Backward Compatibility in Attributive Explanation and Enhanced Model Training Method

Ryuta Matsuno^{a,*}

^aNEC Corporation

Abstract. Model update is a crucial process in the operation of ML/AI systems. While updating a model generally enhances the average prediction performance, it also significantly impacts the explanations of predictions. In real-world applications, even minor changes in explanations can have detrimental consequences. To tackle this issue, this paper introduces BCX, a quantitative metric that evaluates the backward compatibility of feature attribution explanations between pre- and post-update models. BCX utilizes practical agreement metrics to calculate the average agreement between the explanations of pre- and post-update models, specifically among samples on which both models accurately predict. In addition, we propose BCXR, a BCX-aware model training method by designing surrogate losses which theoretically lower bounds agreement scores. Furthermore, we present a universal variant of BCXR that improves all agreement metrics, utilizing L2 distance among the explanations of the models. To validate our approach, we conducted experiments on eight real-world datasets, demonstrating that BCXR achieves superior trade-offs between predictive performances and BCX scores, showcasing the effectiveness of our BCXR methods.

Introduction

For effective operation of machine learning (ML) systems (i.e., MLOps), model updates are essential to exploit newly collected data and to adopt the changes in data [12, 26, 36, 35]. Model updates basically replace an old model (i.e., a pre-update model) with a new model (i.e., a post-update model) trained using more recent and/or larger amounts of data. Typically, this leads to an improvement in the average prediction performance, but local prediction performance may worsen. Backward compatibility metrics have been proposed to assess these performance degradation [2, 39, 33, 28, 17]. Furthermore, backward-compatibility-aware retraining methods for model updates have been developed [2, 39, 28, 17], revealing that there is a trade-off between backward compatibility and prediction performance of a new model over the old model.

While predictive performance is important for ML models, there are other important demands as well; explainability is one of them, which is often as crucial as predictive performance for sensitive and critical domains, such as healthcare and security. Recently, explanation methods for ML models, a.k.a. XAI (eXplainable AI), have been actively researched, and various post-hoc and model-agnostic attributive explanation methods [38] have been proposed, including LIME [24], Anchors [25], and SHAP [15, 16].

Ensuring that the explanations of the new model align with ones of the old model is crucial in real-world applications. Even though the average prediction performance is improved, the practitioners might hesitate to adopt a new model if it presents different explanations, as this can lead to confusion regarding the real use of the prediction with explanation. Typically, users perceive the new model as less reliable when they are already familiar with the behavior of the old model [2]. While a few studies have examined the disagreement between different explanation methods [21, 13, 6, 7], the compatibility of explanations during model updates has yet to be explored.

In this study, we introduce a new metric called BCX (Backward Compatibility of eXplanation) to assess the consistency of attributive explanations between old and new models using four practical top-k feature-based agreement metrics [13]. BCX calculates the average agreement of explanations between the old and new models for samples where both models make correct predictions, providing a measure of compatibility in explanations alongside predictive performance. We then propose BCXR (BCX-aware Retraining) methods. Since the agreement metrics used in BCX themselves are not differentiable, we propose differential surrogate losses that have theoretical validity for substitution. Additionally, we present a universal variant of BCXR that can improve the compatibility regardless of the choice of agreement metrics. To evaluate the effectiveness of our methods, we conduct experiments on eight real-world datasets. The results demonstrate that BCXR achieves a better trade-offs between BCX scores and predictive performance, thus showing promising efficacy. Notably, we observe that when the number of features is large, BCXR even outperforms retraining without considering BCX in terms of the predictive performance. Overall, this study provides a method to evaluate and enhance the compatibility of explanations during model updates, contributing to the establishment of trustworthy and responsible MLOps.

To summarize, our contributions in this study includes:

- (a) We are the first, to the best of our knowledge, to define a backward compatibility metric for prediction explanation and propose BCX. BCX utilizes practical agreement metrics to assess the consistency of explanations between old and new models.
- (b) We propose BCXR, a BCX-aware retraining method that ensures theoretical validity by using differentiable surrogate losses to lower bound the non-differentiable agreement metrics.
- (c) We conduct experiments on eight real-world datasets to validate the effectiveness of BCXR. The empirical evidence obtained from these experiments demonstrates the efficacy of our BCXR methods.

^{*} Email: ryuta-matsuno@nec.com.

The rest of the paper is organized as follows: We begin by introducing our notation and reviewing related works in Section 2. Section 3 presents our proposed methods, BCX and BCXR. In Section 4, we report the results of our numerical evaluation. Finally, Section 5 concludes the paper. The proofs of our theoretical analysis, the details of our experiments, and a discussion on the limitations of our method are provided in Appendix.

2 Preliminary

In this section, we briefly introduce the notation we use throughout this paper, as well as relevant previous methods.

2.1 Notation

We study supervised regression and classification problems. The input space is $\mathcal{X} \subseteq \mathbb{R}^d$, where \mathbb{R} is the space of real values, $d \in \mathbb{N}$ is the number of input features, and \mathbb{N} is the space of integers larger than zero. The output space is $\mathcal{Y} \subseteq \mathbb{R}$ for regression tasks and $\mathcal{Y} = [K]$ for classification tasks, where [K] denotes the set of integers from 1 to $K \in \mathbb{N}$, i.e., $[K] := \{1,...,K\}$, and K > 1 is the number of classes.

We follow the model update schema with additional data, which is set up in studies of backward compatibility metrics [28, 17]. Let $\mathcal{H} = \{h: \mathcal{X} \to \mathcal{Y}\}$ be a hypothesis space. An old model $h_1 \in \mathcal{H}$ is trained with data $D_1 := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_1}$ drawn from a density denoted by $p(\boldsymbol{x}, y)$ in an i.i.d. fashion. After obtaining additional data $D_\Delta := \{(\boldsymbol{x}_i, y_i)\}_{i=n_1+1}^{n_2}$ from $p(\boldsymbol{x}, y)$, we train a new model $h_2 \in \mathcal{H}$ using $D_2 := D_1 \cup D_\Delta$.

An attributive explanation method $E:\mathcal{H}\times\mathcal{X}\to\mathbb{R}^d$ provides the explanation of the prediction of a model $h\in\mathcal{H}$ for an input $\boldsymbol{x}\in\mathcal{X}$ by computing a vector of real values in \mathbb{R}^d whose i-th value represents the influences (e.g., importance, relevance, or contribution) of the i-th feature for the prediction $h(\boldsymbol{x})$.

2.2 Related works

Related works can be categorized into three groups: backward compatibility, explanation methods, and studies on disagreement in ML.

2.2.1 Backward compatibility in ML

The concept of backward compatibility in ML was originally introduced by Bansai et al. [2], who proposed the *Backward Trust Compatibility* (BTC) metric to measure the backward compatibility between old and new classification models (h_1 and h_2 , respectively)

$$BTC(h_1, h_2) := \frac{\mathbb{E}_{p(\boldsymbol{x}, y)}[\mathbb{I}[h_1(\boldsymbol{x}) = y \land h_2(\boldsymbol{x}) = y]]}{\mathbb{E}_{(\boldsymbol{x}, y)}[\mathbb{I}[h_1(\boldsymbol{x}) = y]]}, \quad (1)$$

where $\mathbb{I}[P]$ represents the Iverson bracket, being 1 if the proposition P is true and 0 otherwise, and $\mathbb{E}_{p(\boldsymbol{x},y)}[f(\boldsymbol{x},y)] := \int_{\mathcal{X}\times\mathcal{Y}} f(\boldsymbol{x},y)p(\boldsymbol{x},y)dxdy$ denotes the expectation of $f(\boldsymbol{x},y)$ over the density $p(\boldsymbol{x},y)$. BTC measures the ratio of correct predictions made by the new model among the samples for which the old model makes correct predictions. The authors then proposed a BTC-aware retraining objective for a new classifier h_2 defined by

$$L_{DM}(h_2) := \mathbb{E}_{p(\boldsymbol{x},y)} \left[(1 + \lambda \mathbb{I}[h_1(\boldsymbol{x}) = y]) \ell(h_2(\boldsymbol{x}), y) \right], \quad (2)$$

where $\ell: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \geq 0$ is a loss function, and $\lambda \in \mathbb{R}_{>0}$ is a hyperparameter. The minimization of Eq. (2) is referred to as *Dissonance Minimization* (DM) [2, 28]. DM is versatile since it simply modifies the sample weights of the training data and hence it can be applied to most ML methods.

