0), despite being very closed to. Even directly apply LIME on an attacked image may provide a better IoU than NeuralCleanse. Nevertheless, BayLIME performs the best after considering both the reversed triggers and a surrogate model as LIME (trained on the same number of n samples). As shown in Table 2, statistics on averaging the IoU of 500 random images confirm the above observation (NeuralCleanse is independent from individual images).

Table 2: Statistics on the average IoU by 3 methods.

Model	NeuralCleanse	LIME	BayLIME
BadNet	0.000	0.385	**0.406**
TrojanAttack	0.150	0.599	**0.637**

7 RELATED WORK

Several prior works aim at improving aspects of LIME: [Shi et al., 2020] modify the perturbed sampling method of LIME to cope with the correlation between features. RL-LIM [Yoon et al., 2019] employs reinforcement learning to select a subset of perturbed samples to train the local surrogate model. MAPLE [Plumb et al., 2018] uses local linear regressors along with random forests that determines weights of perturbed instances to optimise the local linear model. Both DLIME [Rehman Zafar and Mefraz Khan, 2019] and ALIME [Shankaranarayana and Runje, 2019] share the similar concern with us on the inconsistency of repeated explanations. DLIME replaces the random perturb-

ation with deterministic methods and ALIME employs an auto-encoder as a better weighting function for the local surrogate model. In comparison to these, BayLIME seeks inherently different solutions in a Bayesian way which has the unique advantage of embedding prior knowledge (thus direct experimental comparisons between them are not sensible – prior knowledge plays an important role in BayLIME that cannot be represented by other two methods). Moreover, BayLIME can deal with strict efficiency constraints (i.e., with small perturbed sample size n) and may improve explanation fidelity as well, while DLIME and ALIME cannot.

To the best of our knowledge, the only model-agnostic XAI techniques with a Bayesian flavour is in [Guo et al., 2018]. It derives generalised insights for an AI model through a *global* approximation (by a Bayesian non-parametric regression mixture model with multiple elastic nets). It can only fit conjugate priors empirically from data, and normally requires a large sample size for the inference on the large number of model parameters. By contrast, BayLIME is the first to exploit priors for better consistency, robustness to kernel settings and explanation fidelity.

8 DISCUSSION

Prior knowledge plays a key role in BayLIME, thus we discuss the following questions on priors to highlight its practical usefulness.

Where can we obtain the priors? The answer to this question varies case by case, depending on the application-specific context. We have implemented three typical ways of obtaining priors, i.e. from other diverse XAI tools, V&V methods, and previous explanations of similar instances. More practical ways of getting priors for BayLIME are certainly possible and we plan to investigate more in future.

How do we know if the prior is good? First of all, we need to be clear about the definition of "good" priors – is it the prior that truly/accurately reflects the unknown behaviour of the AI model? Or the prior gives an explanation that looks good to human users? The latter definition is "cheating" in a Bayesian sense – we cannot rig a prior to get a result we like – and this is not the purpose of XAI methods. The former is sensible, but due to the lack of ground truth behaviours of the black-box AI model, we would never know the prior is good or not *with certainty*. However, we can be *fairly confident* that a given prior is good in some practical cases (e.g., derived from a V&V tool that was shown to be reliable in previous uses), so that it can be utilised by BayLIME.

What if we used a bad prior? Reusing the sensible definition of good priors above, indeed, there could be the case we are using a bad prior that introduces bias. In this case, BayLIME might end up providing an explanation that is "consistently and robustly bad", which seems to imply that BayLIME is only useful when we are certain that the priors

are good. However, in practice, we can never be certain if a prior is good or bad, rather *the prior is simply a piece of evidence to us*. It is not only against the Bayesian spirit but also unwise to discard any evidence without sound reasons. If there *is* a proof that the evidence (either the priors or the new observations) is not trustworthy, then certainly we should not consider it in our Bayesian inference – in this sense, all Bayesian methods depend on good priors, not just BayLIME. So with some caution in deriving priors/evidence from trustworthy ways in practice (a separate engineering problem that is normally out of the scope of a Bayesian method itself), we may ease the concern of BayLIME being "consistently and robustly bad".

9 CONCLUSIONS AND FUTURE WORK

This paper introduces BayLIME, a principled, Bayesian based method to integrate useful prior knowledge into the well-known XAI method LIME. Our theoretical analysis and extensive experiments show that the Bayesian mechanism helps in improving the consistency in repeated explanations of a single prediction, robustness to kernel settings, and the explanation fidelity.

Embedding diverse sources of knowledge has been a clear trend in AI in recent years, as the effort to make black-box learning methods more transparent and interpretable. Emerging techniques peeking inside AI models will lead to many different ways of obtaining knowledge of their black-box behaviour. Therefore, principled methods to integrate diverse knowledge are called for.

