DOI: 10.1111/cgf.14650

COMPUTER GRAPHICS *forum*
Volume 42 (2023), number 1 pp. 7–20

Decision Boundary Visualization for Counterfactual Reasoning

Jan-Tobias Sohns, Christoph Garth and Heike Leitte

Technische Universität Kaiserslautern, Kaiserslautern, Germany
garth@cs.uni-kl.de,leitte@cs.uni-kl.de

Abstract

Machine learning algorithms are widely applied to create powerful prediction models. With increasingly complex models, humans' ability to understand the decision function (that maps from a high-dimensional input space) is quickly exceeded. To explain a model's decisions, black-box methods have been proposed that provide either non-linear maps of the global topology of the decision boundary, or samples that allow approximating it locally. The former loses information about distances in input space, while the latter only provides statements about given samples, but lacks a focus on the underlying model for precise 'What-If'-reasoning. In this paper, we integrate both approaches and propose an interactive exploration method using local linear maps of the decision space. We create the maps on high-dimensional hyperplanes—2D-slices of the high-dimensional parameter space—based on statistical and personal feature mutability and guided by feature importance. We complement the proposed workflow with established model inspection techniques to provide orientation and guidance. We demonstrate our approach on real-world datasets and illustrate that it allows identification of instance-based decision boundary structures and can answer multi-dimensional 'What-If'-questions, thereby identifying counterfactual scenarios visually.

Keywords: visual model evaluation, machine learning explanation, inverse multi-dimensional projection

CCS Concepts: Human-centered computing - Visual analytics; Computing methodologies - Model verification and validation

1. Introduction

Decision making in all aspects of life has become increasingly data-driven and relies on machine learning algorithms to a growing extent [JM15]. While a misclassified email is merely a nuisance, the consequences of misclassification in medical therapy planning are more drastic and require a transparent and trustworthy process. Ensuring transparency and trust for complex machine learning models is thus an ever-growing challenge; this is in particular true for black-box models, which are not interpretable on their own.

In this context, providing supplementary human-understandable explanations for model predictions increases confidence in properly performing systems [BSO15] and allows the appropriate suspicion in flawed ones [SSTea20]. Hence, an emerging direction of research aims at generating explanations for model behaviour in a wide variety of forms [LK19, RSG18, ERH*19]. Without limiting the explanation to a certain architecture, the explanation has to be formed by probing the black-box model's decision function. The interesting section of the decision function is where the classification changes, which is called the *decision boundary*. The decision boundary can be described by samples [WMR17, MM21, RFT18] or visual maps [MHT18, SGH15, BPP*15].

For a given input to a prediction model, similar inputs for whom a different outcome is predicted by the model are called counterfactual examples [WMR17]. In particular, local samples of the decision boundary with regard to a given data instance are counterfactuals. While counterfactuals do not explicitly shed light onto model-internal factors leading to the prediction, they provide insight into what would need to change to generate a different outcome. As humans inherently deduce their internal explanations from comparisons [Lip90], counterfactual- and therefore decision boundary-reasoning is a preferential explanation approach [WMR17, BSR20].

Global samplings of decision boundaries extract scatterplot projections [LJLH19, WFC*18, JZF*09] or sub-spaces [MM21, RFT18, YRWG13] from the high-dimensional input space that visually separate a given dataset colour-coded according to class affiliation. The decision boundary then resides in the interval between instances having different class labels.

By regularly sampling the decision function, visual maps provide an explicit insight into decision zones instead of just samples. Therefore, the decision boundary can be read exactly, even if no samples are nearby. Currently, visual maps cover either univariate changes to an instance [KPN16] or the whole dataset under multi-variate changes mapped to 2D [MHT18, SHH20, RD00]. The multi-variate maps employ non-linear dimensionality reduction, since linear projections fail to capture non-linear manifolds [XYC*18] even though interaction with *local* linear embeddings has been shown to succeed [IMI*10]. Consequently, the non-linearity strongly distorts the input space, preserving the topology of the decision boundaries but distorting their shape and their distance to explained samples [REHT19], which are excellent explanations.

In this paper, we propose a framework for the visual exploration of high-dimensional decision functions that enables the visual identification of feasible instance-based explanations through counterfactuals. We create dense local linear maps around an instance for answering 'What-If' questions about the shape and distance of nearby decision boundaries. We complement the maps with the mentioned non-linear and one-dimensional decision boundary techniques to create a comprehensive interactive framework aimed at classifiers with a limited number of inputs.

2. Related Work

Research on explaining decision functions in machine learning models has produced a broad range of solution approaches. We summarize them by their form of explanation: Explanation through counterfactuals, visual model evaluation, boundaries in labelled datasets, and decision maps. We discuss existing techniques comparable to our work from all fields in the mentioned order.