Another backward compatibility metric is *Backward Error Compatibility* (BEC) [33], which focuses specifically on prediction errors. Meanwhile, the *Negative Flip Rate* (NFR) [39] counts the number of samples for which the old model makes correct predictions while the new model makes incorrect predictions. Sakai [28] generalized these backward compatibility metrics as a *Generalized Backward Compatibility* (GBC) metric and theoretically established a generalization error bound of GBC-based learning. Additionally, *ABCD* [17] is proposed as a robust backward compatibility metric that defines compatibility based on the conditional distribution, which is approximated by *k*-nearest neighbors. While a few backward-compatibility-aware retraining methods [39, 28, 17] have been proposed beside DM, they are not as versatile as DM due to their objective customization.

2.2.2 Explanation methods in ML

Explainability is one of the most critical aspects of ML/AI systems, particularly in sensitive and critical domains such as health-care and social security. As a result, various eXplainable AI (XAI) methods have been proposed [3, 27, 18]. For fundamental tasks, intrinsically explainable methods, such as decision trees, linear models, and k-nearest neighbors, are utilized. However, for complex tasks, black-box models like neural networks and kernel methods are commonly employed. To explain these models, post-hoc feature-attribution-based explanation methods [38] have been developed [24, 31, 25, 29, 32, 34].

One of the most prevalent explanation methods is SHAP [15], which utilizes the concept of Shapley values [30] to explain a prediction by the sum of the contributions of each input feature. While SHAP can be model-agnostic by implementing *Kernel SHAP* [15], various specialized implementation have been proposed. For example, tree-based [16], gradient-based and other SHAP computation methods are officially available¹. In addition, many SHAP-related research have been conducted for better approximation and faster computation [11, 14, 1, 5, 10, 37].

2.2.3 Disagreement measures of attributive explanations

It has been revealed that attributive explanations obtained from different methods often disagree with each other [21, 13]. To measure the disagreements between two explanation methods for a single model, various metrics have been proposed [21, 13, 6, 7]. For example, Krishna et al. [13] proposed top-k feature agreement (Sørensen–Dice coefficient of top-k features), top-k rank agreement, top-k sign agreement, top-k signed rank agreement, based on practitioners' perspectives. Since practically meaningful agreement metrics may depend on applications, these various design of metrics are important. The agreement measures are defined as follows;

$$\begin{aligned} &\operatorname{FtrAgr}(\boldsymbol{e}_{1},\boldsymbol{e}_{2};k) \\ &:= \frac{1}{k} \Big| \big\{ i \in [d] \ \big| \ i \in \operatorname{TopFeat}(\boldsymbol{e}_{1};k) \land i \in \operatorname{TopFeat}(\boldsymbol{e}_{2};k) \big\} \Big| \end{aligned} \tag{3}$$

¹ https://shap-lrjball.readthedocs.io/

$$RnkAgr(e_1, e_2; k)$$

$$:= \frac{1}{k} \Big| \Big\{ i \in [d] \ \Big| \ i \in \text{TopFeat}(\boldsymbol{e}_1; k) \land i \in \text{TopFeat}(\boldsymbol{e}_2; k) \\ \land \text{rank}(\boldsymbol{e}_1, i) = \text{rank}(\boldsymbol{e}_2, i) \Big\} \Big|$$
(4)

 $\operatorname{SgnAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)$

$$:= \frac{1}{k} \Big| \Big\{ i \in [d] \Big| i \in \text{TopFeat}(\boldsymbol{e}_1; k) \land i \in \text{TopFeat}(\boldsymbol{e}_2; k) \\ \land \operatorname{sgn}(e_{1i}) = \operatorname{sgn}(e_{2i}) \Big\} \Big|$$
 (5)

 $\operatorname{SgnRnkAgr}(\boldsymbol{e}_1,\boldsymbol{e}_2;k)$

$$:= \frac{1}{k} \Big| \Big\{ i \in [d] \Big| i \in \text{TopFeat}(\boldsymbol{e}_1; k) \land i \in \text{TopFeat}(\boldsymbol{e}_2; k) \\ \land \text{sgn}(\boldsymbol{e}_{1i}) = \text{sgn}(\boldsymbol{e}_{2i}) \land \text{rank}(\boldsymbol{e}_1, i) = \text{rank}(\boldsymbol{e}_2, i) \Big\} \Big|$$
(6)

where $\operatorname{rank}(\boldsymbol{x},i) := |\{j \in [d] \mid |x_j| \geq |x_i|\}|$ outputs the rank of the absolute of x_i among the absolutes of elements of \boldsymbol{x} in descending order (i.e., $|x_i|$ is the $(\operatorname{rank}(\boldsymbol{x},i))$ -th largest value among $|x_1|, \dots, |x_d|^2$, TopFeat $(\boldsymbol{x};k) := \{i \in [d] \mid \operatorname{rank}(\boldsymbol{x},i) \leq k\}$ is the set of indices where that ranks of the corresponding elements of \boldsymbol{x} are smaller than or equal to k (i.e., set of indices of features whose absolute value is at least k-th largest), and $\operatorname{sgn}(x) := 1$ if $x \geq 0$ else -1, is the sign of x. Note that these agreement metrics are invariant to the replacement of e_1 and e_2 .

Although our interest aligns with these studies to some extent and we utilize the agreement metrics exemplified above, we aim at investigating the differences between two models using a single explanation method, where differences between two explanation methods for a single model have been studied. Thus, although Neely et al. [21] conclude that agreement is not a suitable criterion for evaluating explanations, we still maintain that the explanations of both old and new models should agree for consistent model updates.

3 Proposed method

In this section, we first propose our Backward Compatibility metric in eXplanations, which we call BCX. Then we present our BCX-aware Retraining method, which we call BCXR. Please note that we omit k from notation of agreement metrics in our analyses, e.g., we denote $\operatorname{Agree}(e_1,e_2)$ instead of $\operatorname{Agree}(e_1,e_2;k)$ for the sake of readability, while the statements hold true for any choice of $k \in [d]$. In addition, all proofs are presented in our appendix.

3.1 Backward compatibility in explanations

We define the backward compatibility metric in terms of attributive explanation of models' prediction as follows using any choice of explanation method and agreement metric to quantify the agreement between two explanations.