In future work, we will investigate more practical ways of obtaining priors and how to leverage the posterior confidence for better explanations. Last but not least, we plan to develop novel use cases of BayLIME.

References

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

David Alvarez-Melis and Tommi S. Jaakkola. On the Robustness of Interpretability Methods. *arXiv e-prints*, art. arXiv:1806.08049, June 2018.

Jose M. Bernardo and Adrian F. M. Smith. *Bayesian theory*. Wiley, 1994. ISBN 0-471-92416-4.

Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006.

Gavin C Cawley and Nicola LC Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.

A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847, 2018.

Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. A formal approach to adaptive software: Continuous assurance of non-functional requirements. *Formal Aspects of Computing*, 24(2):163–186, 2012.

T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access*, 7:47230–47244, 2019.

Wenbo Guo, Sui Huang, Yunzhe Tao, Xinyu Xing, and Lin Lin. Explaining Deep Learning Models – A Bayesian Non-parametric Approach. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4514–4524. Curran Associates, Inc., 2018.

Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, CVPR'15, pages 5353–5360, 2015.

Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

M. G. Kendall and B. Babington Smith. The problem of m rankings. *The Annals of Mathematical Statistics*, 10(3): 275–287, 1939.

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretability. In *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, pages 47–53, Stockholm, Sweden, 2018.

Yi-Shan Lin, Wen-Chuan Lee, and Z. Berkay Celik. What Do You See? Evaluation of Explainable Artificial Intelligence (XAI) Interpretability through Neural Backdoors. *arXiv e-prints*, page arXiv:2009.10639, September 2020.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed Systems Security (NDSS) Symposium 2018*, San-Diego,CA,USA, 2018.

David J. C. MacKay. Bayesian Interpolation. *Neural Computation*, 4(3):415–447, 1992.

Christoph Molnar. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* E-book on leanpub.com, 2020.

Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proc. of the British Machine Vision Conference (BMVC)*, 2018.

Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model Agnostic Supervised Local Explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2515–2524. Curran Associates, Inc., 2018.

Muhammad Rehman Zafar and Naimul Mefraz Khan. DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. In *Proc. of SIGKDD Workshop on Explainable AI/ML (XAI) for Accountability, Fairness and Transparency*, page 6. ACM, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA, 2016. ACM.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. of the IEEE Int. Conf. on Computer Vision*, pages 618–626, 2017.

Sharath M. Shankaranarayana and Davor Runje. ALIME: Autoencoder based approach for local interpretability. In Hujun Yin, David Camacho, Peter Tino, Antonio J. Tallón-Ballesteros, Ronaldo Menezes, and Richard Allmendinger, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2019*, volume 11871 of *LNCS*, pages 454–463, Cham, 2019. Springer.

Sheng Shi, Xinfeng Zhang, and Wei Fan. A modified perturbed sampling method for local interpretable model-agnostic explanation. *arXiv e-prints*, art. arXiv:2002.07434, February 2020.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323 – 332, 2012.

Youcheng Sun, Hana Chockler, Xiaowei Huang, and Daniel Kroening. Explaining image classifiers using statistical fault localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12373 of *LNCS*, pages 391–406, Cham, 2020. Springer International Publishing.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. of the IEEE conference on computer vision and pattern recognition*, CVPR'16, pages 2818–2826, 2016.

Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

Gero Walter and Thomas Augustin. Imprecision and prior-data conflict in generalized Bayesian inference. *Journal of Statistical Theory and Practice*, 3(1):255–271, 2009.

B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 707–723, San Francisco, CA, USA, 2019. IEEE.

Jinsung Yoon, Sercan O. Arik, and Tomas Pfister. RL-LIM: Reinforcement Learning-based Locally Interpretable Modeling. *arXiv e-prints*, art. arXiv:1909.12367, 2019.

Xingyu Zhao, Valentin Robu, David Flynn, Fateme Dinmohammadi, Michael Fisher, and Matt Webster. Probabilistic model checking of robots deployed in extreme environments. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pages 8076–8084, Honolulu, Hawaii, USA, 2019.

Xingyu Zhao, Radu Calinescu, Simos Gerasimou, Valentin Robu, and David Flynn. Interval Change-Point Detection for Runtime Probabilistic Model Checking. In *Proc. of the 35th IEEE/ACM Int. Conf. on Automated Software Engineering*, ASE'20, pages 163–174. ACM, 2020.

A APPENDIX: SUPPLEMENTARY MATERIALS FOR RQ3

In this section, we present the design details and more results of the experiments for **RQ3**. All experiments are conducted on a Macbook Pro 64 bit machine with Intel 2.3GHz 8 cores i9 CPU and 16GB RAM.

A.1 RQ3.A: PRIOR KNOWLEDGE FROM OTHER XAI METHODS

In this example scenario, we use the "partial informative priors" option of BayLIME. The prior parameters μ_0 and λ are elicited from the results of GradCAM, while the α is fitted from data.

