Explanations Go Linear: Post-hoc Explainability for Tabular Data with Interpretable Meta-Encoding

Simone Piaggesi[1], Riccardo Guidotti[1,2], Fosca Giannotti[3], Dino Pedreschi[1]
[1]University of Pisa, [2]ISTI-CNR, [3]Scuola Normale Superiore - Pisa, Italy
simone.piaggesi@di.unipi.it, {*riccardo.guidotti*, *dino.pedreschi*}*@unipi.it*, *fosca.giannotti@sns.it*

***Abstract*—Post-hoc explainability is essential for understanding black-box machine learning models. Surrogate-based techniques are widely used for local and global model-agnostic explanations but have significant limitations. Local surrogates capture non-linearities but are computationally expensive and sensitive to parameters, while global surrogates are more efficient but struggle with complex local behaviors. In this paper, we present ILLUME, a flexible and interpretable framework grounded in representation learning, that can be integrated with various surrogate models to provide explanations for any black-box classifier. Specifically, our approach combines a globally trained surrogate with instance-specific linear transformations learned with a meta-encoder to generate both local and global explanations. Through extensive empirical evaluations, we demonstrate the effectiveness of ILLUME in producing feature attributions and decision rules that are not only accurate but also robust and computationally efficient, thus providing a unified explanation framework that effectively addresses the limitations of traditional surrogate methods.**

I. INTRODUCTION

In eXplainable AI (XAI), post-hoc explanations seek to clarify how machine learning (ML) black-box models make decisions. Commonly studied explanations are feature attribution, decision rules, and counterfactuals [1], [2]. Among post-hoc techniques, surrogate explainers are largely adopted given their effectiveness in performing complex model distillation [3]. Global surrogates, like TREPAN [4] and related approaches [5], involve training a single interpretable model (e.g., linear regression or decision tree) to replicate the behavior of the target black-box across the entire dataset, providing broad insight but often missing complex non-linear patterns. Instead, local surrogates, such as LIME [6] and LORE [7], build an interpretable model within a (typically synthetically generated) neighborhood of each specific instance, thus better approximating local non-linearities of decision boundaries. Despite being pivotal for explainability [1], [8], existing local explainers face several limitations [9], [10], including instability [11], sensitivity to hyperparameters [12], computational inefficiencies [13] and misleading behaviors [14]. These drawbacks are primarily attributed to the biases and sampling variability introduced by the neighborhood generation [15], [16]. In this work, we bridge the gap between local and global surrogate methods by providing a comprehensive post-hoc explanation framework that: *(i)* overcomes the restrictions of global surrogates regarding their generalization capabilities, *(ii)* addresses the methodological issues of local surrogates going beyond neighborhood sampling, and *(iii)* supports any type of explanation format.

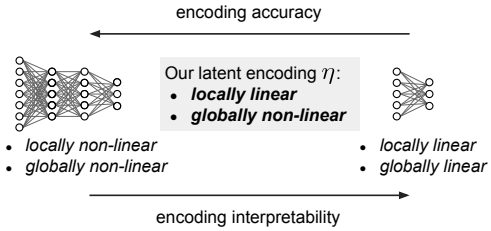


Fig. 1: Accuracy-interpretability trade-off for neural encoding.

Our proposed framework redefines the post-hoc explanation problem by operating in a feature space distinct from the input. Specifically, it transforms the data into a latent representation [17] that enhances the expressiveness of global surrogate explanations. Typically, latent mappings [18] are produced by highly non-linear operations, such as neural networks (NNs), and present uninterpretable features, rendering any surrogate explanations that rely on these features ineffective. Conversely, creating interpretable embeddings is inherently challenging [19] and requires balancing accuracy with transparency [20]. Most straightforward interpretable encodings are achieved through linear transformations, e.g., implemented as single-layer NNs [21], [22] without non-linear activations. As in Figure 1, linear NNs (on the right) can be seen as interpretable considering the single-layer structure and the additive nature of output neurons when activation functions are not used. However, at the same time, they are also too restrictive due to their overly simplified architecture [23]. Specifically, they provide interpretable transformations both globally, since the single-layer network defines a linear function with a unique weights matrix W, and locally, since it applies linearly on each specific instance $\mathbf{x}$, resulting in the latent representation $\mathbf{z} = W\mathbf{x}$. In contrast, deep NNs achieve greater accuracy but are not interpretable, because applying $NN(\mathbf{x})$ involves obscure complex and stacked (global and local) nonlinear operations. Neural Additive Models (NAMs) [24] are possible candidates for balancing this trade-off. NAMs define the mapping as a sum of non-linear functions $NN_j(x_j)$, for each feature j,

This work has been partially supported by the European Community Horizon 2020 programme under the funding scheme ERC-2018-ADG G.A. 834756 *XAI: Science and technology for the eXplanation of AI decision making*, and the NextGenerationEU programme under the funding schemes: PNRR-PE-AI (M4C2, investment 1.3, line on AI) *FAIR* (Future Artificial Intelligence Research), and "*SoBigData.it* - Strengthening the Italian RI for Social Mining and Big Data Analytics" - Prot. IR0000013, and by the Italian Project Fondo Italiano per la Scienza FIS00001966 MIMOSA.

possibly including feature interactions $NN_{jj'}(x_j, x_{j'})$ [25]. NAMs are human-readable by looking at visual plots of NN_js and $NN_{jj'}$s [24]. However, assuming only pairwise interactions, NAMs are restrictive and higher-order extensions lose interpretability and scalability.

As an alternative solution to balance the trade-off, we introduce **locally linear transformations** as non-linear maps η, such that $\eta(\mathbf{x}) = W(\mathbf{x})\mathbf{x}$, where $W(\mathbf{x})$ is a neural function returning a specific weight matrix once applied to any instance $\mathbf{x}$. We see that the output of the function η is linear when applied on instance $\mathbf{x}$, without being a linear model as a whole, i.e., globally non-linear. Intuitively, η acts as a *meta*-encoder, returning for each instance a proper linear encoding used to transform the instance itself. In combination with a global surrogate model, such meta-encoding approach can be remarkably effective. Specifically, we found that *global explanations deriving from latent space surrogates can be turned into local explanations in the original space if the latent mapping is done with locally linear transformations*. From a certain point of view, our proposal is inspired by *HyperNetworks* [26], a class of NNs used to generate the weights for other NNs. However, in our case, the generated NN is single-layer perceptron without biases.

To give an intuition, we examine the case of additive feature importance methods like LIME [6] and SHAP [13]. Let $\mathbf{x} = \{x_1, \ldots, x_m\}$ be one input to a black-box classifier $b(\cdot)$, where x_j is the value of j-th feature in $\mathbf{x}$. Local additive explainers fit an instance-specific linear model $b(\mathbf{x}) \approx \sum_j \psi_j(\mathbf{x}) x_j$, where each learned weight $\psi_j(\mathbf{x})$ is the contribution of feature j to the opaque prediction for instance $\mathbf{x}$. On the other hand, analogous global surrogates enable an additive expression of the log-odds as $\log \frac{b(\mathbf{x})}{1-b(\mathbf{x})} \approx \sum_j \beta_j x_j$ [27]. Contrarily to local methods, here the weights $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_m\}$ do not depend on the specific $\mathbf{x}$, failing to describe black-box behavior locally. In our approach, we fit a latent interpretable surrogate, e.g., the linear regressor $\sum_r \beta_r^L z_r$, whose variables are obtained through locally linear maps, i.e., $z_r = \sum_j W_{j,r}(\mathbf{x}) x_j$. Because W is a linear operation depending on $\mathbf{x}$, the latent global weights can be pulled back to the original features $\log \frac{b(\mathbf{x})}{1-b(\mathbf{x})} \approx \sum_j (W(\mathbf{x})^\top \cdot \boldsymbol{\beta}^L)_j x_j$, yielding local attribution scores valid in the input space. Similarly, the illustrated approach can be applied to any other surrogate explanations, including factual and counterfactual rules returned by LORE [7] or decision trees adopted as global surrogates like in TREPAN [4].

Based on the idea of locally linear transformations, we introduce ILLUME, an Interpretable individuaL Latent neUral Mapping for Explainability. ILLUME is based on latent space meta-encoding designed to guarantee desirable explanation properties. ILLUME trains a regularized meta-encoder that maps inputs into a latent space via locally linear transformations. Then, with the resulting embeddings, it fits a post-hoc surrogate model to globally approximate any target black-box system. This design enables to generate local explanations from global surrogate logic, achieving the precision of local explainers, while maintaining the efficiency of global surrogates. Also, the latent encoding is agnostic with respect to the surrogate model, supporting different types of explanations. With a wide range of experiments on tabular data, we show that ILLUME is able to generate local explanations as feature importance and decision rules, leading to more accurate, robust, faithful, and efficient explanations than state-of-the-art methods. The rest of the paper is organized as follows. After reviewing related literature in Section II, we describe ILLUME in Section III. In Section IV, we present the experimental results. Finally, Section V summarizes our contributions and outlines potential directions for future research.

II. Related Works

Surrogate explainability methods approximate black-box model predictions using simple interpretable models [8]. Global surrogates aim to capture the overall decision logic of the black-box through model distillation [4], [5]. On the other hand, local surrogates focus on specific model decisions. LIME [6] computes feature importance using linear models on locally sampled neighborhoods, while LORE [7] extracts logic rules with locally trained decision trees, enhancing neighborhood generation with genetic algorithms. Other works connecting game theory with local explanations, such as SHAP [28], enable model-agnostic estimation of Shapley values with local surrogate models. Global surrogates can be more efficient than local ones but they might miss complex non-linear relationships of the black-box models [4], [5]. In contrast, local surrogates better capture non-linear decision boundaries but can be unstable [11], sensitive to hyperparameters [12], and computationally demanding [13].

Various XAI methods train surrogate models by leveraging feature projection techniques [29]. These approaches are typically used to visualize high-dimensional data by optimizing low-dimensional representations. Basic methods such as PCA and MDS [30] were designed to preserve the overall structure of the data. Later, more sophisticated techniques such as ISOMAP and LLE [31] were developed to capture local relationships often overlooked by global techniques. Recently, algorithms like T-SNE and UMAP [29] have gained widespread popularity across scientific disciplines due to their ability to efficiently maintain local complexities and non-linear patterns. However, embeddings produced by these techniques are opaque, preventing their direct usage in XAI, as they rely on complex, non-linear transformations. Consequently, there is growing research interest in understanding latent representations [32], or learning inherently interpretable ones [21], [22].

Representation learning involves techniques for automatically deriving feature encoding functions from data [17]. It has become a fundamental component of NN-based models, enabling a wide range of tasks including generative methods [33], classification [34], and regression [24]. In XAI, latent space methods usually employ deep architectures like auto-encoders, to generate explanations such as exemplars [35] and counterfactuals [36], without looking at the transparency of the entire process. Alternatively, interpretable latent spaces have been proposed [21], [22] to facilitate counterfactual search, though

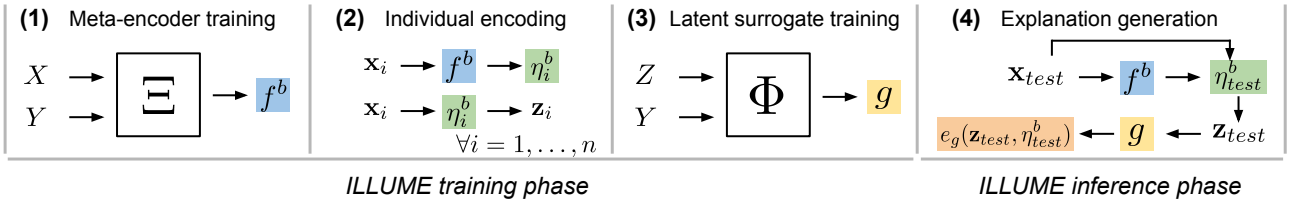


Fig. 2: ILLUME steps: **(1)** the meta-encoder f^b is trained using input instances X and black-box decisions Y; **(2)** f^b generates specific encoding functions η_i^b to individually map instances $\mathbf{x}_i$ into latent representations $\mathbf{z}_i$; **(3)** the set of latent vectors Z is used to train a surrogate model g for imitating the black-box $Y = b(X)$; **(4)** given a test instance $\mathbf{x}_{test}$, it is mapped into $\mathbf{z}_{test}$ with η_{test}^b obtained by f^b, then the explanation is obtained with $e_g(\mathbf{z}_{test}, \eta_{test}^b)$ by combining the surrogate logic g with local mapping η_{test}^b. Training algorithms are marked with white squares. Learned functions are marked with different colored boxes.

TABLE I: Common symbols and functions.

Symbol	Description
$\mathcal{X}, \mathcal{Z}, \mathcal{W}, \mathcal{Y}, \mathcal{E}$ X, Z, W, Y, E	continuous spaces and discrete subsets, respectively for: input instances, latent embeddings, input-to-latent transformations, black-box decisions and explanations.
$\mathbf{x}_i$, $\mathbf{x}_{:,j}$, $x_{i,j}$ $\mathbf{z}_i$, $\mathbf{z}_{:,j}$, $z_{i,j}$	original and latent vectors of i-th instance, vectors with j-th feature for all instances, original and latent j-th feature of i-th instance.
W_i, $W_{i,:,r}$, $W_{i,j,r}$	i-th transformation matrix, r-th column of i-th matrix, (j,r)-th entry of i-th matrix.
$b : \mathcal{X} \to \mathcal{Y}$	black-box classifier $b(\mathbf{x}_i)$
$g : \mathcal{Z} \to \mathcal{Y}$	latent surrogate classifier $g(\mathbf{z}_i)$
$e^b : \mathcal{X} \to \mathcal{E}$	black-box local explainer $e^b(\mathbf{x}_i)$
$f^b : \mathcal{X} \to \mathcal{W}$	meta-encoder function $f^b(\mathbf{x}_i) = W_i^b$
$\eta_i^b : \mathcal{X} \to \mathcal{Z}$	embedding function $\mathbf{z}_i = \eta_i^b(\mathbf{x}_i) = W_i^b \mathbf{x}_i$ (linear application of f^b output on instance $\mathbf{x}_i$)
$e_g : \mathcal{Z} \times \mathcal{W} \to \mathcal{E}$	explanation generator function $e_g(\mathbf{z}_i, \eta_i^b)$
$\Xi : (\mathcal{X} \times \mathcal{Y}) \to (\mathcal{X} \to \mathcal{W})$	meta-encoder training function on $(\mathbf{x}_i, b(\mathbf{x}_i)) \in \mathcal{X} \times \mathcal{Y}$
$\Phi : (\mathcal{Z} \times \mathcal{Y}) \to (\mathcal{Z} \to \mathcal{Y})$	surrogate training function on $(\mathbf{z}_i, b(\mathbf{x}_i)) \in \mathcal{Z} \times \mathcal{Y}$

their expressive power is limited by global linear mappings. Moreover, approaches that jointly train neural encoders with explainers have been developed to create self-interpretable models [37], offering an alternative to post-hoc, model-agnostic explainability methods. While these self-explaining models [38] provide robust explanatory capabilities, they are not designed for post-hoc explainability, which limits their usability.

III. Methodology

We define here ILLUME, a procedure to train an interpretable latent space model that enable learning global post-hoc surrogates to approximate black-box systems in a transparent way. We first outline the problem formulation and the design principles. Then we provide the details of the meta-encoding model and the required regularization terms. Finally, we describe the procedure that leverages the interpretable encoding to realize the explanation generator. In Table I we summarize the notation adopted in the rest of the paper.

A. Problem Setting and Proposed Approach

Given an input space $\mathcal{X} \subset \mathbb{R}^m$ where m is the number of features, let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote a dataset of n instances in $\mathcal{X}$. Each instance $\mathbf{x}_i = \{x_{i,1}, \ldots, x_{i,m}\}$ consists of m feature values, where $x_{i,j}$ represents the value of the j-th feature in $\mathbf{x}_i$. We define a black-box b predictor trained on X, $b : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is the codomain of the black-box. We focus here on classification tasks, where $\mathcal{Y} \subset [0,1]^c$ and c denotes the number of classes. Typically, the black-box outputs probability estimates for each class, i.e., $c = 2$ corresponds to binary classification problems, while $c > 2$ applies to multi-class problems[1]. Local explanations of black-box b are the output of functions $e^b : \mathcal{X} \to \mathcal{E}$, where $\mathcal{E}$ is the space of explanations. Given an instance $\mathbf{x}_i$, a local explainer optimizes a function e_i^b that returns $e_i^b(\mathbf{x}_i)$ as explanation, to highlight the factors that activate black-box decision $b(\mathbf{x}_i)$. Given the pair (X, Y), where $Y = \{b(\mathbf{x}_1), \ldots, b(\mathbf{x}_n)\}$ denotes the predictions of the black-box b on X, the objective of ILLUME is to fit a single *explanation generator* e_g that, given any instance $\mathbf{x}_i$, is able to explain the decision $b(\mathbf{x}_i)$. Thus, instead of returning an independent explainer e_i^b for each instance like for local surrogates, ILLUME produces a single ML function e_g that individually adapts to the instances analyzed. Such a post-hoc explanation generator offers superior advantages over inherently local explainers. As an inductive function, the generator has more generalization capabilities over local surrogates, which require training an independent model for each instance. By retraining for every single explanation, local surrogates are impractical for extensive interpretation tasks.

At the core of ILLUME is the *meta-encoder*. Departing from traditional latent space approaches, such as autoencoders or conditioned autoencoders [34], [39], which directly learn functions $\mathcal{X} \to \mathcal{Z}$ or $(\mathcal{X} \times \mathcal{Y}) \to \mathcal{Z}$, we focus on optimizing a more sophisticated class of functions $f : \mathcal{X} \to (\mathcal{X} \to \mathcal{Z})$. While mimicking classical encoders in the mapping result, proposed meta-encoder optimizes an intermediate space $\mathcal{W}$, i.e., the continuous space of linear applications from $\mathcal{X}$ to $\mathcal{Z}$, returning an *individual, locally-linear transformation* for each instance. Individual and linear maps enhance interpretability

[1]Although here we focus on binary classification, our approach can be easily extended to multi-class problems using one-vs-rest classifiers.

through linear combinations of input variables and, being instance-specific, offer more expressive power than usual linear maps. ILLUME's steps illustrated in Figure 2 are as follows:

Step (1) The meta-encoder training function $\Xi : (\mathcal{X} \times \mathcal{Y}) \rightarrow (\mathcal{X} \rightarrow \mathcal{W})$ is trained to return the meta-encoder $f : \mathcal{X} \rightarrow \mathcal{W}$ over the conditioned space $\mathcal{X} \times \mathcal{Y}$. Being conditioned to b, we denote the learned function as f^b, which returns locally-linear transformations η_i^b for any given instance $\mathbf{x}_i$. Each $\eta_i^b : \mathcal{X} \rightarrow \mathcal{Z}$ maps input records into a k-D latent space, while preserving feature and decision proximities.