Definition 1 (Backward Compatibility in eXplanations). Given two models h_1 and $h_2 \in \mathcal{H}$, an attributive explanation method $E : \mathcal{H} \times \mathcal{X} \to \mathbb{R}^d$, and an agreement metric Agree : $\mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$, the Backward Compatibility in eXplanation (BCX) of h_2 over h_1 is

defined as

$$BCX(h_1, h_2; Agree, E) := \frac{\mathbb{E}_{p(\boldsymbol{x}, y)} \left[Agree(E(h_1, \boldsymbol{x}), E(h_2, \boldsymbol{x})) \cdot s(\boldsymbol{x}, y; h_1, h_2) \right]}{\mathbb{E}_{p(\boldsymbol{x}, y)} \left[s(\boldsymbol{x}, y; h_1, h_2) \right]}, \quad (7)$$

where the sample selection function $s(\boldsymbol{x},y;h_1,h_2)$ is defined as

$$s(\mathbf{x}, y; h_1, h_2) := c(h_1(\mathbf{x}), y) \cdot c(h_2(\mathbf{x}), y)$$
 (8)

and

$$c(\widehat{y}, y) := \begin{cases} \mathbb{I}\left[(\widehat{y} - y)^2 \le \tau\right] & \text{(regression)} \\ \mathbb{I}\left[\widehat{y} = y\right] & \text{(classification)} \end{cases}$$
(9)

is the correctness of the prediction by h for the sample (x,y). au is a predefined threshold to determine the correctness for regression tasks.

Our definition of BCX is both practical and meaningful. A straightforward approach to defining BCX involves computing the expected agreement scores between h_1 and h_2 , for all samples $(\boldsymbol{x},y) \sim p(\boldsymbol{x},y)$. For instance, this can be achieved by computing $\mathbb{E}_{p(\boldsymbol{x},y)}[\mathrm{Agree}(E(h_1,\boldsymbol{x}),E(h_2,\boldsymbol{x}))]$. However, this approach may not be suitable for practical use due to two reasons. First, aligning explanations when the old model gives incorrect predictions (e.g., $h_1(\boldsymbol{x}) \neq y$) may have little practical value. Second, it is impractical to have aligned explanations when the new model provides incorrect predictions (e.g., $h_2(\boldsymbol{x}) \neq y$). Hence, it is essential to focus on the agreement scores for samples where both old and new models make correct predictions. Based on this motivation, we have devised our definition of BCX. A high BCX score indicates that the new model consistently provides compatible explanations for samples with compatibly correct prediction.

In this work, we mainly investigate SHAP for the explanation method E due to its prevalence in real applications and recent active studies. To assess the agreement between explanations, we employ the four agreement metrics introduced in Section 2.2.3. These metrics are specifically designed from a practitioner's perspective and offer practical utility.

3.2 BCX-aware retraining

Next, we aim at training a new model h_2 where a high BCX score of h_2 over h_1 is preferred. The training objective of h_2 is naturally formulated as follows, similarly with the formulation in [2, 17].

$$R(h_2) := \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(h_2(\boldsymbol{x}),y)] + \lambda(1 - \mathrm{BCX}(h_1, h_2; \mathrm{Agree}, E)),$$
(10)

where $\ell: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is a loss function, e.g., squared error for regression and 0-1 loss for classification. Agree is one of feature-agreement, rank-agreement, sign-agreement, and signedrank-agreement with given k.

Differential surrogate loss design. In order to train the model h_2 , it is necessary for its objective function to be differentiable w.r.t. the parameters of h_2 . Then, we have two issues regarding the differentiability; The one is the differentiability of the explanation method E and the other is the differentiability of the agreement metric Agree.

The equation $\exists i>j\in [d], |x_i|=|x_j|, \text{ we set } \mathrm{rank}(\boldsymbol{x},j)=\mathrm{rank}(\boldsymbol{x},i)+1 \text{ for consistency.}$

 $^{^3}$ We abuse to define $\mathrm{sgn}(0)=1$ for mathematical simplicity in our theoretical analysis.

 $^{^4}$ au can be a user-defined hyperparameter. For example, we set the threshold au to be the empirical mean squared error (MSE) of an old model h_1 (i.e., we use $au:=1/|D_2|\sum_{({m x},y)\in D_2}(h_1({m x})-y)^2$ in our experiments).

For the former, we can use a differentiable explanation method for E and use a differentiable model for h_2 (e.g., neural networks). Specifically, we use the gradient-based SHAP as a differentiable SHAP computation in our experiments. It should be noted that our formulation and analysis are general and hence other differentiable explanation methods [31, 29, 32, 34] can also be utilized.

For the latter, however, the agreement metrics lack differentiability due to their discrete nature. Consequently, we propose a differentiable surrogate loss that provides an upper bound for $1-\mathrm{BCX}(h_1,h_2;\mathrm{Agree},E)$ in equation Eq. (10), in order to design a differentiable objective. Specifically, we first consider feature-agreement and we define our surrogate loss $\ell_{\mathrm{Ftr}}(e_2;e_1,k)$ to lower bound $\mathrm{Ftr}\mathrm{Agr}(e_1,e_2;k)$ as follows

$$\ell_{\text{Ftr}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k) := \frac{1}{k} \sum_{i \in \text{TopFeat}(\boldsymbol{e}_1; k)} \max \left(0, \psi_{feat}(\boldsymbol{e}_2) - |e_{2i}| + \varepsilon\right), \quad (11)$$

where $\psi_{feat}: \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\psi_{feat}(\mathbf{e}_{2}) := \begin{cases} \max_{i \notin \text{TopFeat}(\mathbf{e}_{1};k)} |e_{2i}| & (k < d) \\ -\varepsilon & (\text{otherwise}) \end{cases}$$
(12)

and $\varepsilon>0$ is a predefined small constant. We establish the following lemma between the surrogate loss $\ell_{\rm Ftr}$ and feature-agreement metric FtrAgr, which provides theoretical validity of the use of $\ell_{\rm Ftr}$.

Lemma 1. The following inequality holds for any e_1 , e_2 , and k.

$$1 - \operatorname{FtrAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) \le \varepsilon^{-1} \ell_{\operatorname{Ftr}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)$$
 (13)

Lemma 1 shows that $\ell_{\rm Ftr}$ multiplied with ε^{-1} upper bounds one minus feature-agreement (i.e., feature-**dis**agreement) and hence minimization of $\ell_{\rm Ftr}$ maximizes the score of feature-agreement.

Objective for BCXR. Now we can upper bounds the non-differentiable term $(1 - BCX(h_1, h_2; \operatorname{FtrAgr}, E))$ in Eq. (10) based on Lemma 1 as

$$1 - BCX(h_1, h_2; \operatorname{FtrAgr}, E)$$

$$= \frac{\mathbb{E}_{p(\boldsymbol{x}, y)} \left[\left(1 - \operatorname{FtrAgr}(E(h_1, \boldsymbol{x}), E(h_2, \boldsymbol{x})) \right) s(\boldsymbol{x}, y; h_1, h_2) \right]}{\mathbb{E}_{p(\boldsymbol{x}, y)} \left[s(\boldsymbol{x}, y; h_1, h_2) \right]}$$
(14)

$$\leq \varepsilon^{-1} \frac{\mathbb{E}_{p(\boldsymbol{x},y)} \left[\ell_{\text{Ftr}}(E(h_2,\boldsymbol{x}); E(h_1,\boldsymbol{x}), k) s(\boldsymbol{x}, y; h_1, h_2) \right]}{\mathbb{E}_{p(\boldsymbol{x},y)} \left[s(\boldsymbol{x}, y; h_1, h_2) \right]}. \tag{15}$$

Hence we have the following upper bound of $R(h_2)$ with FtrAgr, which is differentiable w.r.t. h_2 ;

$$R(h_2) \leq \mathbb{E}_{p(\boldsymbol{x},y)}[\ell(h_2(\boldsymbol{x}),y)] + \lambda \frac{\mathbb{E}_{p(\boldsymbol{x},y)}\left[\ell_{\text{Ftr}}(E(h_2,\boldsymbol{x});E(h_1,\boldsymbol{x}),k)s(\boldsymbol{x},y;h_1,h_2)\right]}{\mathbb{E}_{p(\boldsymbol{x},y)}\left[s(\boldsymbol{x},y;h_1,h_2)\right]}$$
(16)

$$=: L_{\mathrm{Ftr}}(h_2), \tag{17}$$

where the constant ε^{-1} is absorbed by λ for simplicity and we denote the right hand of Eq. (16) by $L_{\rm Ftr}(h_2)$. In practical scenarios, we resort to the empirical approximation of $L_{\rm Ftr}(h_2)$ for the BCX-aware retraining, since we cannot know the underlying distribution $p(\boldsymbol{x},y)$. Formally, our proposed feature-agreement-based BCX-aware retraining method (referred to as BCXR-Ftr) is defined as follows.