To determine μ_0, we take the following steps:

a) Obtain the heatmap (generated by GradCAM) in which each pixel of the image instance has a gradient-value as its importance in the explanation.

b) Do segmentation on the image (using the LIME's default algorithm), then calculate the importance of each superpixel by averaging the importance of all pixels inside the superpixel. This step essentially bridges the gap between pixel-wised heatmap generated by GradCAM and superpixel-wised importance vector used by BayLIME.

c) Map the gradient-value based importance vector (derived from GradCAM) and the coefficient-value based importance vector (derived from linear surrogate models) into a unified scale. While there are alternative ways of doing the scaling, we apply the *maximum absolute value scaling* method based on the same scale of the coefficient vector.

d) Finally, the *superpixel-wised* and *scaled* importance vector (originally derived from GradCAM) is used as μ_o.

To determine λ, we first present some model-structure information of the three CNNs used in our experiments. As summarised in Table 3, for each CNN, we are showing the dimensions of the input layer and the last convolutional layer, based on which the λ can be calculated via Eq. (13). To be exact, suppose the input image dimension is $i \times j$, the activation map of last convolutional layer is $l \times w \times d$, where l, w, and d are the length, width and depth dimension. λ is then calculated by

$$\lambda = \frac{\sqrt{l \cdot w}}{\sqrt{i \cdot j}} d \qquad (13)$$

Intuitively Eq. (13) says:

a) GradCAM calculates the heatmap based on a number of d feature maps, and each feature map is of size $l \times w$.

b) Each feature map represents an abstraction of some features on the original $i \times j$ image. Therefore, it is "equivalent"

to a single perturbed sample (in LIME) with some features switched on and off.

c) Based on this d number of feature maps (or "equivalent perturbed samples"), we form our prior estimates.

d) However, perturbed samples in LIME are weighted according to the distance to the original image. To mimic such weights of samples, we also associate all feature maps with a factor of $\frac{\sqrt{l \cdot w}}{\sqrt{i \cdot j}}$.

e) Finally, the prior precision parameter λ, as the *weighed pseudo-count of prior samples* based on which we form the prior estimates (cf. Sec. 5.2), can be elicited via Eq. (13).

Table 3: Three CNNs under explanation and the configurations of λ for BayLIME with partial informative priors.

Models	InceptionV3	Xception	ResNet-50
Input Layer	299×299	299×299	224×224
Last Convolutional Layer	$8 \times 8 \times 2048$	$10 \times 10 \times 2048$	$7 \times 7 \times 2048$
Partial Informative Priors	$\lambda = 54.8$	$\lambda = 68.5$	$\lambda = 64$

Experimental results on the fidelity of BayLIME using GradCAM explanations as the priors. As described in Sec. 4, we introduce the *deletion* and *insertion* metrics to study explanation fidelity. We compare BayLIME with three other state-of-the-art XAI methods – LIME, GradCAM and the kernel SHAP – on their fidelity in explaining the behaviours of three popular CNNs (cf. Table 3) trained on the dataset ImageNet. We vary the perturb sample size n in the experiments (i.e. n takes $50, 100, 200, 300, 400, 500$).

Similar as LIME, BayLIME allows the selection of top few features by the K-LASSO algorithm, while for brevity, we select all m features when $n > m$ and top-5 features when $n \leq m$.

In addition to the Fig. 4 presented in the main paper, Fig.s 6 and 7 show more statistics and concrete examples. They essentially confirm our conclusions made in Sec. 6. Nevertheless, it is worth noting the following few additional observations:

a) GradCAM has several inherent problems [Chattopadhay et al., 2018], thus it generally performs worse compared to others with larger sample size (e.g., $n > 100$ in our cases). Especially for ResNet50, our experiments show that Grad-CAM performs noticeably worse, even when comparing to other explainers in the case when they are working with smaller sample size. We conjecture that this is due to the known major drawback of GradCAM – the up-sampling process to create the coarse heatmap results in artefacts and loss of signal. ResNet50 has a severer problem here since it requires a relatively more precise up-sampling process from $7 \times 7 \rightarrow 224 \times 224$, compared to InceptionV3 and Xception.