Step (2) Then, for each instance $\mathbf{x}_i \in X$, the trained meta-encoder derives its local-linear transformations $\eta_i^b = f^b(\mathbf{x}_i)$ that consists in a matrix $W_i^b \in \mathbb{R}^{m \times k}$. Such transformations are applied to each instance $\mathbf{x}_i \in X$, obtaining the latent embeddings $\mathbf{z}_i = \eta_i^b(\mathbf{x}_i) = W_i^b \mathbf{x}_i$.

Step (3) Given the dataset of latent instances $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ and the observed black-box decisions Y, a second optimization $\Phi : (\mathcal{Z} \times \mathcal{Y}) \rightarrow (\mathcal{Z} \rightarrow \mathcal{Y})$ is performed to train the *surrogate model* $g : \mathcal{Z} \rightarrow \mathcal{Y}$ that globally approximate the black-box in the latent space $\mathcal{Z}$.

Step (4) At inference time, given any unseen instance $\mathbf{x}_{test}$, the locally-linear transformation and the latent representation are obtained as $W_{test}^b = f^b(\mathbf{x}_{test})$ and $\mathbf{z}_{test} = W_{test}^b \mathbf{x}_{test} = \eta_{test}^b(\mathbf{x}_{test})$. The decision of the surrogate is obtained as $y_{test} = g(\mathbf{z}_{test})$, while the explanation is extracted from the *explanation generator* $e_g(\mathbf{z}_{test}, \eta_{test}^b)$ depending on the type of the surrogate.

ILLUME constructs explanations according to the chosen surrogate model, e.g., providing feature importance with linear models, or factual/counterfactual rules with decision trees. The encoder layer is trained separately from the surrogate, which allows capturing the black-box behavior in a general-purpose meta-model. This modular design enables seamless integration with any surrogate predictor without redesigning the architecture. ILLUME is designed for input data with interpretable features, such as semantically labeled or concept-based attributes typical of tabular datasets. While this may seem restrictive, interpretable feature representations are the backbone of many explainers that target black-box predictive systems [40]. Hence, we focus on explanations for tabular data, which are easier to analyze and understand without conversions.

B. Principled Design of ILLUME

As depicted in Figure 2, **(1)** consists of training the meta-encoder $f^b : \mathcal{X} \rightarrow (\mathcal{X} \rightarrow \mathcal{Z})$, capable of returning encoding functions $\eta_i^b : \mathcal{X} \rightarrow \mathcal{Z}$ for any input $\mathbf{x}_i$. In **(2)**, individual encodings map input instances into latent representations, that are used in **(3)** as predictor variables for surrogate model fitting. We outline here the key properties that encoding functions, obtained in **(1)**, should satisfy to ensure that any interpretable surrogate used in **(3)** can produce meaningful explanations.

(P1) Decision Conditioning. The encoding captures the relationships among the input features $\mathbf{x}_i$ and the black-box outcomes for each instance $b(\mathbf{x}_i)$, thereby aligning the learned representations with the black-box decision boundary [21], [22]. In other words, the encoding η also depends on the local black-box prediction b, such that $\mathbf{z}_i = \eta^b(\mathbf{x}_i)$. This design choice ensures that the surrogate model trained on the interpretable latent encoding $\mathcal{Z}$ will capture the behavior of the black-box model it aims to explain.

(P2) Local Linearity. The encoding maps linearly the input, by using matrix transformation $W^b \in \mathbb{R}^{m \times k}$, such that relationship between the original and the latent features is human-interpretable [21], [41]. Moreover, instead of using a global transformation W^b for all instances, we allow each instance to have its own *individual* linear map. Indeed, we aim to derive a set of matrices $W = \{W_1^b, \ldots, W_n^b\}$, each of which linearly maps an instance to its corresponding latent representation, i.e., $\mathbf{z}_i = \eta_i^b(\mathbf{x}_i) = W_i^b \mathbf{x}_i$. Specifically, $z_{i,r} = \sum_{j=1}^{m} W_{i,j,r}^b x_{i,j}$ where $W_{i,j,r}$ is the value that models the linear relationship between the j-th input feature and the r-th latent feature for the i-th instance. In this way η_i^b retains the flexibility of a deep architecture, avoiding the loss of expressiveness that arises when using a single globally linear transformation [21], [22].

(P3) Explanation Consistency. Individual transformations inform how to map specific instances locally, influencing surrogate predictions and thus the explanation generator. To ensure *consistent* explanations, we must guarantee that similar instances $\mathbf{x}_1$ and $\mathbf{x}_2$ receive similar encodings, i.e., W_1 and W_2 should be close. Specifically, if $\eta_i^b(\mathbf{x}_i) = W_i^b \mathbf{x}_i$ represents the mapping for $\mathbf{x}_i$, then the same transformation should hold for a sufficiently small perturbation $\mathbf{x}_j = \mathbf{x}_i + \delta$ applied to the input [42], such that $\eta_j^b(\mathbf{x}_j) \approx \eta_i^b(\mathbf{x}_i + \delta) = W_i^b \mathbf{x}_i + W_i^b \delta$.

Building on the above principles, ILLUME is inspired by dimensionality reduction [29], learning an encoding to the space $\mathcal{Z}$. First, ILLUME incorporates both feature-wise and prediction-wise similarity into the latent space model **(P1)**. Inspired by conditioned training in autoencoders and other recent conditioned approaches in the XAI literature [21], [22], [43], through Ξ, we optimize f^b on an augmented space $\mathcal{X}^b \subset \mathcal{X} \times \mathcal{Y}$, using the enriched dataset $X^b = \{(\mathbf{x}_i, b(\mathbf{x}_i)) \mid \forall i \in [1, n]\}$ with black-box predictions paired to each instance. Second, ILLUME permits training linear transformations without constraining the encoding architecture to be strictly linear, thereby enhancing its expressive power **(P2)**. In particular, ILLUME supports the use of a shared model to compute individual linear transformations while maintaining the ability to generalize beyond the training instances. We express $W_i^b = f^b(\mathbf{x}_i)$ to emphasize that the i-th transformation is an explicit learnable function applied to $\mathbf{x}_i$. Third, transformations W_i^b are enforced to be stable w.r.t. infinitesimal displacements of $\mathbf{x}_i$ **(P3)**. Intuitively, we demand local Lipschitz continuity [11] to bound $||f^b(\mathbf{x}) - W_i^b||_{\mathrm{F}}$ with $\Lambda||\mathbf{x} - \mathbf{x}_i||$, for some constant $\Lambda \in \mathbb{R}$ and perturbation $\delta \in \mathbb{R}^m$, such that $||\mathbf{x} - \mathbf{x}_i|| < ||\delta||$. Hence, given $\mathbf{x}_i$, we require that f^b applied to a perturbed instance $(\mathbf{x}_i + \delta)$ remains similar to the transformation applied to $\mathbf{x}_i$. This can obtained by minimizing $L^{st} = \frac{1}{n} \sum_i || J_i - W_i^b ||_{\mathrm{F}}^2$, where $J_i \in \mathbb{R}^{m \times k}$ is the Jacobian matrix of the transformation for the data-point $\mathbf{x}_i$, with entries $J_{i,j,r} = \frac{\partial z_{i,r}}{\partial x_j} = W_{i,j,r}^b + \sum_{v=1}^{m} \frac{\partial W_{i,v,r}^b}{\partial x_j} x_{i,v}$.

Since the surrogate aims to replicate black-box decisions $b(\mathbf{x}_i)$, the linear functions η_i^b are designed not to directly accept $b(\mathbf{x}_i)$ as input argument. This prevents information from leaking from the features to the classification label. Instead, the influence of the black-box is incorporated indirectly through the loss function, as detailed in the following. Conceptually, the training optimization Ξ conditioned on $\mathcal{Y}$ takes as input the training instances along with black-box decisions, and returns the trained meta-encoder f^b that does not require $\mathcal{Y}$ as input. We underline that this is markedly different than using a function f^b on the explicit domain $\mathcal{X} \times \mathcal{Y}$. Indeed, in our proposal, the dependences from $\mathcal{Y}$'s observation are captured implicitly in the optimized weights of the neural function f^b and not requested at inference time.

C. Training the Interpretable Meta-Encoder

Considering the step **(1)** of Figure 2, ILLUME procedure starts by training the meta-encoder function $f^b : \mathcal{X} \to (\mathcal{X} \to \mathcal{Z})$. For any input $\mathbf{x}_i$, the learned model f^b returns a mapping $\eta_i^b : \mathcal{X} \to \mathcal{Z}$ in the form of a matrix $W_i^b \in \mathbb{R}^{m \times k}$, i.e., $W_i^b = f^b(\mathbf{x}_i)$. The scope of this matrix is to linearly embed the original instance $\mathbf{x}_i$ into $\mathbf{z}_i = \eta_i^b(\mathbf{x}_i) = W_i^b \mathbf{x}_i$. In this section, we describe the training details of ILLUME to obtain the model f^b, capable of generating linear transformations and satisfying the design principles discussed.

Learning Objective. To learn the meta-encoder f^b, we enforce that the input pairwise distance distributions, $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$, mirror the distribution $P_{\mathcal{Z}}$ over the corresponding latent representations. This requirement forces $\mathcal{Z}$ to capture the structure of data and black-box decisions in $\mathcal{X}^b$, thus yielding feature embeddings that faithfully incorporate the black-box behavior. Moreover, the introduced latent transformations, $\{W_i^b\}_{i=1\ldots n} \subset \mathcal{W}$, modulating the mapping of inputs into the encoding space $\mathcal{Z}$, are optimized to prevent their distribution from deviating arbitrarily, by requiring $P_{\mathcal{W}}$ to stay close to $P_{\mathcal{Z}}$. This ensures that also $P_{\mathcal{W}}$ as well reflects the black-box behavior and feature distribution. The learning objective for the model returning f^b, then, consists of minimizing the following superposition of Kullback-Leibler divergences:

$$L^{kl} = \frac{1}{n} \sum_i \underbrace{KL_i(P_{\mathcal{X}} || P_{\mathcal{Z}}) + KL_i(P_{\mathcal{Z}} || P_{\mathcal{W}})}_{\text{aligns } P_{\mathcal{Z}} \text{ and } P_{\mathcal{W}} \text{ with } P_{\mathcal{X}}} + \underbrace{KL_i(P_{\mathcal{Y}} || P_{\mathcal{Z}})}_{\text{aligns } P_{\mathcal{Z}} \text{ with } P_{\mathcal{Y}}}$$

where $KL_i(P_\Omega || P_{\Omega'}) = \sum_{j=1}^{n} S_{i,j}(\Omega) \log \frac{S_{i,j}(\Omega)}{S_{i,j}(\Omega')}$. Probability distributions are calculated with pairwise similarity $S_{i,j}$ between instances $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, black-box predictions $b(\mathbf{x}_i), b(\mathbf{x}_j) \in \mathcal{Y}$, latent representations $\mathbf{z}_i, \mathbf{z}_j \in \mathcal{Z}$, or individual mappings $W_i^b, W_j^b \in \mathcal{W}$, as: $S_{i,j}(\Omega) = e^{-d_\Omega(i,j)^2} / \sum_{v \neq i} e^{-d_\Omega(i,v)^2}$, where $d_\Omega(\cdot,\cdot)$ denotes a specified distance metric over the space Ω. In practice, the similarity $S_{i,j}$ represents the probability of j being a neighbor of i according to a Gaussian distribution centered on i. This objective encourages distributions $P_{\mathcal{W}}$ and $P_{\mathcal{Z}}$ to align with $P_{\mathcal{X}^b}$. Instead of directly aligning spaces $\mathcal{W}$ and $\mathcal{X}^b$, we minimize the divergence between distributions $P_{\mathcal{W}}$ and $P_{\mathcal{Z}}$. The concordance with space $\mathcal{Z}$, which is optimized to mirror $\mathcal{X}$ and $\mathcal{Y}$, indirectly aligns $\mathcal{W}$ and $\mathcal{X}^b$, with Z serving as a denoised and compact abstraction of X^b. Also, we express the loss as $L^{kl} = L_x^{kl} + L_y^{kl}$, emphasizing the term $L_y^{kl} = \frac{1}{n} \sum_i KL_i(P_{\mathcal{Y}} \| P_{\mathcal{Z}})$ responsible for conditioning on $\mathcal{Y}$.

Model Regularizations. We employ latent space regularization to impose additional constraints, ensuring *sparsity*, *orthogonality*, and *non-collinearity*. Inspired by previous work on α-sparse autoencoders [44], we *sparsify* the transformation matrices W_i^b into $sp_\alpha(W_i^b)$ by identifying the α largest weights for each column and setting the other to zero. This mechanism ensure that latent space mapping maintains a linear relationship with a limited number of input features. Moreover, it enables the user to choose the preferred sparsity level α. Also, to minimize redundancy and ensure that diverse input features contribute to distinct latent dimensions, we apply *soft-orthogonality* constraints [45] between column pairs of the transformation matrices, i.e., we impose to minimize the loss $L^{so} = \frac{1}{n} \sum_i ||sp_\alpha(W_i^b) sp_\alpha(W_i^b)^\top - \mathbb{1}_k||_{\text{F}}^2$, with $\mathbb{1}_k$ the unitary matrix. Finally, to ensure the resulting latent space is composed of minimally correlated variables [46], we optimize the correlation matrix of latent data-points $C(Z)$, with entries as the empirical Pearson scores[2] $C_{r,s} = \frac{1}{n} \sum_i \left(\frac{z_{i,r} - \mu(\mathbf{z}_{:,r})}{\sigma(\mathbf{z}_{:,r})} \right) \left(\frac{z_{i,s} - \mu(\mathbf{z}_{:,s})}{\sigma(\mathbf{z}_{:,s})} \right)$, denoting with $\mathbf{z}_{:,r} = \{z_{1,r}, \ldots, z_{n,r}\}$ the realizations of the r-th feature, and with $\mu(\cdot)$ and $\sigma(\cdot)$ the empirical average and standard deviation functions. *Non-collinearity* is reached by imposing that correlation matrix is nearly identical to the unitary matrix: $L^{co} = ||C(Z) - \mathbb{1}_k||_{\text{F}}^2$.

Optimization. The training of Ξ to obtain f^b is done using mini-batch gradient minimization, where the objective function a single batch of training instances is:

$$\mathcal{L}(X, Y, \alpha) = L_x^{kl} + \lambda^y L_y^{kl} + \lambda^{st} L^{st} + \lambda^{so} L^{so} + \lambda^{co} L^{co}$$

Similar to α-sparse autoencoders [44], we introduce a sparsity scheduling approach over training epochs to avoid "dead" hidden units within the deep neural network architecture of Ξ. Namely, we begin by pre-training the linear model without sparsity constraints, then gradually increase sparsity linearly to the desired level, and finally fine-tune the resulting sparse model until convergence.

D. Generating the Explanations

In Figure 2, after training the meta-encoder in **(1)**, individual encodings map input instances into latent representations in **(2)**, that serve as predictor variables for fitting the surrogate model g in **(3)**. At inference time **(4)**, for any instance $\mathbf{x}_i$, ILLUME produces an explanation through the generator $e_g : \mathcal{Z} \times \mathcal{W} \to \mathcal{E}$. This function receives both the meta-encoded local transformation $W_i^b = f^b(\mathbf{x}_i)$ and the resulting linear embedding $\mathbf{z}_i = W_i^b \mathbf{x}_i$, to translate the latent decision logic of g into an explanation. Here, we describe how e_g generates local explanations by exploiting the inner logic of the surrogate and the locally linear structure of the latent space.

[2]Non-linear rank-based correlation measures, such as Spearman or Kendall scores, are harder to optimize requiring differentiable sorting algorithms [47].

Feature Importance Explanations. Given an instance $\mathbf{x}_i$, feature importance explainers assign a real-valued vector $\psi_i = \{\psi_{i,1}, \ldots, \psi_{i,m}\}$, in which every $\psi_{i,j}$ is the relevance of the j-th feature for the prediction $b(\mathbf{x}_i)$. For example, the vector $\psi^{(\text{Alice})} = \{\psi_{age} = -0.2, \psi_{inc} = 0.8, \psi_{edu} = 0.5\}$ that we can obtain through ILLUME, illustrates the feature importance for the decision to reject the loan application for $\mathbf{x}^{(\text{Alice})} = \{age{=}25, income{=}15k, education{=}high\}$. Indeed, by learning a logistic classifier over Z, in ILLUME we first derive explanations expressed on latent encodings in the form of additive feature attributions $g(\mathbf{z}_i) = \beta_0 + \sum_{r=1}^{k} \beta_r z_{i,r}$. Then, by exploiting the linearity of η_i^b, through e_g, the projected space attributions[3] are converted into input space attributions $g(\eta_i^b(\mathbf{x}_i)) = \sum_{j=1}^{m} \psi_{i,j} x_{i,j}$, with feature importance defined as $\psi_{i,j} = \sum_{r=1}^{k} \beta_r W_{i,j,r}^b$. Hence, in ILLUME the local importance of input feature j is determined by summing up the global relevances of the latent features $\{\beta_r\}$ weighted by the magnitudes of the mapping $W_{i,j,r}^b$. This weighting reflects how strongly feature j contributes to each latent feature r based on the logistic surrogate coefficients. Recalling the example above, suppose that instance $\mathbf{x}^{(\text{Alice})}$ is mapped into a 2-D vector $\mathbf{z}^{(\text{Alice})} = \{z_1{=}W_{age,1}^{(\text{Alice})} x_{age} + W_{inc,1}^{(\text{Alice})} x_{inc}, z_2{=} W_{age,2}^{(\text{Alice})} x_{age} + W_{edu,2}^{(\text{Alice})} x_{edu}\}$ given by a sparse transformation valid for Alice. Hence, after learning the global logistic explanation $\psi^z = \{\beta_1, \beta_2\}$, the local feature importance vector $\psi^{(\text{Alice})}$ is generated as: $\psi^{(\text{Alice})}{=}\{\psi_{age}{=}\beta_1 W_{age,1}^{(\text{Alice})} + \beta_2 W_{age,2}^{(\text{Alice})}, \psi_{inc}{=} \beta_1 W_{inc,1}^{(\text{Alice})}, \psi_{edu}{=}\beta_2 W_{edu,2}^{(\text{Alice})}\}$.