Definition 2 (Feature-agreement-based BCX-aware Retraining (BCXR-Ftr)). Given an old model $h_1: \mathcal{X} \to \mathcal{Y}$, and a training data $D := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, BCXR trains a new model h_2 by minimizing the following objective;

$$\widehat{L}_{Ftr}(h_2; D) := \frac{1}{|D|} \sum_{(\boldsymbol{x}, y) \in D} \ell(h_2(\boldsymbol{x}), y)$$

$$+ \lambda \frac{1}{|D_s|} \sum_{\boldsymbol{x} \in D_s} \ell_{Ftr}(E(h_2, \boldsymbol{x}); E(h_1, \boldsymbol{x}), k)$$
(18)

where $D_s := \{ \boldsymbol{x} \mid (\boldsymbol{x}, y) \in D \land s(\boldsymbol{x}, y; h_1, h_2) = 1 \}$ is the set of samples where h_1 and h_2 make correct predictions, and $\lambda \in \mathbb{R}_{\geq 0}$ is a hyperparameter.

Similar surrogate losses for other agreement metrics, i.e., rank-, sign-, and signedrank-agreements are defined to lower bound each of agreements with theoretical analyses in the following lemmas from Lemma 2 to Lemma 4.

Lemma 2. Let $I = \{(j, \operatorname{Argsort}(\operatorname{Abs}(e_1))_j) \mid j \in [k]\}$ be the set of tuples each (j,i) of which indicates that the i-th element of e_1 is the j-th largest value among $\{|e_{11}|,...,|e_{1d}|\}$, where Argsort returns the indices that would sort its input in *descending* order. Then we define $\ell_{\operatorname{Rnk}}$ as

$$\ell_{\text{Rnk}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)$$

$$:= \frac{1}{k} \sum_{(j,i) \in I} \max \left(0, \text{Sort}(\text{Abs}(\boldsymbol{e}_2)_{i=-\varepsilon})_j - |e_{2i}| + \varepsilon \right)$$
 (19)

where $a_{i=x} := [a_1, ..., a_{i-1}, x, a_{i+1}, ..., a_d]$ is a copy of a, whose i-th element is replaced to x. Then we have following inequality for any e_1, e_2 and k,

$$1 - \operatorname{RnkAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) \le \varepsilon^{-1} \ell_{\operatorname{Rnk}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)$$
 (20)

where Sort sorts its input in descending order.

Lemma 3. Let us define ℓ_{Sgn} as

$$\ell_{\operatorname{Sgn}}(\boldsymbol{e}_{2}; \boldsymbol{e}_{1}, k) := \frac{1}{k} \sum_{i \in \operatorname{TopFeat}(\boldsymbol{e}_{1}; k)} \max \left(0, \psi_{sign}(\boldsymbol{e}_{2}) - \operatorname{sgn}(\boldsymbol{e}_{1i}) \boldsymbol{e}_{2i} + \varepsilon\right)$$
(21)

where

$$\psi_{sign}(\mathbf{e}_2) := \begin{cases} \max_{i \notin \text{TopFeat}(\mathbf{e}_1; k)} |e_{2i}| & (k < d) \\ 0 & (\text{Otherwise}) \end{cases} . \quad (22)$$

Then we have following inequality for any e_1, e_2 and k,

$$1 - \operatorname{SgnAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) \le \varepsilon^{-1} \ell_{\operatorname{Sgn}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)$$
 (23)

Lemma 4. Let us define ℓ_{SgnRnk} with I defined in Lemma 2 as

$$\ell_{\operatorname{SgnRnk}}(\boldsymbol{e}_{2}; \boldsymbol{e}_{1}, k)$$

$$:= \frac{1}{k} \sum_{(j,i)\in I} \max\left(0, \operatorname{Sort}(\operatorname{Abs}(\boldsymbol{e}_{2})_{i=0})_{j} - \operatorname{sgn}(\boldsymbol{e}_{1i})\boldsymbol{e}_{2i} + \varepsilon\right).$$
(25)

Then we have following inequality for any e_1 , e_2 and k,

$$1 - \operatorname{SgnRnkAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) \le \varepsilon^{-1} \ell_{\operatorname{SgnRnk}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)$$
 (26)

Based on these lemmas, we define rank-agreement-, sign-agreement-, and signedrank-agreement-based BCXR (denoted by BCXR-Rnk, BCXR-Sgn, and BCXR-SgnRnk, respectively) with corresponding objectives $\widehat{L}_{\rm Rnk}, \widehat{L}_{\rm Sng}$ and $\widehat{L}_{\rm SgnRnk}$ as similar with feature-based BCXR, defined in Definition 2, by replacing $\ell_{\rm Ftr}$ in Eq. (18) by $\ell_{\rm Rnk}, \ell_{\rm Sgn}$ and $\ell_{\rm SgnRnk}$, respectively. Each of these are specialized objective to specifically improve the corresponding agreement metric.

3.3 Universal BCXR

We have devised surrogate loss functions to enhance the four agreement metrics. Furthermore, we propose a universal loss function that can effectively lower bound all agreement metrics regardless of the choice of k. Specifically, we utilize NormDisagree, which represents the Euclidean distance between the explanations, defined as follows:

NormDisagree
$$(e_1, e_2) := \|e_1 - e_2\|_2 = \sqrt{\sum_{i=1}^{d} (e_{1i} - e_{2i})^2}.$$
(27)

NormDisagree is differentiable and can be used as a loss function directly. For the validity of the use of NormDisagree, we establish the following lemma.

Lemma 5. Given any $e_1 \in \mathbb{R}^d$ and $k \in [d]$, assume that there exists $\delta > 0$ such that (a) $\left| |e_{1i}| - |e_{1j}| \right| \geq \sqrt{2}\delta$ holds for any $i \neq j \in \text{TopFeatures}(e_1, \max(k+1, d))$, and (b) additionally if k = d, $|e_{1i}| \geq \delta$ holds for any $i \in [d]$. Then the following inequality holds for any $e_2 \in \mathbb{R}^d$.

$$1 - \operatorname{SgnRnkAgr}(e_1, e_2, k) \le \delta^{-1} \operatorname{NormDisagree}(e_1, e_2)$$
 (28)

Note it is trivial by definition that feature agreement lower bounds both rank agreement and sign agreement, each of which lower bounds signed-rank agreement, i.e., the following inequality holds true for any e_1 , e_2 and k as

$$\operatorname{FtrAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2}) \geq \frac{\operatorname{RnkAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2})}{\operatorname{SgnAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2})} \geq \operatorname{SgnRnkAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2}). \tag{29}$$

By Eq. (29) and Lemma 5, NormDisagree multiplied with a constant δ^{-1} upper bounds any disagreement, i.e., one minus feat-sign-, rank-, and signedrank-agreement for any choice of k. Hence, regardless of which agreement metric is used to calculate BCX, NormDisagree can be used for the loss function responsible for the compatibility of BCX. Furthermore, the use of NormDisagree eliminates the burden to tune the hyperparameter ε .