b) The time cost of BayLIME is effectively a sum of running GradCAM (to get the priors) and LIME (to train local sur-
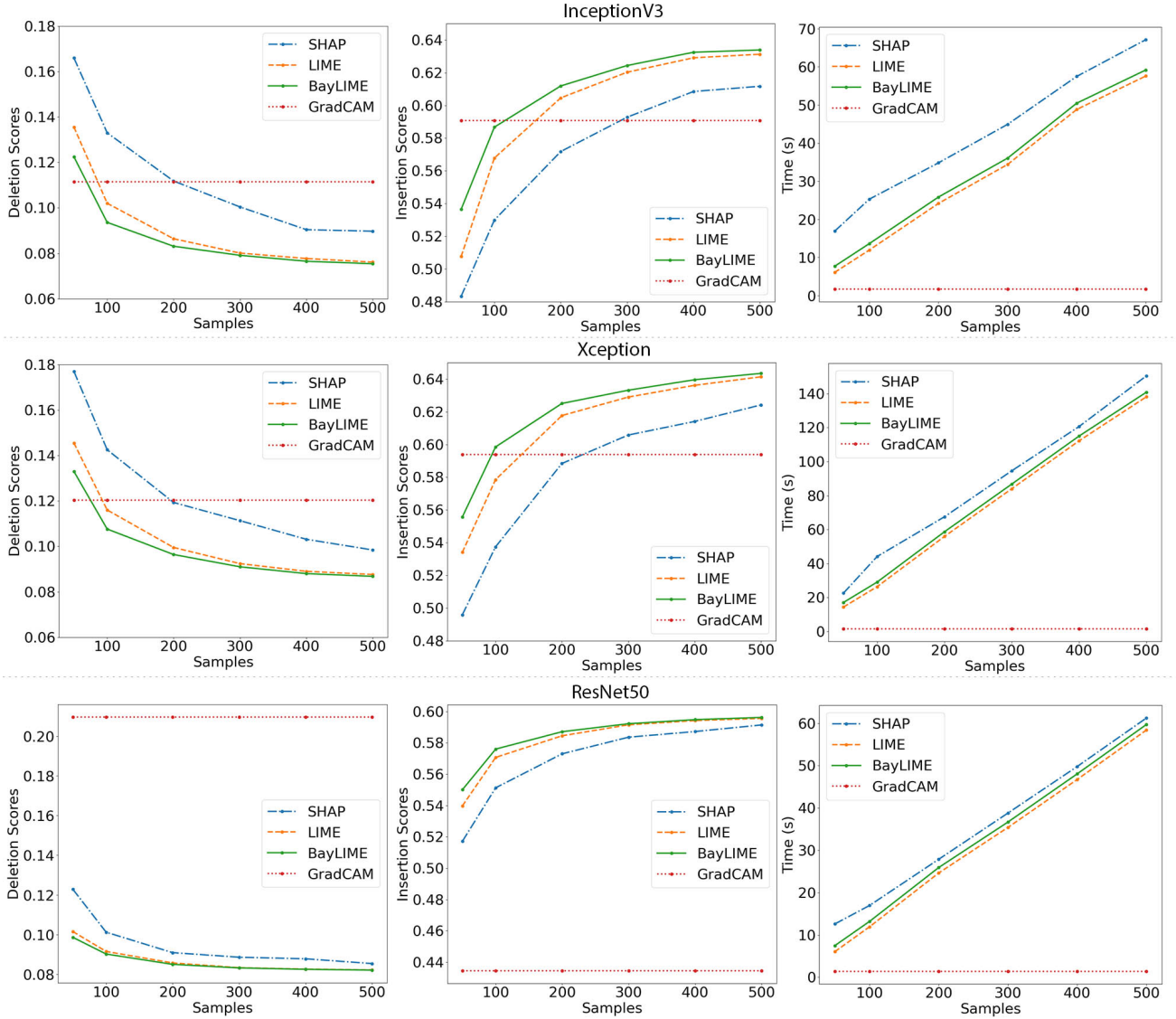
Figure 6: "Deletion and Insertion" evaluation of LIME, GradCAM, kernel SHAP and BayLIME, based on 1000 randomly selected images from ImageNet per sample size value n. The three CNNs InceptionV3, Xception and ResNet50 are studied.

rogate models). GradCAM is quite light-weight that takes little and constant computational time. Benefiting from this, BayLIME takes slightly more time than LIME that is negligible. Thus, we may claim BayLIME is not less computational efficient than LIME, while improves the fidelity.

c) Kernel SHAP (with default hyper-parameters) performs noticeably worse than LIME and BayLIME in all our experiments in terms of both fidelity metrics and computational time. Thus, we omit its examples in Fig. 7.

A.2 RQ3.B: PRIOR KNOWLEDGE FROM V&V METHODS

In this example scenario, we still use BayLIME with partial informative priors. The prior parameters μ_0 and λ are eli-

cited from the results of NeuralCleanse, while the α is fitted from data.