Decision Rule Explanations. Given a record $\mathbf{x}_i$, a set of decision rules ρ_i^x explains the black-box decision $b(\mathbf{x}_i)$ with the logical premises that lead to the decision [7]. ρ_i^x is composed by axis-parallel Boolean conditions on feature values in the form $x_{i,j} \in [l_{i,j}^x, u_{i,j}^x]$, where $l_{i,j}^x, u_{i,j}^x$ are lower and upper bound values in the domain of $x_{i,j}$, extended with $\pm\infty$. For example, the rule $\rho^{(\text{Bob})} = \{age \leq 20, income \leq 30k, education \leq bachelor\}$ that we can obtain through ILLUME, explains the rejection of loan application for $\mathbf{x}^{(\text{Bob})} = \{age{=}18, income{=} 25k, education{=}low\}$. In ILLUME, when g is a decision tree, we first derive global decision rules, determined as root-leaf paths in the decision tree trained over feature space Z, i.e., $\rho_i^z = \{z_{i,r} \in [l_r^z, u_r^z]\}_{r=1\ldots k}$ (lower and upper bound are in the domain of $z_{i,r}$, extended with $\pm\infty$). Then, by exploiting the linearity of η_i^b, these rules are converted into input space local oblique rules $\tilde{\rho}_i^x = \{\sum_{j=1}^{m} W_{i,j,r}^b x_{i,j} \in [l_r^z, u_r^z]\}_{r=1\ldots k}$. Also, for more readability, we convert oblique rules into the axis-parallel format: $\rho_i^x = \{x_{i,j} \in [l_{i,j}^x, u_{i,j}^x]\}_{j=1\ldots m}$. The upper and lower bounds for these rules satisfy the following constraints: $l_{i,j}^x - x_{i,j} = \max_r \frac{l_r^z - z_{i,r}}{W_{i,j,r}^b}$ and $u_{i,j}^x - x_{i,j} = \min_r \frac{u_r^z - z_{i,r}}{W_{i,j,r}^b}$, where the max/min operations ensure taking the most restrictive inequality among the k oblique latent conditions. Essentially, with ILLUME the global rules with bounds $[l_r^z, u_r^z]$ are locally rescaled using the individual weights $W_{i,j,r}^b$. This rescaling makes the explanations more adaptive, tailoring them to the local contributions of each input feature. For instance, w.r.t. the previous example, we fit a surrogate tree g on the 2-D embedding space, and obtain global latent rules $\rho^z = \{-\infty \leq z_1 \leq \xi, \lambda \leq z_2 \leq +\infty\}$. Due to the mapping linearity, we obtain the local axis-parallel rules valid for Bob as: $\rho^{(\text{Bob})}{=}\big\{x_{age} - \frac{\xi - z_1}{W_{age,1}^{(\text{Bob})}} \leq x_{age} \leq x_{age} + \frac{\lambda - z_2}{W_{age,2}^{(\text{Bob})}}, -\infty \leq x_{inc} \leq x_{inc} + \frac{\xi - z_1}{W_{inc,1}^{(\text{Bob})}}, x_{edu} - \frac{\lambda - z_2}{W_{edu,2}^{(\text{Bob})}} \leq x_{edu} \leq +\infty\big\}$.

Perfect Fidelity via Similarity Search. Relying on surrogate logic, in ILLUME the explanations $e_g(\mathbf{z}_i, \eta_i^b)$ are valid iff $g(\eta_i^b(\mathbf{x}_i)) = b(\mathbf{x}_i)$, i.e., when the surrogate correctly predicts the black-box. To ensure producing valid explanations for *every* instance, when $g(\eta_i^b(\mathbf{x}_i)) \neq b(\mathbf{x}_i)$, we perform a vector search to find in Z the nearest latent instance $\mathbf{z}_j$ to $\mathbf{z}_i$ s.t. $g(\eta_j^b(\mathbf{x}_j)) = b(\mathbf{x}_j)$ and $b(\mathbf{x}_i) = b(\mathbf{x}_j)$, using its explanation $e_g(\mathbf{z}_j, \eta_j^b)$ instead. This approach leverages ILLUME's design, which ensures that nearby points in latent space have similar transformations. Hence, explanations for these points remain closely aligned and reliable.

IV. Experiments

We run large-scale experiments to answer these questions:

RQ1 - Is the latent space from ILLUME effective in preserving the original structure concerning both features and decisions?
RQ2 - Are black-box explanations generated with ILLUME accurate and aligned with truthful explanations?
RQ3 - Is ILLUME reliable in the absence of truthful explanations?
RQ4 - Is ILLUME computationally efficient respect to conventional local explainers?

A. Experimental Setting

In this section we detail the experimental setup adopted[4].

Datasets. We present our results on a variety of synthetic and real-world datasets used in prior works. In line with [48], [49], for synthetic datasets, we utilize the SENECA framework for implementing transparent classifiers, as proposed in [50]. Specifically, we generate synthetic datasets with t informative features and u uninformative features. The total number of features, $m = t + u$, is set in the list $\{4, 8, 16, 32, 64\}$, and we set $t = \min\{16, t + u\}$. For a fixed m, we generate five rule-based classifiers and five linear classifiers. Hence, we explain 2,048 instances for each of them. Furthermore, we employ 18 real-world datasets from UCI ML Repository. As black-box classifiers, we consider the ensemble methods [51] XGBoost (XGB), LightGBM (LGB), and CatBoost (CTB) as they are among the most effective techniques for tabular data [52]. For binary classification datasets, each explanation method is asked to generate explanations for the class 1. For multi-class datasets, explanations are referred for the majority class.

Model Setup. In line with [7], [53], we divide the input distance into continuous and categorical (one-hot encoded) features

[3]This follows from applying the encoding $z_{i,r} = \sum_{j=1}^{m} W_{i,j,r}^b x_{i,j}$ in the expression for $g(\mathbf{z}_i)$. For simplicity, we neglected the intercept β_0.

[4]Experiments were performed with CPU 3.0 GHz × 36 Intel Core i9, 252 GB RAM, and GPU Nvidia RTX 6000 24GB. Source code of ILLUME and Appendix are available at: https://github.com/simonepiaggesi/illume/.

TABLE II: Real-world data characteristics and black-box performances. In order: number of instances n, number of total features m, categorical features h, number of classes c, majority and minority class percentages, macro-F1 classification scores for XGB, LGB and CTB on test sets.

Dataset	n	m	h	c	maj(%)	min(%)	XGB	LGB	CTB
aids	2,139	36	26	2	.756	.244	.893	.895	.890
austr	690	46	40	2	.555	.445	.912	.905	.920
bank	4,119	63	53	2	.891	.109	.775	.786	.771
breast	569	30	0	2	.627	.373	.972	.991	.981
churn	3,333	71	55	2	.855	.145	.946	.930	.946
compas	7,214	20	13	2	.723	.277	.714	.711	.713
ctg	2,126	56	33	3	.778	.083	.979	.982	.979
diabetes	768	8	0	2	.651	.349	.803	.805	.824
ecoli	336	7	0	8	.426	.006	.859	.876	.892
fico	10,459	23	0	2	.522	.478	.733	.733	.737
german	1,000	61	54	2	.700	.300	.706	.706	.716
home	492	7	0	2	.545	.455	.949	.959	.949
ionos	351	34	0	2	.641	.359	.938	.953	.953
sonar	208	60	0	2	.534	.466	.881	.881	.905
spam	4,601	57	0	2	.606	.394	.959	.960	.964
titanic	891	9	5	2	.616	.384	.841	.830	.830
wine	6,497	11	0	7	.437	.001	.426	.433	.418
yeast	1,484	8	0	10	.312	.003	.592	.527	.606

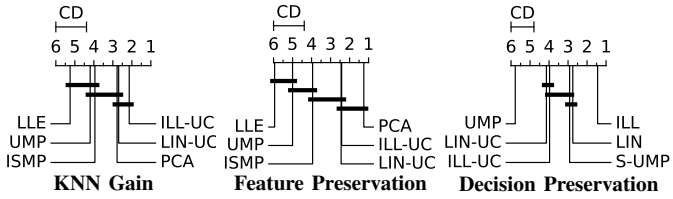
distances. With h categorical features after one-hot encoding, we use cosine and Hamming metrics to express the input distance[5] $d_{\mathcal{X}}(i,j) = \frac{h-m}{m} d_{cos}(\mathbf{x}_i^{con}, \mathbf{x}_j^{con}) + \frac{h}{m} d_{hmm}(\mathbf{x}_i^{cat}, \mathbf{x}_j^{cat})$. For the latent space and black-box distances ($d_{\mathcal{Z}}$ and $d_{\mathcal{Y}}$) we employ cosine distance as well. For latent space transformation matrices, we consider the average per-column cosine distance $d_{\mathcal{W}}(i,j) = \frac{1}{k}\sum_r d_{cos}(W^b_{i,:,r}, W^b_{j,:,r})$. We consider ILLUME-LR and ILLUME-DT, where as surrogate models g are used Logistic Regression to generate feature importance, and Decision Tree to derive rules. After hyper-parameter tuning on a validation set, we employ the best combination of regularizations, i.e., *s*oft-*o*rthogonality, and non-*co*llinearity, abbreviated with *so*, and *co*. For sparsity, we test two opposite situations: $\alpha = m$ (no sparsity), and $\alpha = 2$ (max non-trivial sparsity). In each case, every single loss term is optimized with $\lambda = 1$ or $\lambda = 0$. Also, in order to evaluate the impact of decision conditioning and consistency principles, we consider the UNConditioned (ILLUME-UC) and the UNStable (ILLUME-US) variations, setting to zero respectively λ^y and λ^{st}. Finally, we evaluate the impact of the local linearity assumption, by training a global linear encoding LIN where a single NN layer represents the function η^b, without non-linear activations.

Competitors. We compare ILLUME against feature reduction methods to evaluate the neighborhood structure preservation in the latent space: PCA, ISOMAP, LLE, (parametric-)UMAP [18]. In addition, we compare ILLUME with local explainers to evaluate the quality of explanations: LIME, SHAP, LORE, ANCHOR [54]. We use kernelSHAP method for synthetic black-boxes and treeSHAP for ensemble black-boxes. Finally, we compare ILLUME with global surrogate classifiers trained on the input space (INP-LR and INP-DT). For logistic-based surrogates

[5] $d_{cos}(\mathbf{u},\mathbf{v}) = 1 - \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||}$, $\quad d_{hmm}(\mathbf{u},\mathbf{v}) = \frac{1}{len(\mathbf{u})}\sum_i \mathbb{1}[u_i \neq v_i]$

TABLE III: RQ1 - Latent space quality metrics.

		ILL-UC	LIN-UC	PCA	ISMP	LLE	UMP
KNN Gain	best	1.015	1.002	**1.019**	.980	.943	.968
	mad	.023	.014	**.010**	.012	.017	.011
Feat. Pres.	best	.937	.933	**.986**	.819	.640	.739
	mad	**.043**	.042	**.043**	.014	.012	.006
		ILL	LIN	S-UMP	ILL-UC	LIN-UC	UMP
Dec. Pres.	best	**.700**	.670	.666	.635	.634	.597
	mad	.012	**.011**	.014	.013	.012	.005



(INP-LR and LIN-LR), as local explanation the – global – coefficients $\{\beta_1, \beta_2, \dots\}$ are weighted by the feature values of the specific instance (see model-intrinsic additive scores [27]), namely $\psi^{\text{INP}}_{i,j} = \beta_j x_{i,j}$ and $\psi^{\text{LIN}}_{i,j} = \sum_{r=1}^{k} \beta_r W^b_{j,r} x_{i,j}$. For a fair comparison, we apply the latent search for maximizing surrogate fidelity also for LIN-LR and LIN-DT using distance $d_{\mathcal{Z}}$. Instead, for LR and DT we search the nearest neighbor in the input space according to $d_{\mathcal{X}}$. For remaining methods, i.e., LIME, SHAP, LORE and ANCHOR, we trust each explanation as valid one, since all these methods ensure guarantees for maximal fidelity. Each black-box, explainer or embedding is trained and evaluated on 80/20% splits of every dataset.

B. Results and Discussion

In the following, we describe and discuss the main findings from the experiments. Within real-world data, tables display the average of best metrics across all datasets, along with average sensitivity measured by the median absolute deviation across all hyperparameters and black-boxes. Top methods are highlighted in bold for every metric, while the second-highest results are underlined. Among those, we highlight the less sensitive ones as well. Additionally, we provide Critical Difference (CD) plots to compare statistically significant average ranks (with the null hypothesis rejected at $p\text{-}value$<.001), determined using the non-parametric Friedman test, across multiple methods based on a single evaluation measure [55]. Two methods are tied if the null hypothesis that their performance is the same cannot be rejected using the Nemenyi test at 90% confidence level.

RQ1 - Latent space quality. Using real-world datasets, we compared ILLUME against dimensionality reduction frameworks to assess the quality of the latent spaces in preserving neighborhoods information. In Table III, is reported the *KNN Gain* [22], defined as the ratio of a KNN classifier's accuracy in the latent space to its accuracy in the original one, $\frac{acc_{KNN}(Z)}{acc_{KNN}(X)}$. This evaluation is based on the principle of homophily, assuming that instances within the same ground-truth class are closely clustered together, an effective latent encoding should reinforce these similarities, enabling gains when the latent configuration of instances is better organized. At this stage, for

TABLE IV: RQ2 - Correctness of synthetic explanations. Prediction accuracy of surrogate models inside parentheses.

SENECA-RC	**Feature Importance Correctness**				
$t+u$	$4+0$	$8+0$	$16+0$	$16+16$	$16+48$
ILL-LR(co)	.588 (87.6)	.476 (79.9)	**.181** (78.6)	**.133** (77.5)	.077 (73.3)
ILL-LR-UC(co)	**.603** (87.2)	**.479** (79.6)	.170 (79.1)	.124 (77.0)	**.085** (72.7)
ILL-LR-US(co)	.503 (85.1)	.393 (77.6)	**.181** (78.8)	.110 (73.1)	.081 (72.7)
LIN-LR(co)	.289 (79.1)	.226 (73.7)	.163 (77.3)	.094 (74.8)	.022 (66.0)
INP-LR	.275 (78.3)	.217 (73.2)	.107 (75.3)	.092 (74.9)	.080 (75.8)
LIME	.420	.267	.102	.076	.054
SHAP	.303	.350	.030	.031	.030
SENECA-RB	**Decision Rule Correctness**				
$t+u$	$4+0$	$8+0$	$16+0$	$16+16$	$16+48$
ILL-DT(so,α=2)	.531 (71.6)	.339 (74.4)	.240 (73.3)	.200 (71.9)	.166 (70.3)
ILL-DT-UC(so,α=2)	.523 (72.1)	.318 (73.9)	.209 (71.3)	.165 (69.3)	.128 (66.6)
ILL-DT-US(so,α=2)	.504 (70.9)	.308 (74.4)	.218 (73.1)	.199 (73.1)	.143 (68.8)
LIN-DT(so,α=2)	.545 (71.2)	.344 (74.8)	.226 (73.8)	.165 (71.3)	.116 (65.4)
INP-DT	.545 (70.5)	**.356** (72.8)	**.266** (71.2)	**.244** (69.8)	**.227** (69.6)
LORE	**.557**	.346	.202	.150	.123
ANCHOR	.402	.326	.204	.156	.139

a fair comparison with unsupervised reduction methods, we remove the effects of label conditioning by studying ILLUME-UC. Moreover, we did not observe significant improvement with regularizations. Our analysis reveals that ILLUME-UC ranks as the top-performing method, while its average performance is comparable to that of PCA, but with slightly higher variability with respect the latent dimensionality.

In Table III we also assess whether the global arrangement of the neighborhoods is preserved in terms of *Feature Preservation* and *Decision Preservation* calculated with *triplet accuracy*. The *triplet accuracy* [29] measures the percentage of triplets for which the relative ordering of pairwise distances remains consistent between the original and projected spaces. Thus, *Feature Preservation* compares the pairwise latent distance orderings $||\mathbf{z}_i - \mathbf{z}_j||_2$ with the original feature-based distances $||\mathbf{x}_i - \mathbf{x}_j||_2$, while the *Decision Preservation* compares $||\mathbf{z}_i - \mathbf{z}_j||_2$ with relative distances of black-box decisions $||b(\mathbf{x})_i - b(\mathbf{x})_j||_2$. In line with the *KNN Gain*, when comparing feature preservation across unsupervised methods, ILLUME-UC ranks second only to PCA and performs similarly to LIN-UC, having all these methods similar sensitivity to dimension size. On the other hand, when comparing label-aware methods in terms of decision preservation, label conditioning in ILLUME is determinant for significantly enhancing triplet accuracy, even outperforming LIN and S-UMAP, i.e., UMAP trained in supervised setting. These results highlight ILLUME's ability to capture black-box decision logic and feature proximity in latent representations.

RQ2 - Correctness of explanations. We employed SENECA-RB and SENECA-RC [50] to study the *explanation correctness* of local model-agnostic explainers on tabular data[6]. By exploiting synthetic transparent classifiers and treating them as black-boxes, we can directly compare the explanations provided by an explainer with the ground-truth decision logic of the synthetic black-box[7]. We evaluate the correctness of local explanations by measuring the closeness between the extracted explanations ϵ and the ground-truth $\hat{\epsilon}$ provided by the synthetic classifiers of SENECA. Following [48], [50], for feature importance we measure the proximity of two explanations with the *cosine similarity score* of attribution vectors: $cs\text{-}score(\psi, \hat{\psi}) = \frac{\psi \cdot \hat{\psi}}{||\psi||\ ||\hat{\psi}||}$. Besides, for decision rules we measure the similarity of the bounded regions of the instance space described by two rules, calculating the closeness between upper and lower bounds when both are different from $\pm\infty$ (*complete rule score*[8]): $cplt\text{-}score(\rho, \hat{\rho}) = \frac{1}{N_{\not\infty}}(\sum_j \frac{1}{1+|l_j - \hat{l}_j|^2} + \sum_j \frac{1}{1+|u_j - \hat{u}_j|^2})$.

TABLE V: RQ3 - Quality for feature importance.

Robustness	ILL-LR (co)	ILL-LR-US (co)	LIN-LR (co)	INP-LR	LIME	SHAP
best	.959	.863	.494	.414	**.978**	.373
mad	**.062**	.112	.042	.016	.103	.037
Faithfulness	ILL-LR (.)	ILL-LR-UC (.)	LIN-LR (so,α=2)	INP-LR	LIME	SHAP
best	.656	.397	.496	.485	.230	**.663**
mad	.097	.050	.039	.026	.049	**.026**

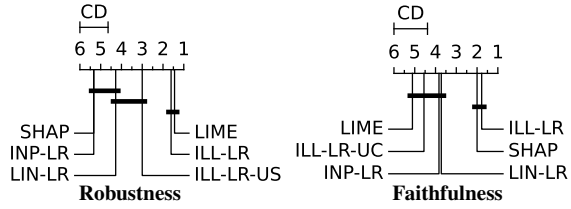


Table IV reports correctness synthetic explanations from SENECA-RC and SENECA-RB, using the best regularizer for ILLUME and LIN. Correctness generally decreases with noisy input dimensions denoted with u (while t refers to informative dimensions). ILL-LR consistently outperforms LIN-LR and INP-LR in surrogate performance. For feature importance explanations, ILL-LR and its variants (ILL-LR-UC, ILL-LR-US) significantly surpass competitors, with label conditioning and consistency having minimal impact. For decision rule explanations, ILL-DT ranks second-best for input dimensions ≥ 16, always outperforming ILL-DT-UC/ILL-DT-US and surpassing LIN-DT for dimensions ≥ 8. INP-DT performs best, as expected, since SENECA-RB is based on decision trees trained in the input space, aligning its structure with the ground-truth logic.