Counterfactuals. Currently, most of the machine learning literature is united under the concept that computing a counterfactual is an algorithmic optimization problem [WMR17, DPB*19, LK19, CRSPG19]. However, identification of optimal counterfactuals is NP-hard [TGR20] and the definition of optimality varies on a case-by-case basis [SF20]. Presenting a diverse set of counterfactuals instead [DMBB20, MST20] increases the chances that an applicable example is found. Still, explanations through algorithmic counterfactuals are missing the flexibility, interactivity [SF20] and context [GHYB20] a visualization can provide.

Visual model evaluation. In the recent years, the analysis of machine learning models has shifted from raw statistical measures to interactive tools that present the model's decision behaviour. Ming *et al.* [MQB18] approximate the decision space of black-box models with global linear rules that can be visually aligned with human understanding. The approach was extended to provide both local and global explanation with visual rules [NP21], which limited the application to random forest models. Cheng *et al.* [CMQ20] proposed a similar iteration of scoped rules [RSG18, Mol19, DCL*18] on interactively refined sub-groups that are evaluated over univariate counterfactuals. They also provide an interface to communicate and influence diversity in instance-specific counterfactuals.

While a common approach is to abstract from the complex model to an easier surrogate model [RSG16, MQB18], the decision space can also be probed explicitly starting from an instance. The What-If-Tool [WPB*19] and Prospector [KPN16] let a user probe the model response under manual perturbations to an instance. The former's focus is on model evaluation through a test set and therefore requires trial-and-error probing in text fields while the latter aids probing by showing model predictions under univariate changes in a colourmap. We extend this analysis to multi-variate changes.

Boundaries in labelled datasets. Prediction models are typically evaluated on a discrete test set of data instances which are then labelled. Hence, explaining boundaries between these labelled instances is a parallel problem to the one we are addressing here. As datasets usually comprise of many dimensions, this often reduces to the interactive scatterplot exploration through dimension reduction for which both linear [JZF*09] and non-linear [YRWG13, NM12] tools have been proposed. Ranking the possible perspectives allows filtering for interesting ones [TMF*12]. Returning to the issue of boundaries, Ma *et al.* [MM21] analyse a labelled dataset to compute a set of local linear boundaries that approximately separates the sample classes. While they generate sparse abstractions of the boundaries between two classes through sub-sectioning, we focus on instance-based dense exploration to support explanations through contextual counterfactuals.

Projections of labelled datasets have also been applied to evaluate prediction models in linear [WPB*19] and non-linear embeddings. The relevant non-linear embeddings are integrated into application-specific frameworks. Their aims vary from improving class separability and thereby model performance through feature selection [RFT18], over latent space interpolation between two high-dimensional samples [LJLH19], to inspecting model behaviour on new samples during transfer learning [MFH*21]. Mazumdar *et al.* [MPNP21] extend on the concept by basing their dimensionality reduction directly on the internal decision paths of instances in random forests.

While approaches based on labelled datasets often times provide sufficient and interpretable explanations, they evaluate a model solely on a discrete set of instances. As a result, the decision function can only be approximated from a sparse sampling of the input space, even when the local projections are chosen to show a clear separation between instances [MM21, MFH*21]. However, the actual decision function may have arbitrary shapes between these instances which is not derivable from the instances alone (ref. Figure 3c). Therefore, our approach moves the emphasis from a sparse sampling to a dense evaluation of the decision function in input space.

Dense decision maps. Sampling the input space on the basis of a two-dimensional embedding creates a dense explanation of the decision function. Espadoto *et al.* [ERT19] perform an extensive comparison for suitable projection techniques, which they later apply to visualize agreement between classifiers [EAS*21]. They come to the conclusion that non-linear dimension reductions are suited best for this application, which is approved by several other articles [SGH15, RD00, SHH20]. In case that the classifier explicitly defines a reduction function, e.g. a support vector machine, this mapping should be used [BPP*15]. However, Rodrigues *et al.* [REHT19] point out that in general, there are three problems with non-linear embeddings. First, non-linear dimension reductions typically do not feature an inverse projection. Learning an approximate inverse projection can take significant time [ERH*19, AVBD*12, AVMC*15], except if it is integrated in the reduction process already [OEHT22]. Second, they tend to overfit in confusion zones leading to uninterpretable noise. Third, the distances to the visible decision boundary in the map and the real decision boundary in feature space do not match.

We use linear projections to create dense maps that inherently do not suffer from these problems. While linear projections have been considered for this application before [CCWH08], they were dismissed due to their poorer performance in cluster separation [SGH15, ERT19] and possible data point overlap as compared to non-linear methods [EAS*21]. We show that by providing complementary interactive selection and interpretation tools, this weakness can be alleviated.

3. Method

The central idea of a counterfactual explanation is to describe the local structure of the decision boundary of a classifier that separates the predicted class from a different one. In order to fully understand and explore this concept, we will first introduce decision boundaries formally and motivate our approach (Section 3.1). We continue with a discussion of desired properties of map explanations (Section 3.2) and conclude with the construction of an embedding to explore the decision boundary around an instance (Section 3.3).