To determine μ_0, we have the following steps:

a) Apply NeuralCleanse to generate the reversed triggers, as shown by the second column in Fig. 8 which are the reversed triggers of BadNet [Gu et al., 2019] and TrojanAttack [Liu et al., 2018] respectively.

b) Convert the reversed triggers into heatmaps. Reversed triggers are approximations on the location of the ground truth backdoor. NeuralCleanse deliberately identifies a minimised trigger as such approximation. Consequently, if we directly use the reversed trigger as our priors, then the importance of pixels that close to the trigger would be ignored. Thus, we use radial basis function to generate the heatmap
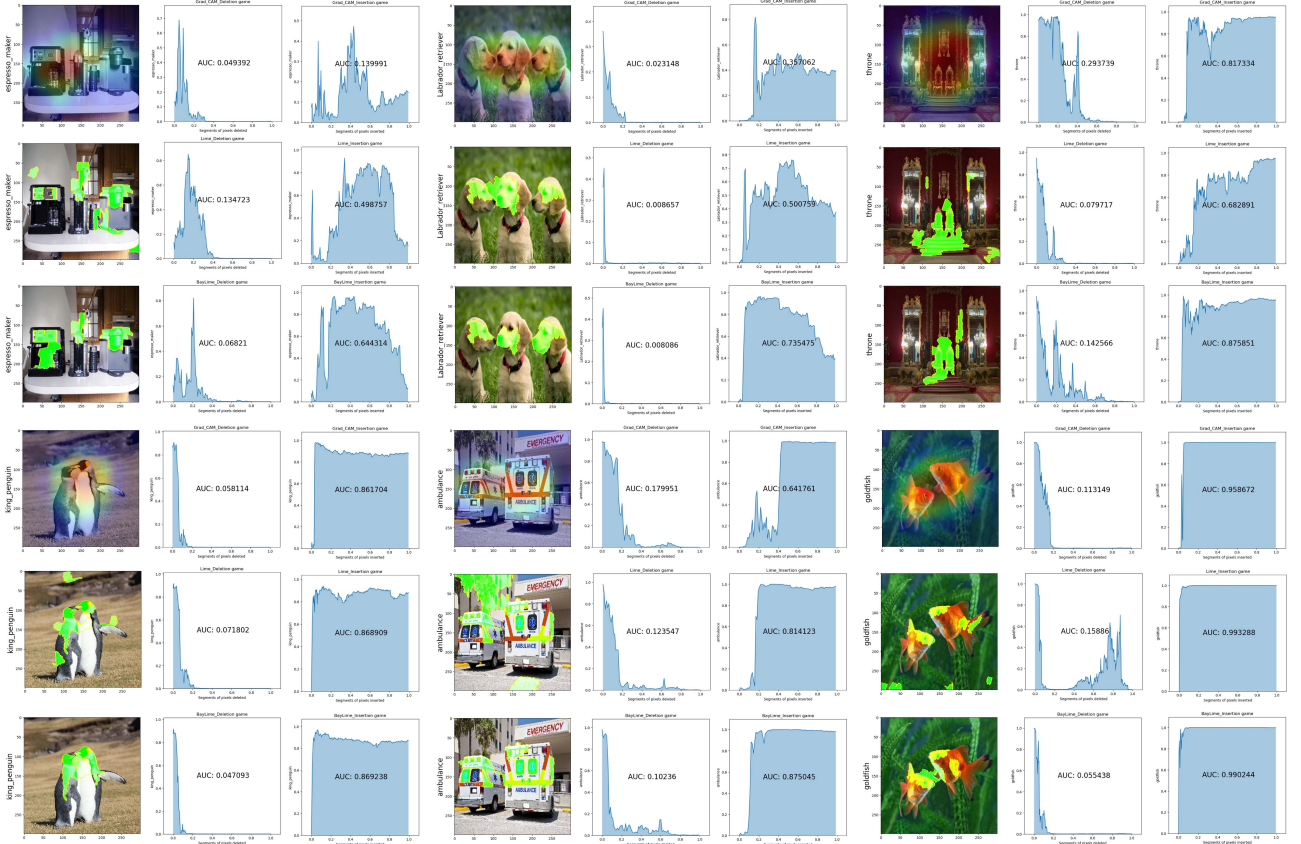
Figure 7: Six sets of examples on the explanations with deletion and insertion measurements (AUC scores). Rows of each set are results from GradCAM, LIME and BayLIME respectively.

by interpolating the importance values around trigger patterns, cf. the third column in Fig. 8.

c) Do segmentation on the image. Instead of using the LIME's default algorithm here, we deliberately apply 1×1 and 2×2 square superpixels (for the two infected CNNs respectively) that are more common in backdoor attacking scenarios.

d) Upon obtaining the heatmap and segmentation of superpixels, the rest of the steps are essentially the same as using GradCAM for the priors (cf. Appendix A.1).
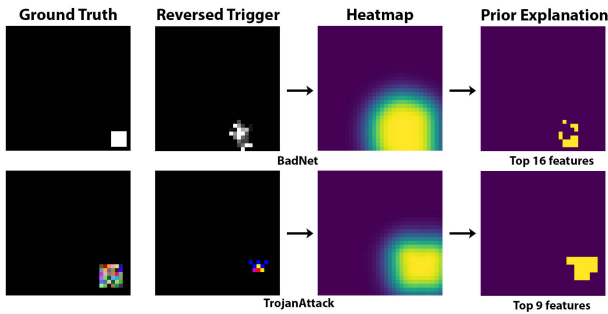

Figure 8: Extract the prior knowledge from reversed triggers.