RQ3 - Robustness and faithfulness of explanations. Resorting real-world datasets, we compared ILLUME against local model-agnostic explainers to assess the quality of the resulting explanations. Since ground-truth for individual explanations is unavailable in real-world data [50], we focus on other criteria that are critical for explanations' evaluation, i.e., *(i)* their sensitivity to feature modifications (*Robustness*) [11], and *(ii)* their ability to accurately reflect the reasoning of the black-

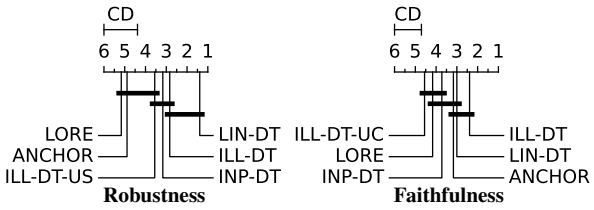
[6]In the repository accompanying the paper, we extend the evaluation with additional synthetic benchmarks [56], obtaining comparable results.

[7]Please refer to [50] for the definitions of ground-truth explanations.

[8]$N_{\not\infty}$ denotes the number of finite lower/upper bounds in a decision rule. When an infinite bound is subtracted from a finite one, the fraction counts zero ($\frac{1}{1\pm\infty} = 0$). Our metric spans between 0 and 1 and captures differences in a continuous spectrum, while in [50] a binarized metric is used.

TABLE VI: RQ3 - Quality for decision rules.

Robustness	ILL-DT (so,α=2)	ILL-DT-US (so,α=2)	LIN-DT (so,α=2)	INP-DT	LORE	ANCHOR
best	.505	.440	**.672**	.439	.225	.221
mad	**.058**	.053	.084	.018	.021	.018
Faithfulness	ILL-DT (so,α=2)	ILL-DT-UC (so,α=2)	LIN-DT (so,α=2)	INP-DT	LORE	ANCHOR
best	**.658**	.578	.626	.605	.500	.600
mad	**.058**	.077	.069	.021	.034	.020



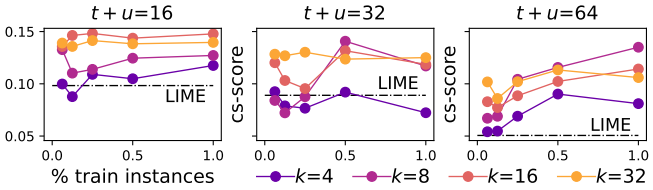
box model (*Faithfulness* or fidelity) [57]. In line with [58], we evaluate robustness for each instance as the maximum change in the explanation with small input perturbations. For each test instance $\mathbf{x}_i$, we compute the average *max-sensitivity* over multiple nearest neighborhoods ($K_{max} = 20$) $\frac{1}{K_{max}} \sum_{K=1}^{K_{max}} \max_{j \in \mathcal{N}_K^{=}(i)} d_{\mathcal{E}}(i,j)$, where $\mathcal{N}_K^{=}(i)$ denotes the set of K nearest neighbors of $\mathbf{x}_i$ with same predicted black-box label, and $d_{\mathcal{E}}(i,j)$ is a suitable pairwise explanations distance. In line with [57], we evaluate fidelity by assessing whether similar black-box predictions result in similar explanations. Thus, faithfulness is evaluated by measuring the rank correlation between the pairwise distances of explanations, $\{d_{\mathcal{E}}(i,j)\}_{i<j}$, and the corresponding pairwise differences in black-box predictions, $\{||b(\mathbf{x}_i) - b(\mathbf{x}_j)||_2\}_{i<j}$. As for SENECA, we use proximity metrics *cs-score* and *cplt-score* to evaluate explanation (dis)similarity. For robustness, we select the most dissimilar explanations within each fixed-size neighborhood.

Tables V and VI present the results, with CD plots summarizing overall performance across datasets and black-boxes. As generally expected, latent methods without stability optimization show weaker robustness, and removing decision conditioning compromises faithfulness. Regarding importance-based explainers in Table V, LIME demonstrates the highest robustness, while SHAP is the most faithful, even though they do not excel in both metric. Conversely, the CD plots highlight that ILL-LR is statistically comparable to the best of them in both metrics. Moreover, in reference to robustness, it appears also less sensitive to hyperparameters than LIME. Regarding rule-based explainers in Table VI, latent methods ILL-DT and LIN-DT exhibit strong robustness, while LIN-DT being more sensitive to hyperparameters. For faithfulness, ILL-DT is the most effective, being less sensitive than its runner-up LIN-DT.

RQ4 - Efficiency of explanations. We measured the inference time required to generate feature importance on synthetic datasets with ground-truth explanations (see **RQ2**). In ILLUME, the explanation generator is shared across all instances. Thus, once the model has been trained, it can produce explanations for unseen instances with a single forward pass through the meta-encoder and the surrogate model. Because conventional local surrogates have to retrain the model for every instance, the inductive capabilities of our explainer make it substantially more efficient [59]. In fact, training costs in ILLUME incur only once, and involve learning the parameterized function f^b via mini-batches, scaling efficiently to large datasets without significant memory constraints. Importantly, the meta-encoder is memory-saving because dynamically computes instance-specific matrices $\{W_i^b\}$ during inference, without the need of storing them. To quantify computational benefits, we denote with explanation time as the time required to explain a new instance with an explainer that has already been trained. Because local surrogates train an independent model for every instance, their reported times include also the training cost. For ILLUME, the training cost is listed separately in parentheses.

TABLE VII: RQ4 - Efficiency of explanations.

SENECA-RC	**Per-instance Explanation Time (s)**				
$t+u$	$4+0$	$8+0$	$16+0$	$16+16$	$16+48$
ILL-LR	$4.4{\cdot}10^{-3}$ (13.0±.2)	$5.3{\cdot}10^{-3}$ (12.2±.1)	$4.5{\cdot}10^{-3}$ (8.5±.08)	$6.6{\cdot}10^{-3}$ (15.1±.2)	$4.7{\cdot}10^{-3}$ (8.2±.1)
LIME	3.0±.4	8.3±1.3	26.9±7.5	27.1±6.8	28.2±6.6
SHAP	.1±.1	.5±.1	20.9±7.6	22.2±8.0	25.7±9.5



In Table VII, we show average per-instance explanation times for producing the most accurate explanations (whose correctness scores are shown in Table IV), revealing that ILLUME delivers explanations orders of magnitude faster than LIME and SHAP. Its training time is comparable or lower than the per-instance training costs of LIME and SHAP, an expense that those methods incur for every instance to explain. The figure below illustrates how ILLUME's performance scales with the number of training instances: the test *cs-score* improves as additional instances are used, confirming the effectiveness of our approach. Notably, the method achieves competitive explanations even when trained on small fractions of training data, making it especially practical for large-scale datasets.

V. Conclusions

We have introduced ILLUME, a generative framework for local explanations compatible with any interpretable surrogate model. Unlike traditional surrogate-based explainers, which often fail to satisfy desirable explanation properties, ILLUME represents a paradigm shift by combining global reasoning capabilities of interpretable surrogates with local linear encodings of input features. Through extensive experiments on tabular datasets, we have empirically demonstrated that ILLUME produces explanations that are accurate, robust, and faithful, achieving performance comparable to or surpassing state-of-the-art attribution-based and rule-based explainers in most cases.

As future work, ILLUME can be extended to end-to-end training a meta-surrogate model, where, like self-interpretable models [37], explainability is built-in architecturally and enforced through task-specific constraints. Moreover, by leveraging recent advances in tabular foundation models [60], the presented methodology can be further generalized to develop cross-dataset explanation inference frameworks.

REFERENCES

[1] F. Bodria *et al.*, "Benchmarking and survey of explanation methods for black box models," *Data Min. Knowl. Discov.*, vol. 37, no. 5, pp. 1719–1778, 2023.

[2] R. Guidotti, "Counterfactual explanations and how to find them: literature review and benchmarking," *Data Min. Knowl. Discov.*, vol. 38, no. 5, pp. 2770–2824, 2024.

[3] J. Herbinger *et al.*, "Leveraging model-based trees as interpretable surrogate models for model distillation," in *ECAI Workshops*, vol. 1947, 2023, pp. 232–249.

[4] M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *NIPS*, 1995, pp. 24–30.

[5] N. Frosst and G. E. Hinton, "Distilling a neural network into a soft decision tree," ser. CEUR Workshop Proceedings, vol. 2071, 2017.

[6] M. T. Ribeiro *et al.*, ""why should I trust you?": Explaining the predictions of any classifier," in *KDD*, 2016.

[7] R. Guidotti *et al.*, "Stable and actionable explanations of black-box models through factual and counterfactual rules," *Data Min. Knowl. Discov.*, vol. 38, no. 5, pp. 2825–2862, 2024.

[8] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *J. Artif. Intell. Res.*, vol. 70, pp. 245–317, 2021.

[9] P. Kindermans *et al.*, "The (un)reliability of saliency methods," in *Explainable AI*, 2019, vol. 11700, pp. 267–280.

[10] Y. Zhou *et al.*, "Do feature attribution methods correctly attribute features?" in *AAAI*, 2022, pp. 9623–9633.

[11] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *CoRR*, vol. abs/1806.08049, 2018.

[12] N. Bansal *et al.*, "SAM: the sensitivity of attribution methods to hyperparameters," in *CVPR Workshops*, 2020, pp. 11–21.

[13] S. M. Lundberg *et al.*, "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.

[14] E. G. Amparore *et al.*, "To trust or not to trust an explanation: using LEAF to evaluate local linear XAI methods," *PeerJ Comput. Sci.*, vol. 7, p. e479, 2021.

[15] A. Dhurandhar *et al.*, "Is this the right neighborhood? accurate and query efficient model agnostic explanations," in *NeurIPS*, 2022.

[16] T. Laugel *et al.*, "Defining locality for surrogates in post-hoc interpretablity," *CoRR*, vol. abs/1806.07498, 2018.

[17] Y. Bengio *et al.*, "Representation learning: A review and new perspectives," *IEEE TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.

[18] T. Sainburg *et al.*, "Parametric UMAP embeddings for representation and semisupervised learning," *Neural Comput.*, vol. 33, no. 11, pp. 2881–2907, 2021.

[19] P. W. Koh *et al.*, "Concept bottleneck models," in *ICML*, 2020.

[20] M. E. Zarlenga *et al.*, "Concept embedding models: Beyond the accuracy-explainability trade-off," in *NeurIPS*, 2022.

[21] S. Piaggesi *et al.*, "Counterfactual and prototypical explanations for tabular data via interpretable latent space," *IEEE Access*, vol. 12, pp. 168 983–169 000, 2024.

[22] F. Bodria *et al.*, "Transparent latent space counterfactual explanations for tabular data," in *DSAA*, 2022, pp. 1–10.

[23] Y. Takai *et al.*, "On the number of linear functions composing deep neural network: Towards a refined definition of neural networks complexity," in *AISTATS*, vol. 130, 2021, pp. 3799–3807.

[24] R. Agarwal *et al.*, "Neural additive models: Interpretable machine learning with neural nets," in *NeurIPS*, 2021, pp. 4699–4711.

[25] Y. Lou *et al.*, "Accurate intelligible models with pairwise interactions," in *KDD*, 2013, pp. 623–631.

[26] D. Ha *et al.*, "Hypernetworks," in *ICLR*, 2017.

[27] A. H. A. Rahnama *et al.*, "Can local explanation techniques explain linear additive models?" *Data Min. Knowl. Discov.*, vol. 38, no. 1, pp. 237–280, 2024.

[28] S. M. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *NIPS*, 2017, pp. 4765–4774.

[29] Y. Wang *et al.*, "Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization," *JMLR*, vol. 22, pp. 201:1–201:73, 2021.

[30] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: survey, insights, and generalizations," *JMLR*, vol. 16, pp. 2859–2900, 2015.

[31] A. J. Izenman, "Introduction to manifold learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 4, no. 5, pp. 439–446, 2012.

[32] L. K. Senel *et al.*, "Semantic structure and interpretability of word embeddings," *IEEE TASLPRO*, vol. 26, no. 10, pp. 1769–1779, 2018.

[33] H. GM *et al.*, "A comprehensive survey and analysis of generative models in machine learning," *Comput. Sci. Rev.*, vol. 38, p. 100285, 2020.

[34] L. Le *et al.*, "Supervised autoencoders: Improving generalization performance with unsupervised regularizers," in *NeurIPS*, 2018, pp. 107–117.

[35] J. Crabbé *et al.*, "Explaining latent representations with a corpus of examples," in *NeurIPS*, 2021, pp. 12 154–12 166.

[36] R. Crupi *et al.*, "Counterfactual explanations as interventions in latent space," *Data Min. Knowl. Discov.*, vol. 38, no. 5, pp. 2733–2769, 2024.

[37] Y. Ji *et al.*, "A comprehensive survey on self-interpretable neural networks," *CoRR*, vol. abs/2501.15638, 2025.

[38] A. Kadra *et al.*, "Interpretable mesomorphic networks for tabular data," in *NeurIPS*, 2024.

[39] W. Harvey *et al.*, "Conditional image generation by conditioning variational auto-encoders," in *ICLR*, 2022.

[40] K. Sokol and P. A. Flach, "Interpretable representations in explainable AI: from theory to practice," *Data Min. Knowl. Discov.*, vol. 38, no. 5, pp. 3102–3140, 2024.

[41] S. Yang *et al.*, "Local interpretation of transformer based on linear decomposition," in *ACL*, 2023, pp. 10 270–10 287.

[42] D. Alvarez-Melis and T. S. Jaakkola, "Towards robust interpretability with self-explaining neural networks," pp. 7786–7795, 2018.

[43] V. Guyomard *et al.*, "Vcnet: A self-explaining model for realistic counterfactual generation," in *ECML/PKDD*, 2022.

[44] A. Makhzani and B. J. Frey, "k-sparse autoencoders," in *ICLR*, 2014.

[45] E. M. Massart, "Orthogonal regularizers in deep learning: how to handle rectangular matrices?" in *ICPR*, 2022, pp. 1294–1299.

[46] K. Aas *et al.*, "Explaining individual predictions when features are dependent: More accurate approximations to shapley values," *Artif. Intell.*, vol. 298, p. 103502, 2021.

[47] M. Blondel *et al.*, "Fast differentiable sorting and ranking," in *ICML*, 2020.

[48] A. H. A. Rahnama, "The blame problem in evaluating local explanations and how to tackle it," in *ECAI Workshops*, vol. 1947, 2023, pp. 66–86.

[49] I. Mollas *et al.*, "Truthful meta-explanations for local interpretability of machine learning models," *Appl. Intell.*, vol. 53, no. 22, pp. 26 927–26 948, 2023.

[50] R. Guidotti, "Evaluating local explanation methods on ground truth," *Artif. Intell.*, vol. 291, p. 103428, 2021.

[51] D. M. Ibomoiye and Y. Sun, "A survey of ensemble learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99 129–99 149, 2022.

[52] L. Grinsztajn *et al.*, "Why do tree-based models still outperform deep learning on typical tabular data?" in *NeurIPS*, 2022.

[53] A. H. Foss *et al.*, "Distance metrics and clustering methods for mixed-type data," *International Statistical Review*, vol. 87, no. 1, pp. 80–109, 2019.

[54] M. T. Ribeiro *et al.*, "Anchors: High-precision model-agnostic explanations," in *AAAI*, 2018, pp. 1527–1535.

[55] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, pp. 1–30, 2006.

[56] P. Cortez and M. J. Embrechts, "Using sensitivity analysis and visualization techniques to open black box data mining models," *Inf. Sci.*, vol. 225, pp. 1–17, 2013.

[57] S. Dasgupta *et al.*, "Framework for evaluating faithfulness of local explanations," in *ICML*, 2022.

[58] C. Yeh *et al.*, "On the (in)fidelity and sensitivity of explanations," 2019.

[59] D. Luo *et al.*, "Parameterized explainer for graph neural network," in *NeurIPS*, 2020.

[60] B. van Breugel and M. van der Schaar, "Position: Why tabular foundation models should be a research priority," in *ICML*, 2024.

[61] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *KDD*, 2016, pp. 785–794.

[62] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *NIPS*, 2017, pp. 3146–3154.
[63] L. O. Prokhorenkova *et al.*, "Catboost: unbiased boosting with categorical features," in *NeurIPS*, 2018, pp. 6639–6649.
[64] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, 2022.
[65] B. McCane and M. H. Albert, "Distance functions for categorical and mixed variables," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 986–993, 2008.
[66] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

APPENDIX

A. Detailed Experimental Settings

Datasets. We implement synthetic classifiers from SENECA framework [50] by varying the dimensions of the synthetic instances. In particular, we adjust the number of informative features (t), and uninformative features (u). The total number of features, $m = t + u$, is set in the list $\{4, 8, 16, 32, 64\}$ and, for a fixed $t+u$, we define $t = \min\{16, t+u\}$. For each dataset dimension, we generate five rule-based classifiers and five linear classifiers. Hence, we explain 2,048 instances for each of them. Besides synthetic data, we employ 18 real-world datasets from UCI ML Repo[9]. In every dataset, we apply standard scaling, with zero mean and unitary variance to continuous features, and one-hot encoding to categorical features.

Black-boxes. As black-box classifiers, we consider the ensemble methods XGBoost (XGB) [61], LightGBM (LGB) [62] and CatBoost (CTB) [63], because they are among the most effective techniques for tabular data, even outperforming deep learning models [52], [64]. Anyway, there is no specified limitation in using other black-boxes, such as NNs or SVMs. Each black-box is trained using 80% of the dataset, with the remaining 20% reserved for testing and evaluation. The black-box models are trained on each dataset's classification tasks, whether binary or multi-class. For binary datasets, each explanation method is asked to generate explanations for the class 1. For multi-class datasets, each explanation method is asked to generate explanations for the majority class.

Experimental setup for ILLUME. The function f^b is implemented as a 3-layer fully-connected neural network. We did not tune the architecture with respect to the best number of layers and/or hidden sizes. Meta-encoder training is done with Adam optimizer, fixing learning rate to 10^{-3}, with early stopping technique to prevent overfitting. To compute the loss functions, for the latent space vectors we use the cosine distance[10], $d_{\mathcal{Z}}(i,j) = d_{cos}(\mathbf{z}_i, \mathbf{z}_j)$. In line with [7], [53], [65], we divide the input distance into contributions from continuous and categorical (one-hot encoded) features. Assuming h categorical features after one-hot encoding, we express the input distance[11] as $d_{\mathcal{X}}(i,j) = \frac{h-m}{m} d_{cos}(\mathbf{x}_i^{con}, \mathbf{x}_j^{con}) + \frac{h}{m} d_{hmm}(\mathbf{x}_i^{cat}, \mathbf{x}_j^{cat})$. For the black-box score vectors, we employ cosine distance as well. For latent space transformation matrices, we consider the average per-column cosine distance $d_{\mathcal{W}}(i,j) = \frac{1}{k}\sum_r d_{cos}(W^b_{i,:,r}, W^b_{j,:,r})$. All the encodings are tested tuning the latent space dimension as hyperparameter from the list $\{2, 4, 8, 16, 32\}$.