Universal BCXR objective. Finally, we define our universal objective for our BCXR, which replaces $\ell_{\rm Ftr}$ in Eq. (18) by NormDisagree as follows.

$$\widehat{L}_{\text{Norm}}(h_2; D) := \frac{1}{|D|} \sum_{(\boldsymbol{x}, y) \in D} \ell(h_2(\boldsymbol{x}), y)$$

$$+ \lambda \frac{1}{|D_s|} \sum_{(\boldsymbol{x}, y) \in D_s} \text{NormDisagree}(E(h_1, \boldsymbol{x}), E(h_2, \boldsymbol{x})) \quad (30)$$

We refer to the minimization of Eq. (30) as BCXR-Norm.

We have established all of our BCXR methods and next investigate the empirical behaviour of BCXR for real-world data sets.

Table 1. Data set statistics.

Task	Data set	Samples	Features
Regresssion	space-ga	3107	6
	cadata	20640	8
	cpusmall	8192	12
	YearPredictionMSD	463715	90
Classification	cod-rna	59535	8
	phishing	11055	68
	a9a	32561	123
	w8a	49749	300

4 Experiments

We conduct experiments on real world data sets to verify the effectivity of the proposed objective function. The implementation is based on python with PyTorch [22] and scikit-learn [23]. All experiments are carried out on a computational server equipping four Intel Xeon Platinum 8260 CPUs with 192 logical cores in total and 1TB RAM.

4.1 Data set

We utilize four regression and four classification data sets obtained from LIBSVM [4]. We vary the number of input features from 6 to 300 in order to study the behavior of BCXR w.r.t. d. The data set statistics are presented in Table 1.

4.2 Setting

We randomly sample 2000 samples from each data set. The first 200 samples are used as D_1 and the first 1000 samples including D_1 are used as D_2 . The remaining 1000 samples are used for evaluation.

We utilize three-layer neural networks to model old and new models. The number of hidden units are 100, the activation function is ReLU [20], and we use batch normalization [9] after each of the activation layers. Each of the D_1 and D_2 is split into training and validation sets at an ratio of 80:20 and the validation set are used for early stopping [19]. The Adam optimizer with a learning rate of 0.01 and weight decay [8] of 1×10^{-4} are used for training. The maximum number of epochs is set to 200.

The old model h_1 is trained by a standard model training method, i.e., empirical risk minimization (ERM) with D_1 . New models are trained with our BCXR-Ftr, BCXR-Rnk, BCXR-Sgn, BCXR-SgnRnk, and BCXR-Norm. The comparison baselines are ERM, DM (BTC-aware retraining method [2]), and ABCD (ABCD-aware retraining method [17]). For DM and ABCD we vary their hyperparameter λ from 1×10^{-4} to 1×10^4 . Other hyperparameters for ABCD follow its author-defined values.

For our BCXR methods, we set $\varepsilon=1\times 10^{-3}$, and for regression tasks, we set the threshold τ to $1/|D_2|\sum_{(\boldsymbol{x},y)\in D_2}(h_1(\boldsymbol{x})-y)^2$. The hyperparameter λ is set from 1×10^{-4} to 1×10^2 .

The evaluation metrics are the standard loss (i.e., mean squared error (MSE) for regression tasks and mean 0-1 loss for classification tasks), BTC and empirical BCX scores with feature-, rank-, sign-, signedrank-, and norm-agreement metrics on evaluation data⁶. The value of k set to 5 following the existing study [13], and the same k is used for our BCXR methods as well. We repeat our experiments for 30 times for each data set and each methods with each settings of λ , and report the average scores for each of the evaluation metrics.

These assumptions may be satisfied by a proper feature engineering, e.g, both features with the same effects to the prediction and ones with constantly zero effects to the prediction can be removed from features.

⁶ Since NormDisagree is a **dis**agreement metric, we use norm-agreement as $1 - \text{NormDisagree}(e_1, e_2)$, and we use it in Eq. (7) for our evaluation.

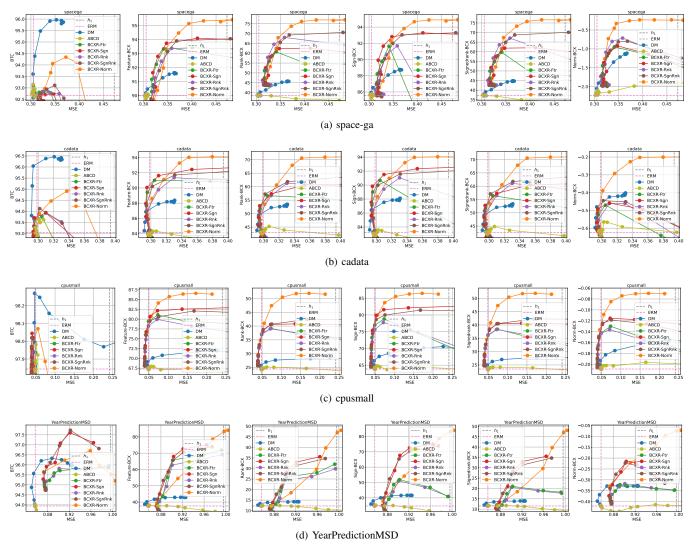


Figure 1. Trade-off for regression data sets. Horizontal axes represents MSE (the lower the better, \leftarrow) and vertical axes represents each of BTC and BCXs with different agreement metrics (the higher the better, \uparrow). In general, points located in the upper left region of each figure indicate better results compared to points in the lower right region. The grey dashed vertical lines indicate the MSE achieved by old models. The pink dashed vertical and horizontal lines represent the MSE and backward compatibility scores achieved by the ERM. Retraining methods that take backward compatibility into account are expected to perform better MSE than the old models (up to the grey dashed lines) and better compatibility than ERM (up to the pink horizontal lines). Since this is a multi-objective optimization problem, the results on the Pareto fronts are considered effective in finding better trade-offs between MSE and backward compatibility scores.

4.3 Results

The results are presented in Figure 1 for regression datasets and Figure 2 for classification datasets, where the backward compatibility scores are plotted against the loss (MSE and 0-1 loss). The explanation of the plots are as follows; the vertical grey dashed lines indicate the loss values of the old models. The intersections of the horizontal and vertical pink dashed lines represent the losses and backward compatibility scores of ERM. Therefore, backward-compatibility-aware retraining methods are expected to be positioned above the horizontal pink dashed lines and to the left of the vertical grey dashed lines. Each of our methods and baselines follows a bottom-to-top pattern as the parameter λ increases. For example, the result with a value of $\lambda = 1 \times 10^{-4}$ is located near the pink dashed lines, while results with larger λ values are found in the upper parts of each figure. Since there are often trade-offs between the loss and BTC and BCX metrics, and the importance of compatibility varies depending on the

application, it is difficult to determine the best retraining method and parameter value of λ . However, the methods that form Pareto fronts in the figures are generally considered to be effective. We now discuss the details of the regression and classification results.