The determination of λ. λ is calculated based on the anomaly detection intermediate step in the NeuralCleanse method. Basically, NeuralCleanse first obtains the reversed trigger for each class of labels. Next, each reversed trigger is associated with a L_1 norm value. Then the set of all L_1 norm values forms a distribution, on which an anomaly detection process is applied to identify the smallest outlier on that distribution, i.e. the minimal trigger is treated as the approximated backdoor. Thus, we set λ as the cardinality of the set of L_1 norm values that used for the anomaly detection (which is also equal to the total number of classes).

Experimental results on the fidelity of BayLIME using reversed triggers (generated by NeuralCleanse) as the priors. Since we know the ground truth in this scenario, we adopt the Intersection over Union (IoU) metric to study the explanation fidelity (cf. Sec. 4). The 2 infected CNNs are trained on the GTSRB dataset [Stallkamp et al., 2012].

In Table 4, we present the statistics of IoU based on 500 backdoor attacked images. Apparently, BayLIME gets the best IoU scores for both infected models, confirming our conclusions in the main paper.

However, as discussed by the end of Sec. 4, IoU only con-

siders the overlapping of the highlighted area and the ground truth, ignoring how geometrically closed they are when there is no overlapping. We believe the distance between the highlighted area and the ground truth may also provide some insights on the fidelity, thus we develop a complementary metric to the IoU, called Average Minimum Distance (AMD). AMD measures how close the explanation is to the ground truth:

Definition 3 *Suppose we get top k important features (or superpixels) from the explainer for image with size $m \times m$. Then, for each superpixel, we can calculate the geometric centre point $(x_i, y_i), i = 1, ..., k$. The AMD from the ground truth pattern $\mathcal{G} = \{(\tilde{x}_i, \tilde{y}_i)|i = 1, \ldots, \tilde{k}\}$ is to calculate:*

$$AMD = \frac{\sum_{i=1}^{k} \min((x_i, y_i), \mathcal{G})}{k \cdot \sqrt{2(m-1)^2}} \quad (14)$$

We scale the distance between the explanation and ground truth by dividing the maximum distance $\sqrt{2(m-1)^2}$, so that AMD is normalised to $[0, 1]$, irrespective of image dimensions.

Now as shown in the AMD column of Table 4, BayLIME again shows better performance than LIME, although the prior by NeuralCleanse shows the best performance. This is unsurprising due to the unique character of NeuralCleanse that only a "minimised trigger" will be identified to be as closed to the ground truth as possible, but at the risk of completely missing it (i.e., without any overlapping as reflected by IoU).

Table 4: Statistics on IoU (higher is better) and AMD (smaller is better) based on 500 backdoor-attacked images. The priors (derived from reversed triggers) are shown in Fig. 8.

Model	IoU			AMD		
	Prior	LIME	BayLIME	Prior	LIME	BayLIME
BadNet	0.000	0.385	0.406	0.121	0.212	0.178
TrojanAttack	0.150	0.599	0.637	0.011	0.072	0.049

B   AN INCONSISTENCY METRIC WITH MORE EXPERIMENTS TO RQ1

While a quick inconsistency example is shown in Fig. 1 (C)-(E), we introduce a new metric to measure such inconsistency in the following steps :

a) In a single explanation, we first normalise the coefficient vector of m features (represented by $\boldsymbol{\beta}$, cf. the step 6 in the LIME workflow in Appendix C) into a unit vector $\hat{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}}{||\boldsymbol{\beta}||}$.

b) For repeated k explanations (denoted by the upper index $^{(k)}$ in the following notations), the normalised importance of a feature i (i.e. $|\hat{\beta}_i|$) and its ranking may form two probability distributions, denoted respectively as $g_i^{(k)} : [0, 1] \rightarrow [0, 1]$ and $f_i^{(k)} : [1, m] \rightarrow [0, 1]$.

c) The *index of dispersion* (IoD) of $f_i^{(k)}$ essentially measures the relative variability of the ranks of the feature i in the k repeated explanations, denoted as $IoD(f_i^{(k)})$. To be consistent, we would expect a small $IoD(f_i^{(k)})$.

d) The average importance of each feature $\mathbb{E}(g_i^{(k)})$ is quite different. Intuitively, we need to weight each $IoD(f_i^{(k)})$ accordingly.

e) Finally, we define:

Definition 4 *The inconsistency of repeated k explanations for an instance with m features is measured by:*

$$\sum_{i=1}^{m} \frac{\mathbb{E}(g_i^{(k)})}{\sum_{j=1}^{m} \mathbb{E}(g_j^{(k)})} IoD(f_i^{(k)}) \quad (15)$$

where $g_i^{(k)} : [0, 1] \rightarrow [0, 1]$ and $f_i^{(k)} : [1, m] \rightarrow [0, 1]$ are probability distributions of the normalised importance of feature i and its ranks in k explanations, respectively.