Experimental Setup for Dimensionality Reduction. We compare ILLUME against latent embedding methods to evaluate the neighborhood structure preservation in the latent space. Considered methods for dimensionality reduction are PCA[12], ISOMAP[13], LLE[14] and p-UMAP[15]. As for ILLUME, the reduction methods are tested tuning the latent space dimension as hyperparameter from the list $\{2, 4, 8, 16, 32\}$. For p-UMAP, we train a 3-layer fully-connected neural network for 50 epochs. For the other methods, we used standard parameters.

Experimental Setup for Local Explainers. We compare ILLUME with well-known local explainers to evaluate the quality of explanations: LIME[16], SHAP[17] for feature importance; LORE[18], ANCHOR[19] for decision rules. LIME is tuned with neighborhood sizes $\{100, 300, 1000, 5000\}$; LORE with sizes $\{300, 1000\}$ and $\{1, 5, 10\}$ decision trees; ANCHOR with batch sizes $\{100, 300\}$ and beam sizes $\{4, 10\}$. With SHAP we use the kernelSHAP method for synthetic black-boxes and treeSHAP for ensemble-tree black-boxes.

Experimental Setup for Global Surrogates. Global surrogates models LR[20] and DT[21] are trained with latent or input space variables by tuning their main hyperparameters to maximize test set prediction accuracy. For surrogates based on LR (INP-LR and LIN-LR), as local explanation the –global– logistic coefficients $\{\beta_1, \beta_2, \dots\}$ are weighted by the feature values of the specific instance (see model-intrinsic additive scores [27]), namely $\psi^{\text{INP}}_{i,j} = \beta_j x_{i,j}$ and $\psi^{\text{LIN}}_{i,j} = \sum_{r=1}^{k} \beta_r W^b_{j,r} x_{i,j}$. For a fair comparison, we apply the latent search for maximizing surrogate fidelity also for LIN-LR and LIN-DT using distance $d_{\mathcal{Z}}$. Instead, for LR and DT we search the nearest neighbor in the input space according to $d_{\mathcal{X}}$. For remaining methods –LIME, SHAP, LORE and ANCHOR– we trust each explanation as valid one, since all these methods ensure guarantees for maximal fidelity.

B. Qualitative Results

In Figure A1 it is illustrated how our approach works in the explanation inference phase with an example on `compas` dataset. At first glance, with ILLUME, instances and their local black-box decisions (A-B) are employed to generate local

[9] https://archive.ics.uci.edu/ml/index.php
[10] $d_{cos}(\mathbf{u}, \mathbf{v}) = 1 - \frac{\mathbf{u}\cdot\mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||}$
[11] $d_{hmm}(\mathbf{u}, \mathbf{v}) = \frac{1}{len(\mathbf{u})}\sum_i \mathbb{1}[u_i \neq v_i]$

[12] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
[13] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.Isomap.html
[14] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.LocallyLinearEmbedding.html
[15] https://umap-learn.readthedocs.io/en/latest/parametric_umap.html
[16] https://lime-ml.readthedocs.io/en/latest/lime.html\#module-lime.lime_tabular
[17] https://shap.readthedocs.io/en/latest/generated/shap.Explainer.html
[18] https://kdd-lab.github.io/LORE_sa/html/index.html
[19] https://github.com/marcotcr/anchor
[20] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[21] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

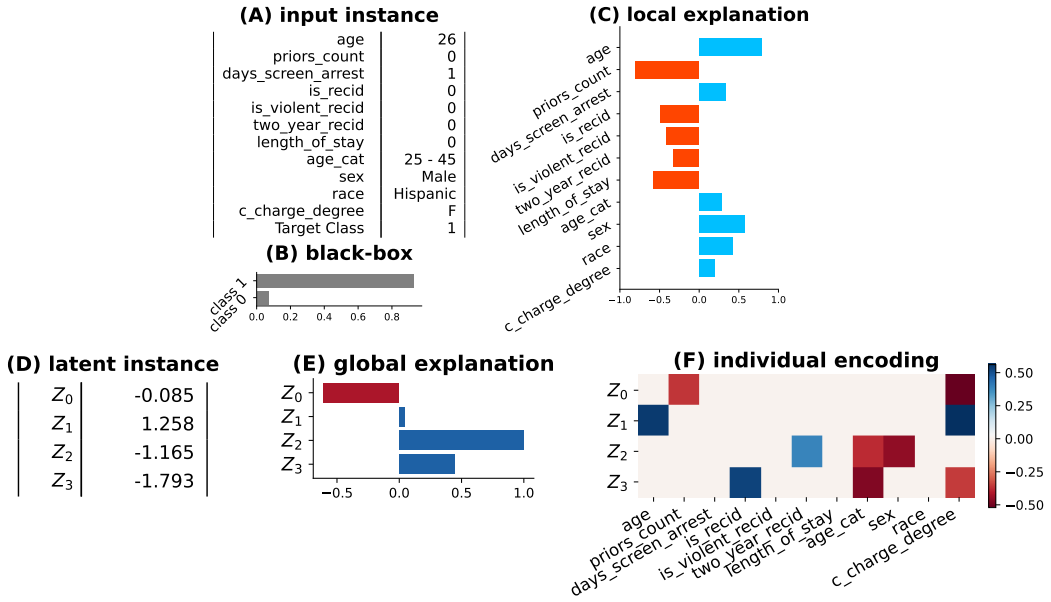


Fig. A1: Exemplification of ILLUME's inference phase on `compas` dataset when generating feature importance local explanations. Given an input data-point $\mathbf{x}_{test}$ (A) and its corresponding black-box prediction $b(\mathbf{x}_{test})$ (B), the method outputs instance-specific explanation $e_g(\mathbf{z}_{test}, \eta^b_{test})$ (C). This explanation is derived: *(i)* by encoding the instance into a latent representation $\mathbf{z}_{test}$ (D), *(ii)* extracting the logic of the global surrogate g (E), and *(iii)* combining it with local interpretable mapping η^b_{test} (F), represented by a sparse and linear transformation returned by the meta-encoder.

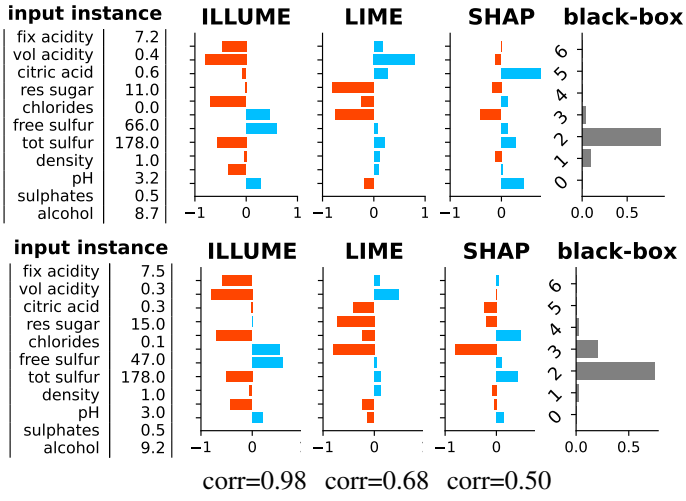


Fig. A2: Feature importance explanations for two similar records of the `wine` dataset. Feature values are reported on the left. In the center, explanations are derived with ILLUME, LIME and SHAP. On the right, the prediction probability returned by an XGB classifier.

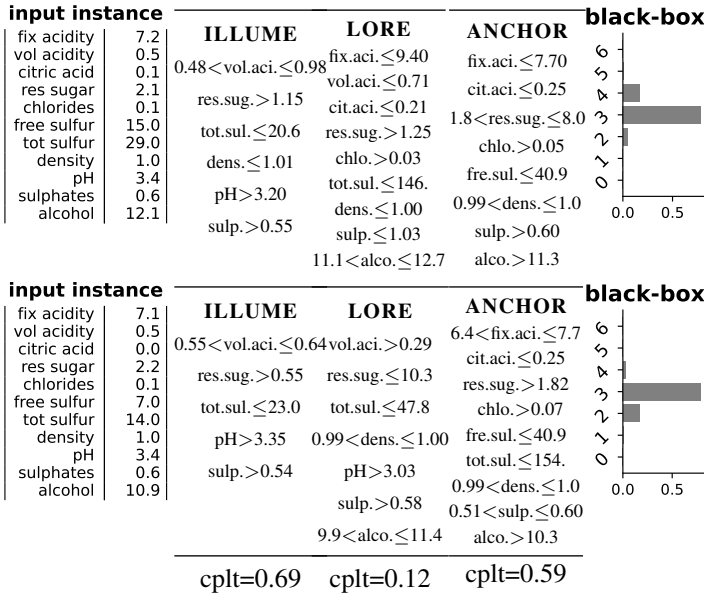


Fig. A3: Decision rule explanations for two similar records of the `wine` dataset. Feature values are reported on the left. In the center, explanations derived with ILLUME, LORE and ANCHOR. On the right, the prediction probability returned by an LGB classifier.

explanations (C). These explanations are computed with the information deriving from an auxiliary embedding space where we project the input (D). In this space we have extracted global latent explanations (E) during the surrogate learning process. The key component of this space is a set of instance-specific linear transformations (F) learned during the meta-encoder training process. Local explanations are produced by combining the latent explanation with the local projection of

the data. Figure A2 showcase feature importance explanations for two neighboring instances from the `wine` dataset. Being closely located in the feature space and yielding similar predictions from the black-box model for the multi-class task, the explanations for these instances are expected to align, ranking important features in comparable ways. ILLUME demonstrates its ability to generate robust and consistent explanations, producing rank-preserving feature importance profiles for neighboring instances. In contrast, competing methods such as LIME and SHAP fail to achieve similar consistency. Additionally, Figure A3 presents a comparable example for decision rules using the same dataset, illustrating that ILLUME is capable of generating robust and consinstent rules. The similarity of these rules is evaluated using the *cplt-score*, like in the main experimental result of the paper. Furthermore, we demonstrate that ILLUME can produce more concise decision rules with respect to the analyzed competitors, a benefit achieved through sparsity regularization.

C. Detailed Formulas for Decision Rules

In ILLUME, we first derive global explanations in the form of axis-parallel decision rules on latent features, determined by root-leaf paths in a trained decision tree, i.e., $\rho_i^z = \{z_{i,r} \in [l_r^z, u_r^z]\}_{r=1\ldots k}$ (lower and upper bound are in the domain of $z_{i,r}$, extended with $\pm\infty$). By exploiting the linearity of the mapping η_i^b, these rules are converted into input space local oblique rules $\tilde{\rho}_i^x = \{\sum_{j=1}^m W_{i,j,r}^b x_{i,j} \in [l_r^z, u_r^z]\}_{r=1\ldots k}$. While oblique rules offer a valid and expressive form of explanation, for consistency with most methods we perform an additional step to convert the oblique rules into the standard axis-parallel format: $\rho_i^x = \{x_{i,j} \in [l_{i,j}^x, u_{i,j}^x]\}_{j=1\ldots m}$. The upper and lower bounds for these axis-parallel rules are derived by isolating $x_{i,j}$ to express the rule in terms of individual input features. For example, from the oblique rules $\sum_{j=1}^m W_{i,j,r}^b x_{i,j} \geq l_r^z$ and $\sum_{j=1}^m W_{i,j,r}^b x_{i,j} \leq u_r^z$, it follows:

$$l_{i,j}^x = \max_r \frac{l_r^z - \sum_{v\neq j} W_{i,v,r}^b x_{i,v}}{W_{i,j,r}^b}$$

$$u_{i,j}^x = \min_r \frac{u_r^z - \sum_{v\neq j} W_{i,v,r}^b x_{i,v}}{W_{i,j,r}^b}.$$

The max/min operations ensure taking the most restrictive inequality among the k latent conditions, i.e., the largest lower bound and the smallest upper bound. Finally, exploiting the equality $z_{i,r} = \sum_{v\neq j} W_{i,v,r}^b x_{i,v} + W_{i,j,r}^b x_{i,j}$, we find the relations reported in the main paper that link upper/lower bounds in the embedding and in the original space:

$$l_{i,j}^x - x_{i,j} = \max_r \frac{l_r^z - z_{i,r}}{W_{i,j,r}^b} \qquad u_{i,j}^x - x_{i,j} = \min_r \frac{u_r^z - z_{i,r}}{W_{i,j,r}^b}.$$

D. Perfect Fidelity via Similarity Search

Relying on surrogate logic, in ILLUME the explanations $e_g(\mathbf{z}_i, \eta_i^b)$ are valid iff $g(\mathbf{z}_i) = b(\mathbf{x}_i)$, i.e., when surrogate model agrees with the black-box on the corresponding instance. To ensure producing valid explanations for *every* instance, we refine the latent representation of those samples for which $g(\mathbf{z}_i) \neq b(\mathbf{x}_i)$. The goal of this refinement is to realign misclassified latent points with nearby correctly predicted ones that share the same black-box label. This approach leverages ILLUME's design, which ensures that nearby points in latent space have similar transformations. Hence, explanations for these points remain closely aligned and reliable. Thus, for each $\mathbf{z}_i \in Z$ such that $g(\mathbf{z}_i) \neq b(\mathbf{x}_i)$, we proceed as follows:

(1) Closest valid neighbor search. We perform a cosine-based nearest-neighbor search in latent space to identify $\mathbf{z}_{nn} \in Z$ such that $g(\mathbf{z}_{nn}) = b(\mathbf{x}_{nn})$ and $b(\mathbf{x}_i) = b(\mathbf{x}_{nn})$. This provides a nearby latent point that is locally consistent with the black-box prediction and serves as a reference.

(2) First-order approximation. Since $\mathbf{z}_{nn}$ is the closest latent vector to $\mathbf{z}_i$, we assume their corresponding inputs differ by a small displacement (i.e., $\mathbf{x}_{nn} \approx \mathbf{x}_i + \delta$). Similarly, we assume the associated local transformations satisfy $W_{nn}^b \approx W_i^b + \Delta W_i$. We use a first-order approximation from the formula $W_{nn}^b \mathbf{x}_{nn} = (W_i^b + \Delta W_i)(\mathbf{x}_i + \delta)$ to obtain:

$$\mathbf{z}_{nn} \approx \mathbf{z}_i + J_i\delta = W_i^b \mathbf{x}_i + \Delta W_i \mathbf{x}_i + W_{nn}^b \delta + \mathcal{O}(||\Delta W_i \delta||).$$

This decomposition separates the effect of input displacement δ with the projection variation ΔW_i. Here, $W_{nn}^b \delta \approx W_i^b \delta$ because we neglected second-order terms.

(3) Interpolated latent representation. Guided by the previous approximation, we define the latent perturbation:

$$\mathbf{z}_i^\gamma = W_i^b \mathbf{x}_i + \gamma_W (W_{nn}^b - W_i^b)\mathbf{x}_i + \gamma_x W_{nn}^b (\mathbf{x}_{nn} - \mathbf{x}_i),$$

where $(W_{nn}^b - W_i^b)$ and $(\mathbf{x}_{nn} - \mathbf{x}_i)$, both scaled by small factors γ_W and γ_x, identify the directions of the perturbations δ and ΔW_i. This expression can be interpreted as an interpolation toward the nearest valid neighbor:

$$\mathbf{z}_i^\gamma = (1 - \gamma_W)\mathbf{z}_i + \gamma_W W_{nn}^b \mathbf{x}_i + \gamma_x W_{nn}^b (\mathbf{x}_{nn} - \mathbf{x}_i).$$

(4) Minimal constrained perturbation. We determine the optimal interpolation parameters by solving

$$\gamma_W^*, \gamma_x^* = \underset{(\gamma_W, \gamma_x)\in(0,1]^2}{\arg\min} ||\mathbf{z}_i^\gamma - W_i^b \mathbf{x}_i||^2 \text{ s.t. } g(\mathbf{z}_i^\gamma) = b(\mathbf{x}_i).$$

This ensures that the refined latent vector remains as close as possible to the original one, while restoring the agreement between surrogate and black-box predictions. Notably, at least the solution $(\gamma_W^* = 1, \gamma_x^* = 1)$ exists because $\mathbf{z}_i^{(\gamma_W=1,\gamma_x=1)} \equiv \mathbf{z}_{nn}$ and $g(\mathbf{z}_{nn}) \equiv b(\mathbf{x}_i)$.

The refined latent instance is then given by $\mathbf{z}_i^* = W_i^* \mathbf{x}_i + \boldsymbol{\varepsilon}_i^*$, where $W_i^* = W_i^b + \gamma_W^*(W_{nn}^b - W_i^b) \equiv W_i^b + \Delta W_i$ is a updated transformation matrix and $\boldsymbol{\varepsilon}_i^* = \gamma_x^* W_{nn}^b (\mathbf{x}_{nn} - \mathbf{x}_i) \equiv W_{nn}^b \delta$ is an induced offset. In practice, we approximate the solution via grid-search over $(0, 1]^2$. Among feasible solutions, we select the one minimizing $\gamma_W + \gamma_x$, favoring minimal corrections in both transformation and input space. Overall, the refinement corresponds to the smallest local perturbation of the latent mapping that restores surrogate–black-box agreement.

This approach enables the construction of a valid explanation even for those instances where the surrogate is not capable to replicate black-box label. Specifically, for feature importance explanations, the attribution vectors will be linearly updated:

$$\psi^*_{i,j} = \sum_{r=1}^{k} \beta_r (W^b_{i,j,r} + \Delta W_{i,j,r}) \equiv \psi_{i,j} + \Delta\psi_{i,j}$$

Furthermore, for decision rule explanations, we will have corrected oblique rules with translated upper/lower bounds as:

$$\tilde{\rho}^*_i = \Big\{ \sum_{j=1}^{m} (W^b_{i,j,r} + \Delta W_{i,j,r}) x_{i,j} \in [l^z_r - \varepsilon^*_{i,r}, u^z_r - \varepsilon^*_{i,r}] \Big\}_{r=1\ldots k}$$

Once converted to axis-aligned rules $\rho^*_i = \{x_{i,j} \in [l^*_{i,j}, u^*_{i,j}]\}_{j=1\ldots m}$, they will satisfy the following:

$$l^*_{i,j} - x_{i,j} = \max_r \frac{l^z_r - z^*_{i,r}}{W^b_{i,j,r} + \Delta W_{i,j,r}}$$

$$u^*_{i,j} - x_{i,j} = \min_r \frac{u^z_r - z^*_{i,r}}{W^b_{i,j,r} + \Delta W_{i,j,r}}.$$

E. Counterfactual Rules and Counter-Examples

While decision rules are directly obtained from the root-to-leaf paths of a decision tree, counterfactual rules are derived through symbolic reasoning applied to the same tree. Following the methodology proposed in [7], we analyze the latent decision tree (already used to generate the decision rules) to identify paths leading to a prediction opposite to that of the input instance.