4.3.1 Results for regression tasks.

The BCXR-Norm method forms part of the Pareto fronts in all BCX-based plots for regression data sets, demonstrating the universality of BCXR. This result confirms the findings discussed in Section 3.3. Additionally, we observe that for results with smaller MSE, other BCXR methods (BCXR-Ftr, -Rnk, -Sgn, and BCXR-SgnRnk) offer comparable or better trade-offs than BCXR-Norm. This is particularly evident in the case of YearPredictionMSD, as NormDisagree considers all 90 features, while only the top 5 features significantly influence the agreement scores. As a result, BCXR methods other than BCXR-Norm effectively consider these five features and conse-

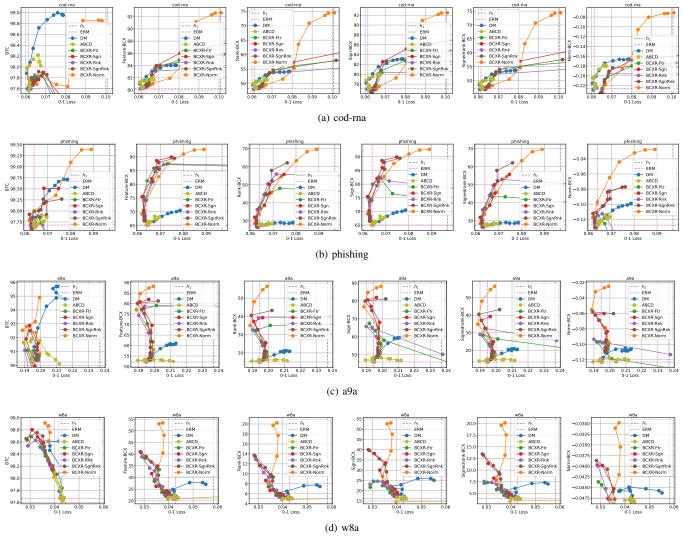


Figure 2. Trade-off for classification data sets. Explanation of figures follow Figure 1.

quently optimize new models more efficiently. Furthermore, among the BCXR results excluding BCXR-Norm, both BCXR-Sgn and BCXR-SgnRnk consistently demonstrate better trade-offs in all BCX scores compared to BCXR-Ftr and BCXR-Rnk. This suggests that enforcing constraints based on the signs of the explanations is crucial for maintaining consistent explanations of the top-k features.

Among the baselines, DM exhibits some improvements in the BCX scores as λ increases, and it partially contributes to the formation of the front lines. However, the degree of improvement is relatively limited. Interestingly, in the case of YearPredictionMSD, our BCXR methods outperform DM in terms of the BTC score. Additionally, since ABCD focuses on enhancing conditional losses, it does not exhibit better trade-offs in our evaluation. These findings further underscore the superiority of BCXR in providing consistent explanations during model updates.

4.3.2 Results for classification tasks.

The results obtained from the cod-rna and phishing exhibit similar patterns to those observed in regression tasks. Specifically, when the 0-1 losses are small, agreement-based BCXR methods yield better results. Conversely, BCXR-Norm demonstrates better trade-offs

when the losses are large. Interestingly, for a9a and w8a, datasets with large numbers of features (123 and 300, respectively), which are significantly larger than the value of k=5, our BCXR methods not only improve the BCX scores but also enhance BTC and reduce 0-1 losses. These outcomes suggest that the inclusion of BCX as a constraint potentially leads to the improved optimization of training new models. Consequently, our BCXR methods are proven to be effective and can be applied across a wider range of applications, extending beyond the sole purpose of maintaining explanation compatibility.

5 Conclusion

In this study, we have introduced BCX as a novel approach to assess the consistency of explanations in model updates. Then, to overcome the challenge of non-differentiability in the agreement metrics, we propose differential surrogate losses that possess theoretical validity for substitution. Building upon this, we have proposed BCXR, a BCX-aware retraining method, which leverages the surrogate losses to achieve high BCX scores as well as high predictive performances. Furthermore, we have presented a universal variant of BCXR that improves all agreement metrics simultaneously. By conducting experiments on eight real-world datasets, we have demonstrated that

BCXR offers superior trade-offs between BCX scores and predictive performances, which underscores the effectiveness of our proposed approaches. Overall, our study contributes to the advancement of trustworthy and responsible MLOps by providing a method to assess and enhance the consistency of explanations in model updates.

References

- K. Aas, M. Jullum, and A. Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.
- [2] G. Bansal, B. Nushi, E. Kamar, D. S. Weld, W. S. Lasecki, and E. Horvitz. Updates in human-ai teams: Understanding and addressing the performance/compatibility tradeoff. In AAAI Conference on Artificial Intelligence, 2019.
- [3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
 [4] C.-C. Chang and C.-J. Lin. Libsvm data: Classification, regression, and
- [4] C.-C. Chang and C.-J. Lin. Libsvm data: Classification, regression, and multi-label. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 2023. Accessed: 2023-09-25.
- [5] I. Covert and S.-I. Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Arti*ficial Intelligence and Statistics, volume 130 of Machine Learning Research, pages 3457–3465. PMLR, 2021.
- [6] M. Flora, C. Potvin, A. McGovern, and S. Handler. Comparing explanation methods for traditional machine learning models part 1: An overview of current methods and quantifying their disagreement. arXiv preprint, 2022.
- [7] M. Flora, C. Potvin, A. McGovern, and S. Handler. Comparing explanation methods for traditional machine learning models part 2: Quantifying model explainability faithfulness and improvements with dimensionality reduction. arXiv preprint, 2022.
- [8] S. J. Hanson and L. Y. Pratt. Comparing biases for minimal network construction with back-propagation. In Advances in Neural Information Processing Systems, page 177–185, 1988.
- [9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Con*ference on Machine Learning, volume 37, page 448–456, 2015.
- [10] N. Jethani, M. Sudarshan, I. C. Covert, S.-I. Lee, and R. Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022.
- [11] G. Jiang, F. Zhuang, B. Song, T. Zhang, and D. Wang. Prishap: Prior-guided shapley value explanations for correlated features. In ACM International Conference on Information and Knowledge Management, page 955–964, 2023.
- [12] D. Kreuzberger, N. Kühl, and S. Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 11: 31866–31879, 2023.
- [13] S. Krishna, T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu, and H. Lakkaraju. The disagreement problem in explainable machine learning: A practitioner's perspective. *arXiv preprint*, 2022.
 [14] I. Kumar, C. Scheidegger, S. Venkatasubramanian, and S. Friedler.
- [14] I. Kumar, C. Scheidegger, S. Venkatasubramanian, and S. Friedler. Shapley residuals: Quantifying the limits of the shapley value for explanations. In *Advances in Neural Information Processing Systems*, volume 34, pages 26598–26608, 2021.
- [15] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural In*formation Processing Systems, 2017.
- [16] S. M. Lundberg, G. G. Erion, and S. Lee. Consistent individualized feature attribution for tree ensembles. *arXiv* preprint, 2018.
- [17] R. Matsuno and K. Sakuma. A robust backward compatibility metric for model retraining. In ACM International Conference on Information and Knowledge Management, pages 4190–4194, 2023.
- [18] C. Molnar. Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. Second edition, 2022.
- [19] N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. In Advances in Neural Information Processing Systems, page 630–637, 1989.
- [20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, page 807–814, 2010.