Intuitively, the inconsistency is a weighted sum of IoD of each feature's ranks in repeated explanations, so that the IoD of a more important feature is weighted higher. While there could be a range of similar metrics that can be defined, we choose the most intuitive one to use in this paper. Since our main purpose is to complement Kendall's W in some corner cases and double check our experimental conclusions, rather than to justify any particular metric.

An example explanation used to illustrate the calculation of (15) is shown in Tab. 5. First, the normalised coefficient vectors of the 3 repeat explanations[6] are (presented in row)

$$[0.799, -0.599, 0.044, 0.036]$$
$$[0.726, -0.685, 0.044, 0.036]$$
$$[0.784, -0.619, 0.041, 0.037]$$

Thus,

$$\mathbb{E}(g_1^{(3)}) = \frac{|0.799| + |0.726| + |0.784|}{3} = 0.770$$

$$\mathbb{E}(g_2^{(3)}) = \frac{|-0.599| + |-0.685| + |-0.619|}{3} = 0.634$$

$$\mathbb{E}(g_3^{(3)}) = \frac{|0.044| + |0.036| + |0.037|}{3} = 0.039$$

$$\mathbb{E}(g_4^{(3)}) = \frac{|0.036| + |0.044| + |0.041|}{3} = 0.040$$

---

[6] We set $k = 3$ here for illustration, while generally k should be large enough to let both the $f_i^{(k)}$ and $g_i^{(k)}$ converge for more accurate measurements.
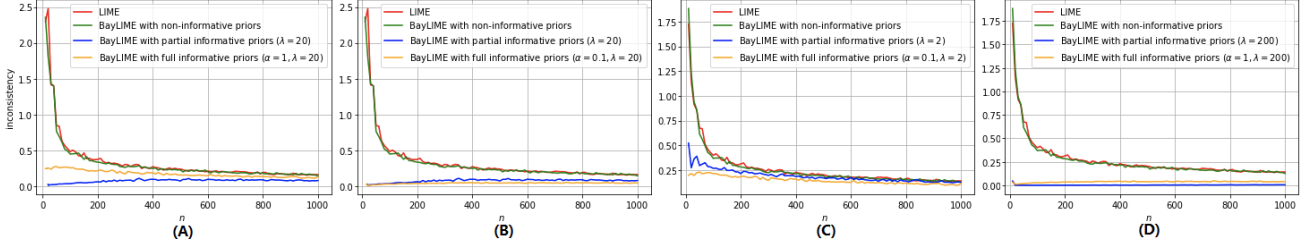
Figure 9: To complement Kendall's W, the inconsistency metric Eq. (15) in $k = 200$ repeated explanations by LIME and BayLIME on tabular data. Each set shows an illustrative combination of α and λ.
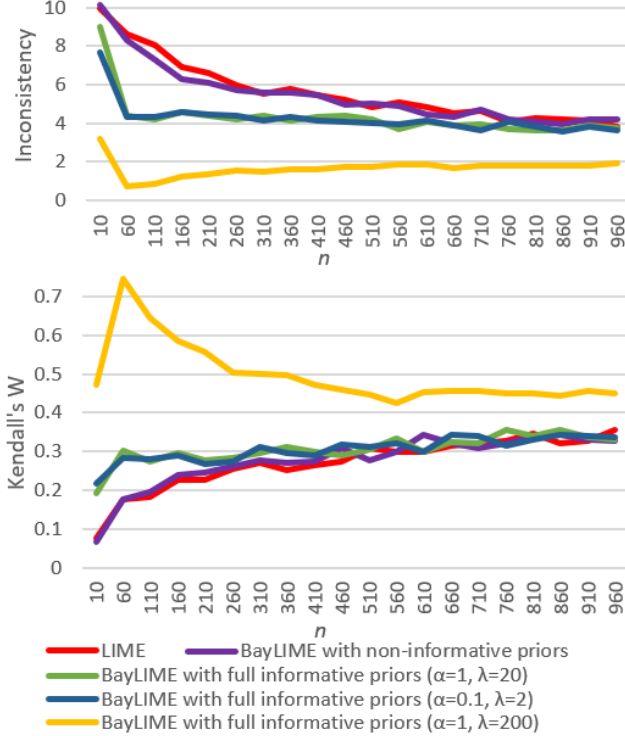

Table 5: An illustrative example of 3 repeated explanations on an instance with 4 features.

feature rankings	exp. 1		exp. 2		exp. 3	
	fea. ID	coef.	fea. ID	coef.	fea. ID	coef.
1st	1	0.2	1	0.18	1	0.19
2nd	2	-0.15	2	-0.17	2	-0.15
3rd	3	0.011	4	0.011	4	0.01
4th	4	0.009	3	0.009	3	0.009

equally. In contrast, our metric Eq. (15) *weights* the IoD of the discrete ranks of each feature by its (continuous) importance. With the extra information considered (compared to Kendall's W), our metric Eq. (15) thus complements Kendall's W in terms of: (i) discriminating explanations with the same ranks of features but the importance vectors are different; (ii) minimising the effect from the normal fluctuation of the ranks of less-important/irrelevant features.