For each test instance, we first rank training instances whose surrogate tree predictions differ from that of the test instance based on the cosine similarity between their latent representations. We then examine the decision rules associated with these instances and select those requiring the minimal number of split condition changes with respect to the rule satisfied by the test instance. This strategy identifies the closest instances that achieve the desired prediction change with the smallest possible modifications, thereby optimizing both proximity and sparsity, which are key properties of high-quality counterfactual explanations [21].

Finally, we also return the corresponding training instances from which the counterfactual rules are derived, providing them as counterfactual examples.

F. Analysis of Stability Loss

Here, we provide an intuition behind the stability loss computation, which guarantees that small changes in the original data space lead to minimal perturbations in the latent space, thus ensuring a consistent and reliable mapping across data points. We enforce each local transformation to remain valid when applied to slightly perturbed input. Specifically, omitting b for simplicity, if the encoding is given by $\mathbf{z} = \eta(\mathbf{x}) = f(\mathbf{x}) \cdot \mathbf{x}$, where $f(\mathbf{x})$ represents the individual transformation for $\mathbf{x}$, then the same transformation must hold for a small perturbation $\mathbf{x} + \delta$. In other words, the encoding operates locally around the instance $\mathbf{x}$ as a linear transformation characterized by a matrix $f(\mathbf{x})$, which is approximately constant within the neighborhood of $\mathbf{x}$. While the coefficients of the matrix dynamically adapt to the input, their rate of variation is slower than that of the input $\mathbf{x}$, ensuring stability and consistency of the transformation. This implies the following constraint:

$$\eta(\mathbf{x} + \delta) = f(\mathbf{x} + \delta) \cdot (\mathbf{x} + \delta) \approx f(\mathbf{x}) \cdot (\mathbf{x} + \delta).$$

We enforce this property by minimizing specific quantities which ensure the validity of the equation above. First, we show the first-order Taylor expansions of matrix entries of the local transformation f under small input perturbations:

$$f_{j,r}(\mathbf{x} + \delta) = f_{j,r}(\mathbf{x}) + \sum_{v=1}^{m} \frac{\partial f_{j,r}(\mathbf{x})}{\partial x_v} \delta_v + \mathcal{O}(||\delta||^2)$$

Substituting these approximations into the expression for the encoding of the perturbation, $\eta(\mathbf{x} + \delta) = f(\mathbf{x} + \delta) \cdot (\mathbf{x} + \delta)$, for each entry of the vector we get (neglecting second-order terms):

$$\begin{aligned}
\eta_r(\mathbf{x} + \delta) &\approx \sum_{j=1}^{m} \Big(f_{j,r}(\mathbf{x}) + \sum_{v=1}^{m} \frac{\partial f_{j,r}(\mathbf{x})}{\partial x_v} \delta_v \Big) (x_j + \delta_j) \\
&= \sum_{j=1}^{m} f_{j,r}(\mathbf{x})(x_j + \delta_j) + \\
&\quad + \sum_{j=1}^{m} \Big(\sum_{v=1}^{m} \frac{\partial f_{v,r}(\mathbf{x})}{\partial x_j} x_v \Big) \delta_j + \mathcal{O}(||\delta||^2).
\end{aligned}$$

Last equation tells us that describing the variation of the latent encoding η, when applied to a minimal perturbation of $\mathbf{x}$, requires the sum of two quantities (up to second-order corrections):

$$\eta(\mathbf{x}+\delta) \approx f(\mathbf{x})(\mathbf{x}+\delta) + D(\mathbf{x}) \cdot \delta \qquad \Big[D_{j,r} = \sum_v \frac{\partial f_{v,r}}{\partial x_j} x_v \Big]$$

The first term, $f(\mathbf{x}) \cdot (\mathbf{x} + \delta)$, represents the mapping of the perturbation δ applied to the input $\mathbf{x}$, with the linear transformation f held constant. The second term, $D(\mathbf{x}) \cdot \delta$, captures the change in the transformation due to the perturbation δ and its interaction with the input $\mathbf{x}$. Therefore, we enforce stability by minimizing $||D(\mathbf{x})||^2_{\mathrm{F}}$ during training. Reordering the terms $\eta(\mathbf{x} + \delta) \approx f(\mathbf{x}) \cdot \mathbf{x} + [f(\mathbf{x}) + D(\mathbf{x})] \cdot \delta$, we obtain the Taylor expansion of the encoding η under small input perturbations:

$$\eta(\mathbf{x} + \delta) \approx \eta(\mathbf{x}) + J(\mathbf{x}) \cdot \delta \qquad \Big[J(\mathbf{x}) = f(\mathbf{x}) + D(\mathbf{x}) \Big]$$

where we highlight the Jacobian matrix J around the datapoint $\mathbf{x}$, with entries $J_{j,r} = \frac{\partial \eta_r}{\partial x_j}$. To minimize $||D(\mathbf{x})||^2_{\mathrm{F}}$, we optimize the reported loss involving Jacobian matrix $L^{st}(\mathbf{x}) = ||J(\mathbf{x}) - f(\mathbf{x})||^2_{\mathrm{F}}$ for each instance.

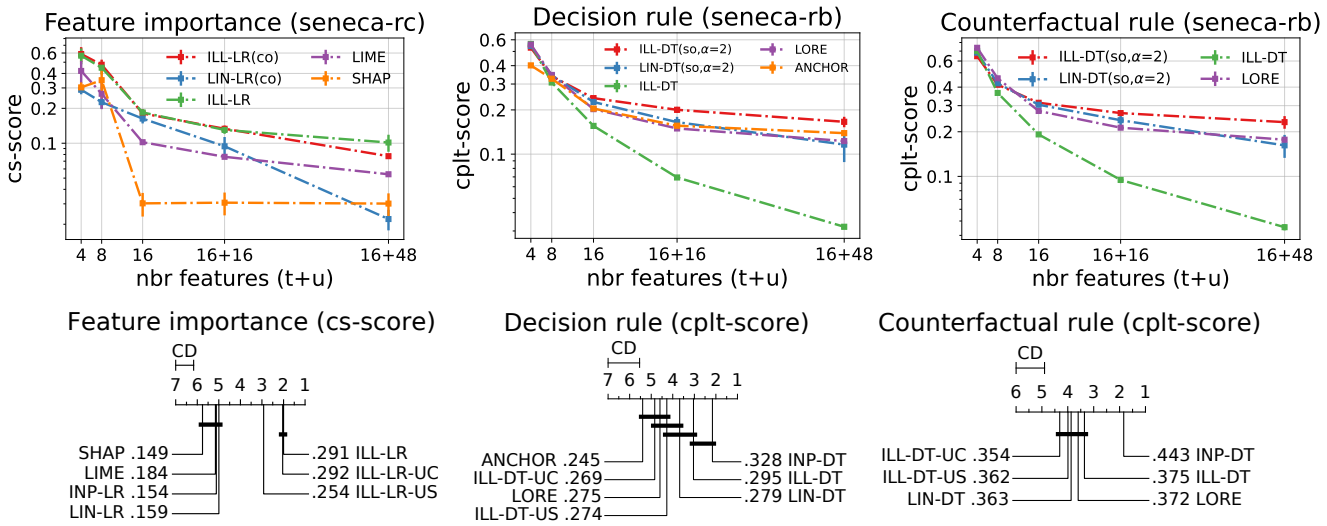


Fig. A4: Explanation correctness for synthetic classifiers. In the Figures, average metrics for correctness in feature importance and decision rules are reported varying the input feature size. In the CD plots, average rankings for individual metrics are reported for different methods and datasets.

TABLE A1: Feature-based latent space quality metrics for all datasets and methods.

	Feature Preservation						**KNN Gain**					
	ILL-UC	LIN-UC	PCA	ISOMAP	LLE	UMAP	ILL-UC	LIN-UC	PCA	ISOMAP	LLE	UMAP
aids	.842	.851	**1.000**	.706	.527	.683	.946	.960	**1.008**	.896	.791	.895
austr	.852	.855	**.993**	.790	.612	.732	**1.009**	1.001	1.000	.984	.960	.984
bank	.845	.843	**.952**	.786	.594	.666	.963	.963	.959	.938	.990	**.998**
breast	**1.000**	.996	**1.000**	.928	.746	.849	1.000	**1.010**	1.000	1.001	1.001	1.001
churn	.908	.861	**.919**	.668	.521	.679	1.078	1.026	**1.234**	.977	.960	1.000
compas	.859	.870	**1.000**	.724	.563	.643	.985	.994	1.002	1.019	.976	**1.020**
ctg	.939	.927	**1.000**	.716	.635	.678	**1.033**	1.023	1.000	1.000	.996	.996
diabetes	**1.000**	**1.000**	**1.000**	.845	.700	.792	**1.019**	1.000	1.000	.964	.848	.894
ecoli	**1.000**	**1.000**	**1.000**	.871	.806	.785	**1.024**	1.007	1.000	.977	.950	.961
fico	**1.000**	**1.000**	**1.000**	.832	.657	.680	**1.006**	1.003	1.000	.984	.964	.990
german	.849	.832	**.945**	.765	.589	.686	.994	.963	1.054	**1.067**	1.034	.992
home	**1.000**	**1.000**	**1.000**	.897	.679	.826	**1.033**	1.000	1.011	1.011	.956	.966
ionos	.994	.992	**1.000**	.868	.713	.749	**1.053**	1.045	1.052	1.006	1.004	1.006
sonar	.986	.967	**.995**	.871	.671	.800	1.003	.977	**1.027**	.890	.998	.946
spam	.921	.930	**.953**	.816	.585	.673	**1.016**	1.015	.994	.975	.924	.977
titanic	.868	.869	**1.000**	.896	.603	.797	**1.047**	**1.047**	1.000	1.014	.977	1.002
wine	**1.000**	**1.000**	**1.000**	.868	.670	.789	**1.063**	1.000	1.000	.974	.890	.933
yeast	.999	**1.000**	**1.000**	.887	.652	.799	.995	**1.000**	**1.000**	.965	.746	.870

G. Detailed Experimental Results

Here, we report in details all the results that are shown in aggregated form in the main paper.

RQ1 - Goodness of Latent Space

In Tables A1, A4 and A5 are reported extensive results for all metrics, datasets and black-boxes that have been summarized in the main paper to answer **RQ1**. In particular, central and right columns of Tables A4 and A5 report Macro-F1 accuracies for LR and DT trained on latent spaces both with and without black-box decision conditioning. The same results are aggregated and ranked in Figure A5. Firstly, we observe that models without conditioning consistently achieve lower accuracies in surrogate classification tasks. Secondly, particularly for the LR surrogate, ILLUME demonstrates significant improvements compared to LIN encodings. This underscores the importance of label conditioning, combined with using more expressive latent features, in enhancing the accuracy of surrogate predictions.

RQ2 - Correctness for Synthetic Black-Box Explanations

In Figure A4 we report additional results from synthetic black-box models. We also present preliminary results on generating **counterfactual explanations** using the approach described in E.

In the top pictures, we display with line plots the variation of explanation accuracy metrics with respect to the number of input dimensions, comparing the best-regularized setup of ILLUME with the un-regularized one. For feature importance explanations, we observe that the non-collinear (*co*) regularizer slightly enhances the correctness of the explanations across

most scenarios. However, in cases involving 16+48 input features, the explanation accuracy experiences a minor decline. In contrast, for both factual and counterfactual rules, the combination of sparsity (α=2) and soft-orthogonality (so) significantly improves the correctness of the explanations. These findings underscore the importance of consistently applying sparsity regularizers within ILLUME when generating rule-based explanations.

In the CD plots, corresponding to the results presented for **RQ2** in the main paper, we display aggregated rankings for each metric across all datasets. We observe the same trends as reported in the main paper: ILLUME and its variants perform similarly in terms of feature importance correctness, while ILLUME-DT shows comparable performance to INP-DT in factual rules correctness. Regarding counterfactual rules, there is no statistically significant difference between ILLUME-DT and LORE. However, INP-DT outperforms the others, consistent with its superior performance on factual rules.

RQ3 - Faithfulness and Robustness in Real-World Datasets

Tables A6 and A7 present comprehensive results for feature importance metrics, while Tables A8 and A9 display detailed results for decision rules metrics. These tables include all datasets and black-box models that were summarized for answering **RQ3** in the main paper.

Figure A6 presents additional findings that compare the effects of different regularizations on the robustness and faithfulness metrics for feature importance explanations. We observe that sparse regularizations (so, α=2) do not enhance either robustness or faithfulness for feature importance. Additionally, models regularized for non-collinearity (co) perform similarly to unregularized models, although they achieve the highest average robustness scores. In contrast, Figures A7 show additional results comparing the impact of various regularizations on the robustness and faithfulness metrics for decision rules. Here, sparse regularizations (so, α=2) significantly improve the quality of decision rules for both metrics, thereby confirming our earlier observations regarding sparsity and decision rule correctness. Overall, for both types of explanations, models optimized for stability and label conditioning perform significantly worse in terms of robustness and faithfulness, respectively.

Tables A10 and Tables A11 present additional results on explainers' **robustness** with the evaluation of a **global metric** rather than local metrics based on sensitivity to perturbations. Intuitively, a –globally–robust explainer should produce similar explanations for closely located data points – with the same back-box predicted label – and distinct explanations for those farther apart. Thus, robustness is assessed globally by calculating the Spearman's rank correlation coefficient between the pairwise distances of explanations, $\{d_{\mathcal{E}}(i,j)\}_{i<j}$, and the corresponding pairwise distances of input records, $\{||\mathbf{x}_i - \mathbf{x}_j||_2)\}_{i<j}$, for those pairs of points with accordant predicted labels. Like in SENECA, we use similarity metrics $cs\text{-}score(\cdot,\cdot)$ and $cplt\text{-}score(\cdot,\cdot)$ instead of distance metrics. Figure A8 also presents the aggregate performance of feature importance explainers. In this analysis, ILLUME-LR achieves the highest performance, while ILLUME-LR-US performs comparably. This is in contrast to the local robustness results reported in the main paper, where ILLUME-LR-US was significantly worse. Additionally, LIME demonstrates the poorest performance in this context, despite showing adequate local robustness in the main paper. Regarding decision rule explainers, LIN-LR experiences a decline in performance compared to the local robustness results and ranks similarly to both ILLUME-DT and ILLUME-DT-US.

TABLE A2: Ranking correctness of feature importance. Prediction accuracy of surrogate models inside parentheses.

	Feature Ranking Correctness (*spearman*)		
LGB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**1.000**±.000 (98.6)	.184±.038 (92.3)	.204±.039 (81.7)
ILL-LR-UC(co)	**1.000**±.000 (98.5)	.185±.054 (91.8)	.202±.043 (81.7)
ILL-LR-US(co)	.904±.041 (98.7)	**.473**±.107 (96.3)	**.235**±.079 (82.1)
LIN-LR(co)	.836±.009 (97.2)	.033±.001 (89.1)	.117±.013 (80.4)
INP-LR	.806±.006 (96.6)	.032±.001 (89.0)	.110±.012 (80.1)
LIME	.993±.001	.009±.001	.082±.003
SHAP	.257±.001	.197±.010	.163±.020
XGB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**1.000**±.000 (98.0)	.264±.043 (91.6)	.197±.019 (82.7)
ILL-LR-UC(co)	**1.000**±.000 (98.3)	.186±.055 (91.8)	.199±.039 (82.7)
ILL-LR-US(co)	.947±.028 (98.1)	**.676**±.095 (96.0)	**.458**±.109 (83.3)
LIN-LR(co)	.847±.008 (97.0)	.036±.004 (89.0)	.117±.012 (81.4)
INP-LR	.812±.009 (96.1)	.033±.001 (88.7)	.108±.013 (81.2)
LIME	.995±.002	.010±.001	.081±.005
SHAP	.276±.008	.211±.007	.155±.018
CTB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**1.000**±.000 (98.6)	.165±.048 (90.6)	**.278**±.021 (82.7)
ILL-LR-UC(co)	**1.000**±.000 (98.4)	.168±.055 (91.1)	.203±.042 (82.7)
ILL-LR-US(co)	.965±.015 (98.6)	**.542**±.050 (97.2)	.147±.025 (82.9)
LIN-LR(co)	.824±.006 (97.3)	.038±.006 (89.0)	.116±.011 (81.6)
INP-LR	.806±.006 (96.9)	.033±.002 (88.9)	.111±.011 (81.4)
LIME	.993±.001	.008±.001	.084±.002
SHAP	.259±.012	.218±.009	.122±.007

H. Additional Experimental Results on Synthetic Data

In this section, we further extend the evaluation of explanation methods on synthetic benchmarks beyond SENECA framework. In line with [56], we adopt synthetic nonlinear functions derived from the Friedman model [66] and adapted for classification tasks. Concretely, we employ publicly available datasets[22] ssin-2c, int2-3c and int2-8c respectively with two, three and eight classes. They consists of $n = 1000$ synthetic instances described by 4 continuous features $\{x_1, x_2, x_3, x_4\}$. Instead of providing ground-truth importance values $\{\hat{\psi}_1, \hat{\psi}_2, \hat{\psi}_3, \hat{\psi}_4\}$, in [56] the authors provides importance rankings: in ssin-2c, the ranking is $\hat{\psi}_1 > \hat{\psi}_2 > \hat{\psi}_3 > \hat{\psi}_4$; in int2-3c and int2-8c, the ranking is $(\hat{\psi}_1, \hat{\psi}_2) > \hat{\psi}_3 > \hat{\psi}_4$. Using the same experimental setting adopted for the SENECA benchmarks, we evaluate local explanation methods with respect to ranking quality (in Table A2 via Spearman correlation) and importance proximity (in Table A3 via the already used *cs-score*) across all test instances and five independent train–test

[22] http://www3.dsi.uminho.pt/pcortez/data

TABLE A3: Correctness of synthetic explanations. Prediction accuracy of surrogate models inside parentheses.