- [21] M. Neely, S. F. Schouten, M. J. R. Bleeker, and A. Lucic. Order in the court: Explainable AI methods prone to disagreement. arXiv preprint, 2021.
- [22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, 2019.
- [23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [24] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 1135–1144, 2016.
- [25] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In AAAI Conference on Artificial Intelligence, volume 32, 2018.
- [26] P. Ruf, M. Madan, C. Reich, and D. Ould-Abdeslam. Demystifying mlops and presenting a recipe for the selection of open-source tools. *Applied Sciences*, 2021.
- [27] W. Saeed and C. Omlin. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. Knowledge-Based Systems, 263:110273, 2023. ISSN 0950-7051.
- [28] T. Sakai. A generalized backward compatibility metric. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 1525–1535, 2022.
- [29] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, feb 2020. ISSN 0920-5691.
- [30] L. S. Shapley. A value for n-person games. In *Contributions to the Theory of Games*, volume 2, pages 307–317, 1953.
- [31] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *International Con*ference on Machine Learning, page 3145–3153, 2017.
- [32] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint, 2014.
- [33] M. Srivastava, B. Nushi, E. Kamar, S. Shah, and E. Horvitz. An empirical analysis of backward compatibility in machine learning systems. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 2020.
- [34] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, page 3319–3328, 2017.
- [35] G. Symeonidis, E. Nerantzis, A. Kazakis, and G. A. Papakostas. Mlops - definitions, tools and challenges. In 2022 IEEE 12th Annual Computing and Communication Workshop and Conference, pages 0453–0460, 2022.
- [36] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, and G. Vessio. Mlops: A taxonomy and a methodology. *IEEE Access*, 10: 63606–63618, 2022.
- [37] G. Van den Broeck, A. Lykov, M. Schleich, and D. Suciu. On the tractability of shap explanations. *J. Artif. Int. Res.*, 74, sep 2022.
 [38] Z. Wang, C. Huang, Y. Li, and X. Yao. Multi-objective feature attribu-
- [38] Z. Wang, C. Huang, Y. Li, and X. Yao. Multi-objective feature attribution explanation for explainable machine learning. ACM Trans. Evol. Learn. Optim., 4(1), feb 2024.
- [39] S. Yan, Y. Xiong, K. Kundu, S. Yang, S. Deng, M. Wang, W. Xia, and S. Soatto. Positive-congruent training: Towards regression-free model updates. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14294–14303, 2021.

Appendix

We provide the detailed proofs in Appendix A and supplemental information on our numerical experiments in Appendix B. Moreover, we discuss the limitation of our method in Appendix C.

A Proofs

A.1 Proof of Lemma 1

The inequality is trivial when $\operatorname{FtrAgr}(e_1, e_2; k) = 1$. Suppose $\operatorname{FtrAgr}(e_1, e_2; k) = 1 - a/k$ with $a \in \{1, ..., d - k\}$. We have

$$\frac{1 - \operatorname{FtrAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)}{\ell_{\operatorname{Ftr}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)} = \frac{a}{k} \frac{1}{\ell_{\operatorname{Ftr}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)}$$
(A.1)

$$\leq \frac{a}{k} \sup_{\boldsymbol{e}_2: \operatorname{FtrAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) = 1 - \frac{a}{k}} \frac{1}{\ell_{\operatorname{Ftr}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)}$$
(A.2)

$$\leq \frac{a}{k} \frac{1}{\inf_{\boldsymbol{e}_{2}:\operatorname{FtrAgr}(\boldsymbol{e}_{1},\boldsymbol{e}_{2};k)=1-\frac{a}{k}} \ell_{\operatorname{Ftr}}(\boldsymbol{e}_{2};\boldsymbol{e}_{1},k)}$$
(A.3)

$$\leq \frac{a}{k} \frac{k}{a\varepsilon} = \varepsilon^{-1} \tag{A.4}$$

where $\ell_{\mathrm{Ftr}}(\boldsymbol{e}_2;\boldsymbol{e}_1,k)$ is infimized when $|e_{2i}|=\psi_{feat}(\boldsymbol{e}_2)$ for each $i\in\{i\in\mathrm{TopFeat}(\boldsymbol{e}_1;k)\mid |e_{2i}|\leq\psi_{feat}(\boldsymbol{e}_2)\}$. At this time, $\ell_{\mathrm{Ftr}}(\boldsymbol{e}_2;\boldsymbol{e}_1,k)$ is at least $a\varepsilon/k$. Since Eq. (A.4) does not depend on $a,\varepsilon^{-1}\ell_{\mathrm{Ftr}}(\boldsymbol{e}_2;\boldsymbol{e}_1,k)$ always bounds $(1-\mathrm{FtrAgr}(\boldsymbol{e}_1,\boldsymbol{e}_2;k))$ from above, concluding the proof.

A.2 Proof of Lemma 2

The proof is almost identical with Lemma 1. When $RnkAgr(e_1, e_2) = 1$, we have $\ell_{Rnk}(e_1, e_2; k) = 0$ and the inequality is trivial. Suppose $RnkAgr(e_1, e_2; k) = 1 - a/k$ with $a \in \{1, ..., k\}$. We have

$$\frac{1 - \operatorname{RnkAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2}; k)}{\ell_{\operatorname{Rnk}}(\boldsymbol{e}_{2}; \boldsymbol{e}_{1}, k)} \leq \frac{a}{k} \frac{1}{\inf_{\boldsymbol{e}_{2}: \operatorname{RnkAgr}(\boldsymbol{e}_{1}, \boldsymbol{e}_{2}; k) = 1 - \frac{a}{k}} \ell_{\operatorname{Rnk}}(\boldsymbol{e}_{2}; \boldsymbol{e}_{1}, k)} \leq \frac{a}{k} \frac{k}{a\varepsilon} = \varepsilon^{-1}$$
(A.5)

where the infimum of $\ell_{\text{Rnk}}(e_2; e_1, k)$ is lower bounded by $a\varepsilon/k$, which may be achieved when $|e_{2i}| = \text{Sort}(\text{Abs}(e_2)_{i=-\varepsilon})_j$ for some a pairs of (j, i) in I, concluding the proof.

A.3 Proof of Lemma 3

The proof is almost identical with Lemma 1. The inequality is trivial when $\operatorname{SgnAgr}(e_1, e_2; k) = 1$. Suppose $\operatorname{SgnAgr}(e_1, e_2; k) = 1 - a/k$ with $a \in \{1, ..., k\}$.

$$\frac{1 - \operatorname{SgnAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)}{\ell_{\operatorname{Sgn}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)} \le \frac{a}{k} \frac{1}{\inf_{\boldsymbol{e}_2: \operatorname{SgnAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) = 1 - \frac{a}{k}} \ell_{\operatorname{Sgn}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)}$$
(A.6)

(A.7)

When $\psi_{sign}(\boldsymbol{e}_2) = 0$,

$$\ell_{\operatorname{Sgn}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k) = \frac{1}{k} \sum_{i \in \operatorname{TopFeat}(\boldsymbol{e}_1; k)} \max \left(0, \varepsilon - \operatorname{sgn}(\boldsymbol{e}_{1i}) \boldsymbol{e}_{2i} \right)$$
(A.8)

Based on the fact that $\operatorname{sgn}(e_{1i}) \neq \operatorname{sgn}(e_{2i})$ for a indices, we have $\ell_{\operatorname{Sgn}}(e_2; e_1, k) \geq a\varepsilon/k$. For the cases when $\psi_{sign}(e_2) > 0$, $\ell_{\operatorname{Sgn}}(e_2; e_1, k)$ is infimized when $\psi_{sign}(e_2) = \operatorname{sgn}(e_{1i})e_{2i}$ for some a indices. For both cases, the infimum is lower bounded by $a\varepsilon/k$ and we have

$$\frac{1 - \operatorname{SgnAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)}{\ell_{\operatorname{Sgn}}(\boldsymbol{e}_2; \boldsymbol{e}_1, k)} \le \varepsilon^{-1}$$
(A.9)

which concludes the proof.

A.4 Proof of Lemma 4

The proof is trivial by the proofs of Lemma 2 and Lemma 3.