C PRELIMINARIES ON LIME

LIME [Ribeiro et al., 2016] implements a local surrogate model that is used to explain individual predictions of black-box AI models. The intuition behind LIME is as follows. For a given black-box AI model (**model-agnostic**), we may probe it as many times as possible by perturbing some features (e.g., hiding superpixels of an image) of the input instance of interest (**locally**) and see how the prediction changes. Such changes in the predictions, as we vary the features of the instance, can help us understand why the AI model made a certain prediction over the instance. Then a new dataset consisting of the perturbed inputs and the corresponding predictions made by the black-box AI model can be generated, upon which LIME trains a surrogate model that is **interpretable** to humans (e.g., linear regressors, cf. [Molnar, 2020, Chpt. 4] for more). The training of the interpretable surrogate model is weighted by the proximity of the perturbed samples to the original instance.

To be exact, the basic steps of how LIME works are:

1) Choose an instance $\boldsymbol{x}$ to interpret (e.g., an image or a row

Figure 10: For different explainers, the inconsistency of (15) and Kendall's W in repeated explanations of images labelled by InceptionV3 as functions of the perturbed sample size n.

Also,

$$IoD(f_1^{(3)}) = Var([1,1,1])/\mathbb{E}([1,1,1]) = 0$$
$$IoD(f_2^{(3)}) = Var([2,2,2])/\mathbb{E}([2,2,2]) = 0$$
$$IoD(f_3^{(3)}) = Var([3,4,4])/\mathbb{E}([3,4,4]) = 0.09$$
$$IoD(f_4^{(3)}) = Var([4,3,3])/\mathbb{E}([4,3,3]) = 0.1$$

Finally we get 0.00506 as the inconsistency via (15).

Fig.s 9 and 10 show more experimental results of both Kendall's W and the inconsistency metric Eq. (15) based on more instances, e.g., images, confirming our conclusions on RQ1. As expected, the plots of metric Eq. (15) can be interpreted similarly as Kendall's W. But Kendall's W only considers the (discrete) ranks of each feature and treats all features

from a tabular dataset) in which there are m features[7] (e.g., the superpixels for images or columns for tabular data).

2) Do perturbation on $\boldsymbol{x}$ (e.g., switch on/off image superpixels) and generate a new input set $\boldsymbol{X} = \{\boldsymbol{x_1}, ..., \boldsymbol{x_n}\}$ with n samples, i.e. $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{n \times m}$.

3) Probe the black-box AI model with $\boldsymbol{X}$ and record the predictions as a column vector $\boldsymbol{y} = [y_1, ..., y_n]^T$.

4) Weight the n perturbed samples in $\boldsymbol{X}$ according to their proximity to the original instance $\boldsymbol{x}$. Say the weights calculated by some kernel function (by default, an exponential kernel defined on some kernel width) are $\{w_1, ..., w_n\}$, and denote the new weighted dataset[8] as $(\boldsymbol{X}', \boldsymbol{y}')$.

5) On the dataset $(\boldsymbol{X}', \boldsymbol{y}')$, train a linear regressor

$$\boldsymbol{y}' = \boldsymbol{X}'\boldsymbol{\beta} + \epsilon \qquad (16)$$

where $\boldsymbol{\beta}$ and ϵ are the coefficients and Gaussian white noise.

6) Then, the *absolute values* of the coefficient vector $\boldsymbol{\beta}$ represent the importance of the m features, based on which rankings can be done. By default, LIME [9] uses the Ridge regressor with weighted samples:

$$\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} + r\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y} \qquad (17)$$

where r is the Ridge regularisation parameter, and $\boldsymbol{W} = diag(w_1, ..., w_n)$ is a diagonal matrix with diagonal elements equal to those w_i's. It becomes the weighted ordinary least squares estimate $\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{y}$ when $r = 0$.

---

[7]LIME allows the selection of top few features by K-LASSO, while for brevity, we use all features when $n > m$ and top-5 features when $n \leq m$.

[8]Depends on the specific implementation, e.g., by Python scikit-learn-0.22.1, this is equivalent to train a linear regressor on a scaled dataset $(diag(\sqrt{w_1}, \ldots, \sqrt{w_n})\boldsymbol{X}, diag(\sqrt{w_1}, \ldots, \sqrt{w_n})\boldsymbol{y})$.

[9]LIME's Python implementation at `https://github.com/marcotcr/lime/`.