	Feature Importance Correctness (*cs-score*)		
LGB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**.998**±.000 (98.6)	.139±.042 (92.3)	.210±.012 (81.7)
ILL-LR-UC(co)	**.998**±.000 (98.5)	.123±.050 (91.8)	.211±.017 (81.7)
ILL-LR-US(co)	.963±.011 (98.7)	**.332**±.027 (96.3)	**.305**±.074 (82.1)
LIN-LR(co)	.899±.004 (97.2)	.050±.003 (89.1)	.249±.012 (80.4)
INP-LR	.891±.004 (96.6)	.047±.003 (89.0)	.248±.012 (80.1)
LIME	.993±.001	.000±.000	.259±.017
SHAP	.223±.011	.315±.015	.142±.021
XGB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**.996**±.001 (98.0)	.176±.050 (91.6)	.219±.021 (82.7)
ILL-LR-UC(co)	**.997**±.001 (98.3)	.124±.054 (91.8)	.213±.020 (82.7)
ILL-LR-US(co)	.967±.010 (98.1)	**.411**±.028 (96.0)	**.305**±.104 (83.3)
LIN-LR(co)	.903±.005 (97.0)	.053±.005 (89.0)	.254±.014 (81.4)
INP-LR	.893±.005 (96.1)	.049±.002 (88.7)	.254±.013 (81.2)
LIME	.992±.001	.000±.000	.269±.017
SHAP	.236±.006	.321±.008	.137±.023
CTB	ssin-2c	int2-3c	int2-8p
ILL-LR(co)	**.998**±.001 (98.6)	.100±.050 (90.6)	.221±.012 (82.7)
ILL-LR-UC(co)	**.998**±.001 (98.4)	.117±.052 (91.1)	.211±.017 (82.7)
ILL-LR-US(co)	.989±.003 (98.6)	**.357**±.015 (97.2)	.230±.058 (82.9)
LIN-LR(co)	.899±.005 (97.3)	.054±.007 (89.0)	**.251**±.013 (81.6)
INP-LR	.891±.004 (96.9)	.044±.001 (88.9)	**.250**±.013 (81.4)
LIME	.993±.001	.000±.000	.244±.012
SHAP	.231±.014	.308±.004	.082±.011

splits in every dataset. To evaluate correctness scores, we set ground-truth importance values aligned with reference rankings as following: in ssin-2c, we define $\hat{\psi} = \{4., 2., 1., 0.\}$; in int2-3c and int2-8c, we define $\hat{\psi} = \{3., 3., 1., 0.\}$. Because these synthetic datasets do not provide white-box predictive functions like in SENECA, we fit ensemble-tree black-boxes to mimic the underlying decision process, and then explain their local predictions. In multi-class datasets, explanations are generated for the majority class. The reported results reveal that ILLUME consistently outperforms competing explainers. On ssin-2c, ILL-LR and ILL-LR-UC achieve perfect rankings and similarities, with LIME as the second best approach. For the int2 datasets, ILL-LR and ILL-LR-US mostly attains the best results with XGB and LGB, consistently outperforming LIME and treeSHAP. When CTB is explained, however, the global linear baselines LIN-LR and INP-LR yield more aligned explanations on int2-8p. Overall, these findings underscore the value of carefully designing synthetic benchmarks for explanation methods evaluation.

TABLE A4: Decision-based latent space quality metrics with LGB as black-box for all datasets and methods.

LGB	**Decision Preservation**				**LR Accuracy** (Macro-F1)				**DT Accuracy** (Macro-F1)			
	ILL	LIN	ILL-UC	LIN-UC	ILL	LIN	ILL-UC	LIN-UC	ILL	LIN	ILL-UC	LIN-UC
aids	**.618**	.596	.549	.555	**.952**	.902	.899	.902	**.930**	.874	.768	.780
austr	**.743**	.729	.613	.623	.920	**.927**	.919	.920	.927	**.942**	.869	.913
bank	**.659**	.656	.595	.588	**.901**	.866	.839	.840	**.881**	**.881**	.759	.736
breast	**.779**	.739	.728	.718	.981	.990	.981	**.991**	.972	.963	.981	**.991**
churn	**.610**	.583	.542	.548	**.875**	.637	.650	.623	**.852**	.827	.738	.760
compas	**.667**	.665	.644	.640	**.926**	.914	.902	.914	**.913**	.893	.891	.891
ctg	**.693**	.689	.624	.611	.993	**.996**	**.996**	.993	.993	**.996**	.969	.949
diabetes	**.642**	.640	.621	.627	**.906**	.871	.881	.871	**.874**	.832	.836	.864
ecoli	**.726**	.700	.706	.706	**.970**	.936	.947	.936	**.939**	**.939**	.910	.921
fico	**.642**	.578	.586	.580	**.908**	.903	.894	.903	.862	**.869**	.843	.854
german	**.669**	.627	.566	.557	.814	.807	**.816**	.806	**.805**	.791	.699	.707
home	**.677**	.626	.614	.614	**.949**	**.949**	**.949**	**.949**	**.970**	.939	.929	.949
ionos	.631	**.665**	.586	.597	.886	.906	**.934**	.854	.919	.952	**.968**	.907
sonar	**.671**	.629	.576	.610	**.904**	.857	.810	.833	**.881**	.833	.785	.786
spam	**.704**	.647	.674	.623	**.963**	.942	.942	.936	**.955**	.946	.913	.903
titanic	.743	**.744**	.650	.672	.885	**.893**	**.893**	.885	**.937**	.919	.922	.931
wine	**.550**	.546	.536	.540	**.305**	.266	.292	.286	.516	**.537**	.497	.440
yeast	**.630**	.615	.609	.623	**.629**	.565	.612	.572	**.556**	.534	.525	.512

TABLE A5: Decision-based latent space quality metrics with XGB as black-box for all datasets and methods.

XGB	**Decision Preservation**				**LR Accuracy** (Macro-F1)				**DT Accuracy** (Macro-F1)			
	ILL	LIN	ILL-UC	LIN-UC	ILL	LIN	ILL-UC	LIN-UC	ILL	LIN	ILL-UC	LIN-UC
aids	**.661**	.610	.563	.557	**.941**	.871	.880	.868	**.938**	.877	.730	.774
austr	.739	**.751**	.657	.652	**.934**	.913	.912	.912	**.941**	.911	.898	.898
bank	.653	**.662**	.593	.582	**.893**	.868	.832	.853	**.886**	.873	.750	.744
breast	**.821**	.737	.775	.730	.981	**.990**	.981	**.990**	.972	.972	.961	**1.000**
churn	**.602**	.587	.536	.541	**.872**	.660	.649	.644	**.852**	.786	.728	.728
compas	.671	**.673**	.640	.638	**.933**	.915	.911	.915	**.915**	.912	.891	.888
ctg	**.743**	.739	.653	.629	**1.000**	.996	.996	.996	.957	.945	**1.000**	.945
diabetes	**.684**	.649	.625	.626	**.910**	.871	.879	.871	.857	**.895**	.849	.857
ecoli	.721	.703	**.726**	.709	**1.000**	.969	.988	.980	.980	**.992**	.947	.980
fico	**.650**	.588	.592	.588	**.937**	.929	.922	.929	.886	**.897**	.882	.874
german	**.655**	.617	.535	.516	.789	**.810**	.761	.778	.780	**.791**	.714	.697
home	**.663**	.642	.630	.626	.939	**.949**	.939	.939	.929	**.970**	.939	.929
ionos	.566	**.603**	.575	.572	.891	.868	**.904**	.883	.874	**.954**	.936	.880
sonar	.619	**.629**	.595	**.629**	.786	.785	.810	**.833**	**.857**	.762	.786	.762
spam	**.721**	.667	.675	.631	**.968**	.947	.942	.940	**.966**	.952	.919	.901
titanic	**.725**	.687	.655	.694	.877	**.890**	.883	.877	**.923**	.921	.921	.910
wine	**.587**	.574	.562	.566	**.334**	.276	.319	.287	.422	**.544**	.441	.494
yeast	**.700**	.695	.690	.694	**.790**	.755	.776	.752	.676	**.698**	.644	.656

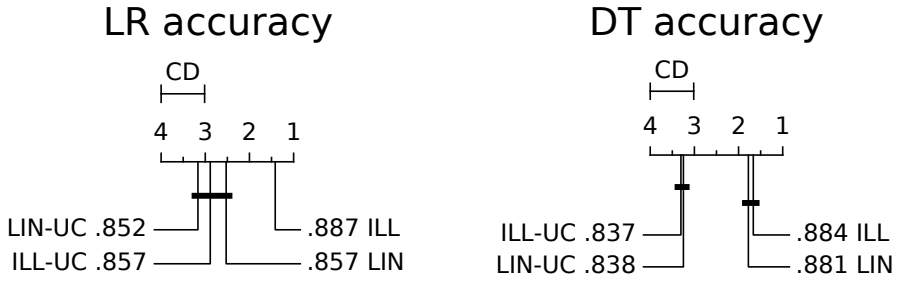


Fig. A5: Critical Difference plots with Nemenyi test at 90% confidence level for prediction accuracy metrics in real-world datasets, comparing LR and DT performances over different training feature spaces.

TABLE A6: Robustness and faithfulness metrics for feature importance methods in all datasets with LGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

LGB	**Feature Importance Robustness**						**Feature Importance Faithfulness**					
	ILL-LR	ILL-LR-US	LIN-LR	INP-LR	LIME	SHAP	ILL-LR	ILL-LR-UC	LIN-LR	INP-LR	LIME	SHAP
aids	**.994** (95.3)	.797 (97.0)	.582 (93.7)	.244 (93.7)	.978	.445	**.908** (97.4)	.234 (93.5)	.540 (93.5)	.610 (93.7)	.054	.806
austr	**.980** (93.5)	.901 (89.9)	.659 (92.0)	.500 (92.0)	.904	.443	.785 (92.0)	.520 (92.0)	.561 (89.9)	.659 (92.0)	.033	**.798**
bank	**.994** (96.1)	.958 (96.0)	.536 (94.9)	.271 (95.0)	.916	.127	**.790** (96.8)	.304 (95.4)	.537 (95.0)	.540 (95.0)	.122	.410
breast	.777 (98.2)	.811 (97.4)	.388 (99.1)	.219 (98.2)	**.964**	.328	.846 (99.1)	.859 (98.2)	.819 (97.4)	.833 (98.2)	.113	**.906**
churn	**.981** (95.8)	.932 (94.8)	.719 (87.3)	.672 (87.6)	.618	.160	**.679** (96.6)	.142 (89.1)	.055 (87.4)	.247 (87.6)	.091	.644
compas	**.993** (94.7)	.711 (95.1)	.711 (94.4)	.635 (94.4)	.971	.538	**.798** (95.8)	.317 (94.0)	.319 (94.4)	.313 (94.4)	.015	.448
ctg	**.980** (99.8)	.840 (99.8)	.758 (99.5)	.617 (99.3)	.781	.656	**.741** (99.8)	.357 (99.5)	.624 (99.5)	.519 (99.3)	.487	.682
diabetes	.966 (92.9)	.797 (94.2)	.180 (89.6)	.176 (89.6)	**.985**	.163	.396 (92.9)	.298 (90.9)	.454 (89.6)	.450 (89.6)	.021	**.580**
ecoli	.958 (100.)	.961 (100.)	.735 (100.)	.673 (100.)	**.981**	.585	**.640** (100.)	.623 (100.)	.575 (100.)	.586 (100.)	.260	.503
fico	**.981** (90.3)	.899 (90.8)	.400 (90.3)	.324 (90.3)	.918	.268	.443 (91.4)	.338 (89.5)	.379 (90.3)	.377 (90.3)	.000	**.600**
german	**.996** (89.0)	.930 (85.5)	.394 (85.5)	.338 (88.5)	.840	.069	**.531** (87.5)	.198 (88.0)	.174 (85.5)	.190 (88.5)	.124	.238
home	.878 (96.0)	.756 (97.0)	.204 (94.9)	.122 (94.9)	**.976**	.178	.741 (97.0)	.337 (94.9)	.495 (94.9)	.486 (94.9)	.077	**.752**
ionos	.845 (94.4)	.688 (94.4)	.371 (88.7)	.353 (88.7)	**.932**	.366	.385 (93.0)	.153 (94.4)	.568 (91.5)	.576 (88.7)	.193	**.841**
sonar	.621 (85.7)	.698 (92.9)	.104 (83.3)	.085 (83.3)	**.914**	.078	.211 (90.5)	.055 (88.1)	.201 (88.1)	.077 (83.3)	.032	**.469**
spam	**.923** (96.0)	.859 (96.2)	.396 (94.0)	.311 (94.4)	.808	.166	**.848** (96.4)	.441 (94.5)	.522 (94.6)	.037 (94.4)	.000	.662
titanic	**.991** (90.5)	.919 (90.5)	.616 (89.9)	.646 (89.4)	.971	.536	**.769** (91.6)	.640 (90.5)	.639 (89.9)	.641 (89.4)	.389	.733
wine	**.943** (64.2)	.640 (67.2)	.192 (59.1)	.176 (58.6)	.779	.166	**.152** (72.3)	.105 (62.8)	.135 (58.8)	.146 (58.6)	.022	.104
yeast	**.934** (76.4)	.705 (75.1)	.323 (69.4)	.305 (69.4)	.289	.128	**.280** (78.5)	.143 (74.1)	.216 (70.7)	.217 (69.4)	.018	.177

TABLE A7: Robustness and faithfulness metrics for feature importance methods in all datasets with XGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

XGB	**Feature Importance Robustness**						**Feature Importance Faithfulness**					
	ILL-LR	ILL-LR-US	LIN-LR	INP-LR	LIME	SHAP	ILL-LR	ILL-LR-UC	LIN-LR	INP-LR	LIME	SHAP
aids	**.994** (95.3)	.901 (95.1)	.505 (91.1)	.328 (91.1)	.965	.393	**.838** (95.8)	.252 (91.8)	.654 (91.1)	.574 (91.1)	.083	.799
austr	**.987** (92.0)	.872 (92.8)	.614 (90.6)	.514 (90.6)	.929	.341	**.783** (94.2)	.560 (91.3)	.704 (90.6)	.698 (90.6)	.000	.737
bank	**.989** (96.2)	.929 (96.0)	.503 (95.3)	.260 (95.5)	.910	.213	**.826** (97.1)	.397 (94.9)	.352 (93.8)	.473 (95.5)	.269	.765
breast	.875 (99.1)	.803 (99.1)	.451 (100.)	.502 (97.4)	**.970**	.302	.888 (98.2)	.883 (98.2)	.850 (98.2)	.864 (97.4)	.128	**.927**
churn	**.974** (95.1)	.840 (95.1)	.776 (88.0)	.680 (87.9)	.549	.163	**.612** (95.8)	.128 (89.1)	.154 (88.5)	.197 (87.9)	.078	.611
compas	**.989** (95.4)	.742 (95.2)	.713 (94.5)	.597 (94.5)	.971	.532	**.731** (96.1)	.320 (94.4)	.330 (94.5)	.321 (94.5)	.004	.525
ctg	**.992** (100.)	.859 (99.8)	.670 (99.8)	.477 (99.8)	.824	.502	**.776** (100.)	.323 (99.8)	.688 (99.8)	.434 (99.8)	.066	.696
diabetes	.971 (92.2)	.780 (92.9)	.225 (90.3)	.208 (87.0)	**.984**	.171	.390 (92.9)	.273 (90.3)	.517 (89.0)	.507 (87.0)	.048	**.671**
ecoli	.973 (100.)	.945 (100.)	.722 (100.)	.659 (98.5)	**.990**	.726	**.720** (100.)	.711 (100.)	.683 (100.)	.691 (98.5)	.358	.516
fico	**.973** (93.4)	.944 (93.2)	.433 (92.9)	.368 (92.9)	.971	.304	.516 (95.3)	.409 (92.3)	.453 (92.9)	.450 (92.9)	.000	**.672**
german	**.996** (87.0)	.953 (87.0)	.512 (86.0)	.320 (86.0)	.782	.071	**.492** (89.0)	.131 (86.5)	.167 (82.5)	.200 (86.0)	.005	.358
home	.956 (93.9)	.912 (96.0)	.162 (93.9)	.141 (93.9)	**.974**	.148	.636 (97.0)	.277 (94.9)	.474 (93.9)	.470 (93.9)	.027	**.712**
ionos	.772 (91.5)	.734 (91.5)	.382 (87.3)	.351 (90.1)	**.950**	.322	.244 (97.2)	.189 (91.5)	.533 (88.7)	.551 (90.1)	.162	**.914**
sonar	.784 (81.0)	.686 (81.0)	.090 (76.2)	.069 (78.6)	**.935**	.168	.183 (83.3)	.137 (83.3)	.124 (81.0)	.084 (78.6)	.042	**.633**
spam	**.912** (96.5)	.819 (96.9)	.291 (94.5)	.141 (94.6)	.878	.140	**.823** (97.0)	.462 (94.5)	.421 (94.6)	.265 (94.6)	.013	.721
titanic	**.993** (91.1)	.921 (91.1)	.662 (89.4)	.608 (89.9)	.973	.633	.571 (90.5)	.671 (89.9)	.638 (88.8)	.634 (89.9)	.155	**.809**
wine	**.952** (66.3)	.581 (66.6)	.183 (60.4)	.174 (59.3)	.856	.278	**.223** (72.9)	.165 (64.3)	.206 (60.1)	.209 (59.3)	.042	.107
yeast	**.964** (86.5)	.795 (88.2)	.391 (81.5)	.394 (81.8)	.941	.813	**.528** (88.2)	.353 (84.2)	.325 (81.5)	.311 (81.8)	.152	.290

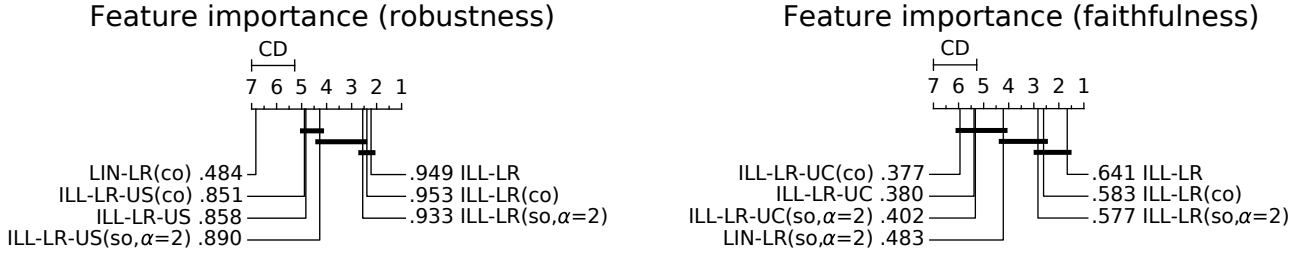


Fig. A6: Critical Difference plots with Nemenyi test at 90% confidence level for robustness and faithfulness metrics in real-world datasets, comparing various regularized versions of ILLUME-LR.