A.5 Proof of Lemma 5

The inequality holds true if $\operatorname{SgnRnkAgr}(e_1, e_2; k) = 1$. Suppose $\operatorname{SgnRnkAgr}(e_1, e_2; k) < 1$. We have

$$\frac{1 - \operatorname{SgnRnkAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)}{\operatorname{NormDisagree}(\boldsymbol{e}_1, \boldsymbol{e}_2)} \le \frac{1}{\inf_{\boldsymbol{e}_2: \operatorname{SgnRnkAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k) < 1} \operatorname{NormDisagree}(\boldsymbol{e}_1, \boldsymbol{e}_2)}$$
(A.10)

The infimum of NormDisagree(e_1, e_2) under SgnRnkAgr($e_1, e_2; k$) < 1 is achievable when exactly one of the following two proposition holds:

- 1. for $(i,j) = \underset{(i \neq j \in \text{TopFeatures}(e_1, \max(k+1, d)))}{\arg\min} ||e_{1i}| |e_{1j}||, |e_{2i}| = |e_{2j}| = (|e_{1i}| + |e_{1j}|)/2 \text{ and for any other } t \in [d] \setminus \{i, j\}, e_{2t} = e_{1t}.$ 2. k = d and for $i = \underset{j}{\arg\min} |e_{1i}|, e_{2i} = 0$ and for any other $t \in [d] \setminus \{i\}, e_{2t} = e_{1t}.$
- When the first proposition holds, NormDisagree is lower bounded as

NormDisagree
$$(e_1, e_2) = \sqrt{\left(|e_{1i}| - \frac{|e_{1i}| + |e_{1j}|}{2}\right)^2 + \left(|e_{1j}| - \frac{|e_{1i}| + |e_{1j}|}{2}\right)^2} = \sqrt{\frac{\left(|e_{1i}| - |e_{1j}|\right)^2}{2}} \ge \sqrt{\frac{\left(\sqrt{2}\delta\right)^2}{2}} = \delta,$$
 (A.11)

and for the second proposition, NormDisagree is bounded as

NormDisagree
$$(e_1, e_2) = \sqrt{(|e_{1i}| - 0)^2} = |e_{1i}| \ge \delta$$
 (A.12)

Hence, under the condition that $\operatorname{SgnRnkAgr}(e_1, e_2; k) < 1$, $\operatorname{NormDisagree}(e_1, e_2)$ is no less than δ . By Eq. (A.10), we have

$$\frac{1 - \operatorname{SgnRnkAgr}(\boldsymbol{e}_1, \boldsymbol{e}_2; k)}{\operatorname{NormDisagree}(\boldsymbol{e}_1, \boldsymbol{e}_2)} \le \delta^{-1}, \tag{A.13}$$

which conclude the proof.

B Sensitivity against λ

While we have provided BCX-against-loss plots in our main paper, we have also included additional plots that illustrate the results against the hyperparameters λ for each data set and each metric. The plots are presented in Figure B.1 for regression tasks and Figure B.2 for classification tasks. The results clearly indicate that the agreement-based BCXR methods are highly sensitive to changes in the value of λ . For example, when λ is set to 10^2 for the space-ga data set, the mean squared error (MSE) of the BCXR methods often exceeds the MSE of the old models. A lower MSE than the old model is crucial for successful model updates, and therefore, setting λ to a large value can adversely affect the training of a new model. However, based on these findings, we can conclude that setting λ to a value between 1 and 10 would generally yield good results for most data sets. Therefore, when applying our BCXR method in practical tasks, it is recommended to tune λ within this range to ensure the training of a suitable model in MLOps.

C Limitation

In this section, we discuss the possible limitation of BCX and BCXR.

Intractable computational cost. BCX and BCXR may suffer from intractable computational costs due to the high complexity of the explanation methods they employ. For example, in our experiments, we utilize SHAP. However, as the number of features d increases, SHAP becomes increasingly computationally intensive. Therefore, when dealing with a very large number of features (e.g., in image and text classification), it is preferable to approximate the SHAP calculation or to use more lightweight explanation methods, such as gradient-based methods [29, 32, 34]. Although these alternatives may sacrifice some of the validity of the explanation, they provide a more computationally feasible solution.

Difficulty of requirement design. Although BCX quantitatively assesses the consistency of explanations, determining the practical requirements for BCX in real-world applications can be challenging. Both BCX and BCXR are mathematically defined metrics, which means that outliers, abnormal explanations, and distribution shifts do not affect their computation, as is the case with any BTC-related scores. However, we observe that these scores might be practically meaningless in certain contexts. For example, if a pre-update model is trained before a severe distribution shift, aligning a post-update model with the pre-update model may not be reasonable or useful. The meaningfulness of BCX and other BTC-related scores depends on the compatibility requirements for the ML system. Unfortunately, these requirements cannot be uniquely determined from a theoretical perspective alone. Therefore, data scientists and customers need to collaboratively discuss the detailed requirements to determine the necessary level of compatibility for different situations. Based on these requirements, it may be necessary to remove outliers and abnormal explanations from the computation of BCX. Although determining these requirements may limit the practical usefulness of BCX and BCXR, this challenge is common in the broader fields of explainability, fairness, and privacy in machine learning.

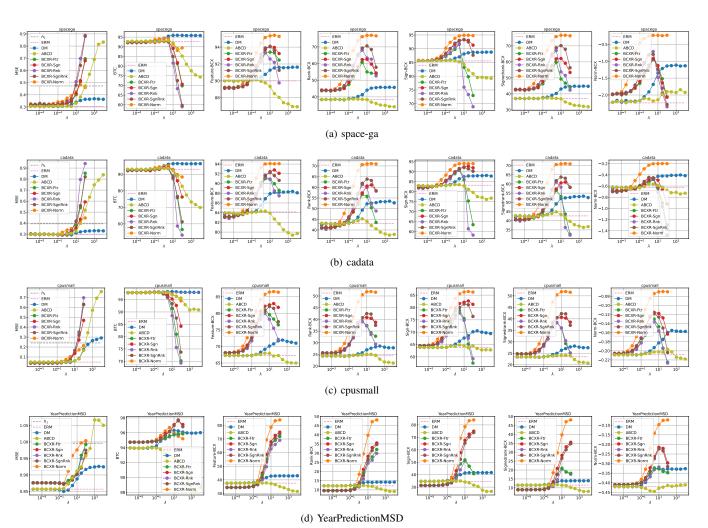


Figure B.1. Sensitivity plot of λ for regression data sets.

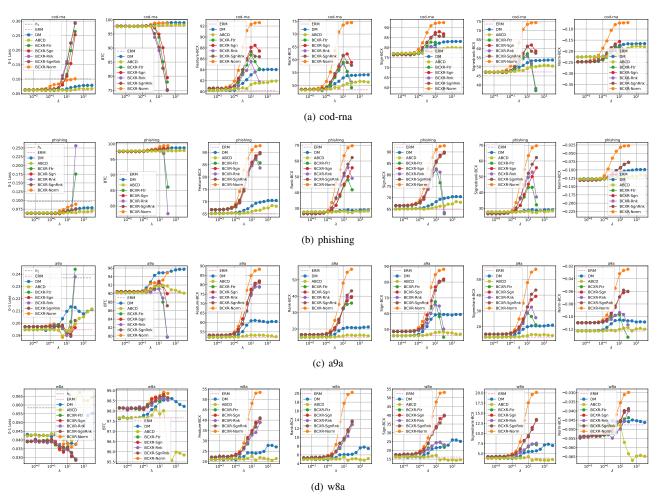


Figure B.2. Sensitivity plot of λ for classification data sets.