TABLE A8: Robustness and faithfulness metrics for decision rules methods in all datasets with LGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

LGB	**Decision Rule Robustness**						**Decision Rule Faithfulness**					
	ILL-DT	ILL-DT-US	LIN-DT	INP-DT	LORE	ANCHOR	ILL-DT	ILL-DT-UC	LIN-DT	INP-DT	LORE	ANCHOR
aids	.676 (97.4)	.301 (96.7)	**.911** (97.9)	.486 (98.8)	.060	.180	.739 (97.4)	.644 (96.5)	.825 (97.9)	.804 (98.8)	.762	**.868**
austr	.404 (92.0)	.438 (92.0)	**.573** (97.1)	.288 (94.9)	.147	.218	**.707** (92.0)	.653 (94.9)	.694 (97.1)	.661 (94.9)	.666	.672
bank	.595 (95.5)	.504 (95.5)	**.679** (95.3)	.600 (96.1)	.038	.053	.642 (95.5)	.566 (93.8)	**.734** (95.3)	.675 (96.1)	.510	.457
breast	.352 (98.2)	.114 (96.5)	.452 (98.2)	**.536** (96.5)	.077	.025	**.863** (98.2)	.788 (98.2)	.817 (96.5)	.736 (96.5)	.735	.381
churn	.618 (95.7)	.471 (95.2)	**.890** (97.3)	.358 (98.2)	.215	.099	**.609** (95.7)	.399 (91.6)	.560 (97.3)	.605 (98.2)	.605	.413
compas	.603 (95.1)	.351 (94.3)	**.961** (95.1)	.361 (95.8)	.115	.168	.584 (95.1)	.564 (94.5)	**.619** (95.1)	.536 (95.8)	.503	.592
ctg	.791 (99.3)	.677 (99.3)	**.975** (99.3)	.866 (99.3)	.317	.031	.777 (99.3)	.779 (99.1)	.743 (99.3)	**.839** (99.3)	.666	.335
diabetes	.116 (89.6)	.057 (87.7)	**.166** (89.6)	.098 (88.3)	.047	.097	.449 (89.6)	.545 (90.9)	.344 (89.6)	.295 (88.3)	.276	**.646**
ecoli	.632 (100.)	.569 (100.)	**.784** (100.)	.529 (100.)	.478	.462	.570 (100.)	.587 (100.)	.531 (100.)	.513 (100.)	.541	**.600**
fico	.374 (90.1)	.262 (89.5)	**.390** (89.5)	.200 (90.5)	.038	.167	.512 (90.1)	.425 (89.4)	.512 (89.5)	.481 (90.5)	.457	**.637**
german	**.618** (86.0)	.287 (85.5)	.562 (83.0)	.154 (81.0)	.058	.085	.327 (86.0)	.142 (87.0)	.112 (83.0)	.141 (81.0)	.118	**.369**
home	.064 (96.0)	.058 (96.0)	.051 (98.0)	**.150** (94.9)	.014	.073	.511 (96.0)	.420 (97.0)	.469 (98.0)	.485 (94.9)	.407	**.680**
ionos	.438 (98.6)	.297 (97.2)	**.570** (95.8)	.376 (95.8)	.204	.155	**.793** (98.6)	.722 (97.2)	.751 (95.8)	.697 (95.8)	.748	.645
sonar	.194 (83.3)	.147 (85.7)	**.515** (81.0)	.398 (83.3)	.108	.043	**.458** (85.7)	.444 (85.7)	.422 (81.0)	.453 (83.3)	.214	.441
spam	.402 (96.1)	.459 (95.1)	**.652** (94.5)	.249 (93.7)	.120	.113	**.767** (95.1)	.622 (92.8)	.701 (94.5)	.642 (93.7)	.253	.495
titanic	.628 (93.9)	.476 (95.0)	**.948** (96.1)	.687 (94.4)	.288	.420	.762 (95.0)	.748 (96.1)	.792 (96.1)	.702 (94.4)	.697	**.795**
wine	.296 (72.2)	.230 (70.1)	**.525** (70.6)	.154 (70.2)	.134	.224	.097 (72.2)	.087 (71.4)	.099 (70.6)	.108 (70.2)	.214	**.388**
yeast	.568 (78.5)	.471 (77.8)	**.713** (76.4)	.362 (78.5)	.467	.418	.412 (78.5)	.367 (77.8)	.274 (75.4)	.319 (78.5)	**.427**	.358

TABLE A9: Robustness and faithfulness metrics for decision rules methods in all datasets with XGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

XGB	**Decision Rule Robustness**						**Decision Rule Faithfulness**					
	ILL-DT	ILL-DT-US	LIN-DT	INP-DT	LORE	ANCHOR	ILL-DT	ILL-DT-UC	LIN-DT	INP-DT	LORE	ANCHOR
aids	.666 (95.6)	.378 (93.0)	**.798** (94.2)	.098 (95.1)	.023	.153	.642 (95.6)	.518 (93.7)	.657 (94.2)	.645 (95.1)	.639	**.821**
austr	.513 (92.0)	.387 (92.8)	**.713** (92.8)	.262 (89.9)	.123	.256	**.780** (92.8)	.691 (90.6)	.587 (92.8)	.591 (89.9)	.697	.751
bank	.620 (95.9)	.528 (95.5)	**.767** (94.9)	.549 (95.1)	.022	.047	.728 (95.9)	.539 (95.6)	.551 (94.9)	**.743** (95.1)	.360	.465
breast	.243 (96.5)	.184 (96.5)	.465 (97.4)	**.563** (96.5)	.068	.032	**.813** (96.5)	.772 (97.4)	.769 (97.4)	.795 (96.5)	.703	.374
churn	.291 (96.1)	.625 (94.5)	**.895** (95.8)	.355 (99.0)	.229	.112	.517 (96.1)	.365 (93.4)	.504 (95.8)	**.571** (99.0)	.533	.394
compas	.811 (94.0)	.448 (94.2)	**.965** (94.5)	.385 (94.6)	.101	.154	**.606** (94.2)	.523 (93.4)	.576 (94.5)	.506 (94.6)	.489	.592
ctg	.869 (99.8)	.852 (100.)	**.976** (99.5)	.882 (99.5)	.340	.033	.787 (100.)	.727 (99.3)	.722 (99.5)	**.795** (99.5)	.654	.347
diabetes	.124 (91.6)	.062 (90.3)	.124 (90.9)	.176 (92.2)	.044	**.177**	.532 (91.6)	.563 (92.2)	.503 (90.9)	.411 (92.2)	.293	**.688**
ecoli	.680 (100.)	.533 (100.)	**.800** (100.)	.588 (100.)	.581	.497	.748 (100.)	.673 (100.)	**.753** (100.)	.666 (100.)	.623	.562
fico	.340 (93.8)	.279 (92.1)	**.413** (92.8)	.221 (93.8)	.053	.187	.617 (93.8)	.632 (93.0)	.614 (92.8)	.628 (93.8)	.643	**.679**
german	.337 (87.0)	.129 (82.5)	**.565** (81.5)	.107 (83.5)	.033	.081	.226 (87.0)	.169 (81.0)	.052 (81.5)	.041 (83.5)	.098	**.388**
home	.095 (96.0)	.037 (96.0)	.082 (97.0)	**.132** (94.9)	.010	.062	.578 (96.0)	.414 (97.0)	.555 (97.0)	.488 (94.9)	.455	**.730**
ionos	.385 (95.8)	.374 (94.4)	.514 (97.2)	**.615** (97.2)	.212	.232	.701 (95.8)	.683 (95.8)	.790 (97.2)	**.829** (97.2)	.710	.730
sonar	.211 (85.7)	.221 (85.7)	**.533** (88.1)	.254 (92.9)	.141	.072	.546 (85.7)	**.597** (85.7)	.324 (88.1)	.535 (92.9)	.427	.516
spam	.419 (96.1)	.483 (97.1)	**.699** (94.6)	.403 (94.7)	.130	.090	**.797** (97.1)	.643 (93.1)	.663 (94.6)	.675 (94.7)	.227	.478
titanic	.620 (95.5)	.469 (95.0)	**.955** (97.2)	.693 (96.1)	.387	.463	**.819** (95.5)	.690 (97.2)	.781 (97.2)	.737 (96.1)	.738	.807
wine	.303 (71.0)	.244 (71.0)	**.458** (70.6)	.162 (68.2)	.150	.229	.160 (71.0)	.148 (71.5)	.164 (70.6)	.145 (68.2)	.298	**.403**
yeast	.533 (89.2)	.437 (89.9)	**.788** (88.2)	.427 (89.9)	.469	.387	.590 (89.9)	.469 (87.5)	.470 (88.2)	.538 (89.9)	**.613**	.361

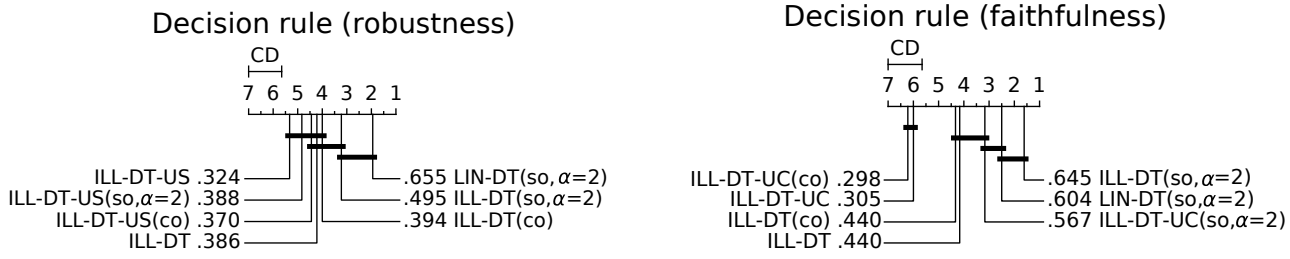


Fig. A7: Critical Difference plots with Nemenyi test at 90% confidence level for robustness and faithfulness metrics in real-world datasets, comparing various regularized versions of ILLUME-DT.

TABLE A10: Global robustness metrics in all datasets with LGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

LGB	**Feature Importance Robustness**						**Decision Rule Robustness**					
	ILL-LR	ILL-LR-US	LIN-LR	INP-LR	LIME	SHAP	ILL-DT	ILL-DT-UC	LIN-DT	INP-DT	LORE	ANCHOR
aids	**.743** (95.3)	.653 (97.0)	.563 (93.7)	.352 (93.7)	.163	.254	.267 (97.4)	.252 (96.7)	**.595** (97.9)	.192 (98.8)	.170	.191
australian	**.801** (93.5)	.651 (89.9)	.488 (92.0)	.396 (92.0)	.041	.181	.297 (91.3)	**.383** (92.0)	.361 (97.1)	.274 (94.9)	.197	.141
bank	**.897** (96.1)	.839 (96.0)	.798 (94.9)	.671 (95.0)	.147	.298	**.540** (95.5)	.387 (95.5)	.425 (95.3)	.192 (96.1)	.055	.066
breast	.532 (97.4)	**.638** (96.5)	.568 (99.1)	.553 (98.2)	.280	.226	**.285** (98.2)	.266 (96.5)	.227 (96.5)	.123 (96.5)	.024	.122
churn	**.664** (95.8)	.655 (94.8)	.614 (87.0)	.476 (87.6)	.000	.299	**.350** (95.7)	.256 (95.2)	.210 (97.3)	.167 (98.2)	.138	.100
compas	**.875** (94.7)	.685 (95.1)	.527 (94.4)	.421 (94.4)	.142	.352	.432 (95.1)	**.547** (94.3)	.493 (95.1)	.205 (95.8)	.193	.152
ctg	**.758** (99.8)	.729 (99.8)	.732 (99.5)	.647 (99.3)	.117	.405	.293 (99.3)	.445 (99.3)	**.511** (99.3)	.062 (99.3)	.011	.275
diabetes	**.700** (92.9)	.684 (94.2)	.499 (89.6)	.498 (89.6)	.486	.335	.305 (89.6)	.304 (87.7)	.255 (89.6)	**.344** (88.3)	.198	.231
ecoli	.825 (100.)	**.871** (100.)	.834 (100.)	.838 (100.)	.447	.749	.524 (100.)	**.653** (100.)	.451 (100.)	.277 (100.)	.456	.633
fico	**.744** (90.3)	.678 (90.8)	.501 (90.3)	.426 (90.3)	.415	.411	**.318** (90.1)	.300 (89.5)	.274 (89.5)	.224 (90.5)	.131	.247
german	**.736** (89.0)	.653 (85.5)	.559 (85.5)	.523 (88.5)	.072	.242	.243 (86.0)	.252 (85.5)	**.354** (83.0)	.214 (81.0)	.193	.251
home	**.720** (96.0)	.612 (97.0)	.606 (94.9)	.590 (94.9)	.437	.311	.478 (96.0)	.341 (96.0)	.430 (98.0)	**.510** (94.9)	.434	.449
ionosphere	**.741** (94.4)	.740 (94.4)	.723 (88.7)	.519 (88.7)	.259	.586	.630 (98.6)	**.663** (90.1)	.581 (95.8)	.144 (95.8)	.054	.212
sonar	**.793** (85.7)	.735 (92.9)	.725 (83.3)	.715 (83.3)	.288	.388	**.581** (83.3)	.382 (85.7)	.379 (81.0)	.228 (83.3)	.129	.102
spam	.386 (96.0)	.178 (96.2)	**.416** (94.0)	.357 (94.4)	.274	.069	.201 (94.7)	.273 (95.1)	.152 (94.5)	.097 (93.7)	.147	**.289**
titanic	**.864** (90.5)	.770 (90.5)	.777 (89.9)	.689 (89.4)	.551	.681	.612 (93.9)	.616 (95.0)	**.658** (96.1)	.569 (94.4)	.470	.548
wine	**.714** (64.2)	.602 (67.2)	.577 (59.1)	.554 (58.6)	.243	.484	.269 (72.2)	.271 (70.1)	.225 (70.6)	.253 (70.2)	.270	**.359**
yeast	**.617** (76.4)	.474 (75.1)	.436 (69.4)	.440 (69.4)	.096	.299	.331 (78.5)	.355 (77.8)	.380 (75.4)	.376 (78.5)	.248	**.411**

TABLE A11: Global robustness metrics in all datasets with XGB as black-box. Prediction accuracy of surrogate classifiers is reported inside parentheses.

XGB	**Feature Importance Robustness**						**Decision Rule Robustness**					
	ILL-LR	ILL-LR-US	LIN-LR	INP-LR	LIME	SHAP	ILL-DT	ILL-DT-UC	LIN-DT	INP-DT	LORE	ANCHOR
aids	**.720** (95.3)	.492 (95.1)	.485 (91.1)	.400 (91.1)	.159	.250	.298 (95.6)	**.400** (92.3)	.249 (94.2)	.196 (95.1)	.194	.186
australian	**.781** (92.0)	.482 (92.8)	.554 (90.6)	.569 (90.6)	.022	.269	**.378** (92.0)	.377 (92.8)	.291 (92.8)	.193 (89.9)	.262	.262
bank	**.872** (96.2)	.810 (96.0)	.758 (95.3)	.668 (95.5)	.105	.293	**.530** (95.9)	.496 (95.5)	.436 (94.9)	.145 (95.1)	.012	.055
breast	.698 (99.1)	**.707** (99.1)	.506 (100.0)	.272 (97.4)	.262	.251	.305 (96.5)	**.356** (95.6)	.295 (97.4)	.183 (96.5)	.127	.108
churn	.585 (95.1)	**.648** (95.1)	.604 (87.6)	.473 (87.9)	.017	.339	.324 (96.1)	**.326** (94.5)	.315 (95.8)	.192 (99.0)	.132	.175
compas	**.844** (95.4)	.713 (95.2)	.589 (94.5)	.407 (94.5)	.127	.354	.327 (94.0)	.546 (94.2)	**.552** (94.5)	.179 (94.6)	.162	.137
ctg	**.811** (100.)	.776 (99.8)	.760 (99.8)	.552 (99.8)	.000	.545	.402 (99.8)	.390 (100.)	**.405** (99.5)	.081 (99.5)	.000	.271
diabetes	**.661** (92.2)	.620 (92.9)	.487 (90.3)	.486 (87.0)	.600	.380	.278 (91.6)	.271 (90.3)	**.307** (90.9)	.281 (92.2)	**.307**	.239
ecoli	**.892** (100.)	.851 (100.)	.811 (100.)	.816 (98.5)	.381	.662	.466 (100.)	**.723** (100.)	.551 (100.)	.288 (100.)	.527	.587
fico	**.718** (93.4)	.674 (93.2)	.519 (92.9)	.342 (92.9)	.432	.371	**.346** (93.8)	.326 (92.1)	.234 (92.8)	.197 (93.8)	.108	.236
german	**.714** (87.0)	.629 (87.0)	.640 (86.0)	.541 (86.0)	.054	.315	.270 (87.0)	.209 (77.5)	**.403** (81.5)	.195 (83.5)	.160	.261
home	**.729** (93.9)	.671 (96.0)	.618 (93.9)	.613 (93.9)	.611	.355	**.506** (96.0)	.408 (96.0)	.452 (97.0)	.473 (94.9)	.407	.457
ionosphere	.678 (91.5)	**.741** (91.5)	.733 (87.3)	.479 (90.1)	.235	.216	**.718** (95.8)	.647 (94.4)	.456 (97.2)	.090 (97.2)	.086	.120
sonar	.723 (81.0)	**.756** (81.0)	.671 (76.2)	.673 (78.6)	.062	.325	.391 (85.7)	**.496** (85.7)	.259 (88.1)	.283 (92.9)	.138	.027
spam	.577 (96.5)	**.579** (96.9)	.425 (94.5)	.279 (94.6)	.351	.069	.234 (95.9)	**.238** (97.1)	.173 (94.6)	.046 (94.7)	.132	.232
titanic	**.830** (91.1)	.704 (91.1)	.682 (88.8)	.710 (89.9)	.433	.588	.621 (95.5)	.642 (95.0)	**.661** (97.2)	.594 (96.1)	.550	.541
wine	**.667** (66.3)	.629 (66.6)	.616 (60.4)	.563 (59.3)	.338	.485	.286 (71.0)	.285 (71.0)	.253 (70.6)	.262 (68.2)	.273	**.347**
yeast	**.644** (86.5)	.637 (88.2)	.503 (81.5)	.472 (81.8)	.314	.412	.385 (89.2)	.425 (89.9)	.434 (88.2)	**.448** (89.9)	.329	.436

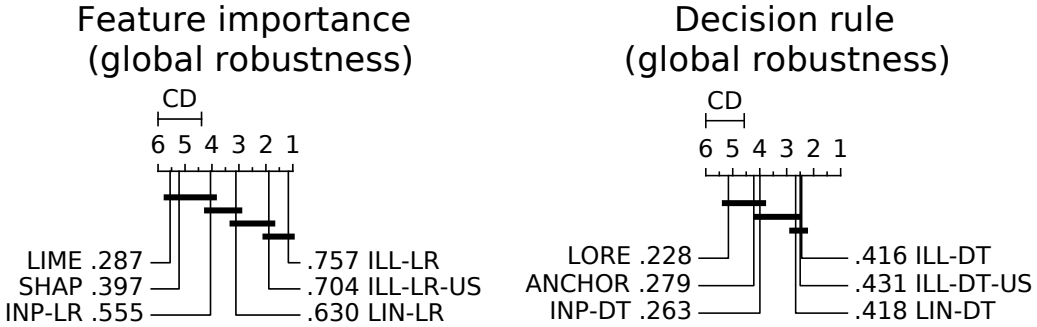


Fig. A8: Critical Difference plots with Nemenyi test at 90% confidence level for global robustness metrics in real-world datasets.