3.1. Decision boundary

Consider the point cloud in Figure 1a which depicts a synthetic dataset of three anisotropic clusters in 2D. The colours indicate the class membership. The task of a probabilistic classifier model is to compute for any given point in 2D space a probability of class affiliation. We consider a sample point x to be predicted class A by classifier f if f predicts a probability for A higher than a defined target threshold t. Assuming a continuous input space, the decision boundary is then formed by the set of sample points that lie exactly on the threshold t:

$$B(A) = \{x \mid f_A(x) = t\} \tag{1}$$

Without loss of generality, $t = 0.5$ lends itself as a suitable threshold for binary classification [BPP*15] and is therefore used throughout this paper. This threshold can be chosen arbitrarily to match the respective application scenario. In multi-class classification, we consider $t = 0.5$ equally applicable since in our experiments boundaries locally collapsed to only two neighbouring classes. For regression analysis, the definition follows analogously with the threshold t as a chosen target value. In the visualization of the decision space in Figure 1b, the hue indicates the highest predicted class with saturation dropping to white at $f_A(x) = t = 0.5$. The decision boundary is the white band in-between classes.

As counterfactual explanations either flip the class with as little change as possible or reach a certain target threshold, all counterfactual points will lie on the decision boundary. Thus, they are a sampling of a decision boundary. If one finds the decision boundary of a class A as the points where the threshold is reached, they have found all possible counterfactuals. The explanation then follows from the set of counterfactuals or the choice of a point within.

While this concept is intuitive in our synthetic example, the set of points can be difficult to imagine in higher dimensions. In 3D, it forms surfaces that can still be easily rendered using scalar field visualization techniques such as isosurfaces [LC87]; for datasets with more than three attributes, direct visualization is no longer feasible. As a prediction model forms a continuous function that outputs a value for any point in the input space, the input space can be considered a high-dimensional scalar field. A common approach that is well established for the interpretable rendering of high-dimensional scalar fields is the use of cutting planes [vWvL93, HJ11]. The general idea is to sample the function on a low-dimensional manifold, commonly a straight line or a plane.

This concept has already been applied for the explanation of classifiers, namely in the form of partial dependence plots (PDPs). While the PDPs introduced by Friedman [Fri01] average the

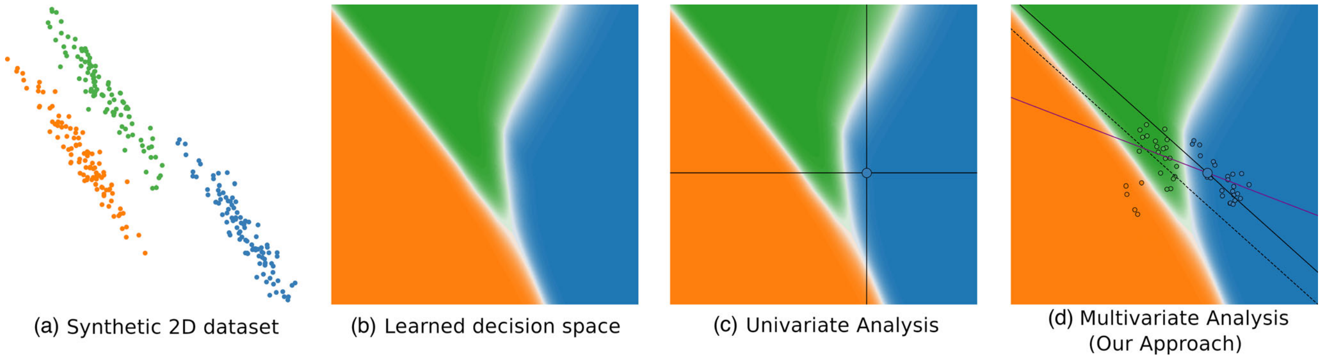


Figure 1: *(a) Synthetic 2D data. (b) Input space coloured by model class probability. Hard to view for >2D. (c) Black lines are PDP sampling for a data point. (d) Our approach. The dashed line is the regular PCA-embedding. The black line is the global PCA-embedding shifted into an explained data point. The purple line is the local, nearest neighbour-based embedding. The small circles are the nearest neighbours.*

sensitivity over a full dataset, the focus of this paper is on explaining instance-specific behaviour and therefore, we follow the notation of Krause *et al.* [KPN16] to inspect the partial dependence of a single instance. A PDP samples the decision function in one dimension, keeping all other attributes constant. In Figure 1c, this corresponds to a line starting in one of the data points and being parallel to the sampled axis.

The approach can be directly extended to 2D partial dependence by changing two attributes. This constructs an axis-parallel plane. As our synthetic dataset is only two-dimensional, this plane perfectly describes our scalar field as seen in Figure 1. In practice however, machine learning is applied to high-dimensional input spaces. Now, the *n*D dependency is hard to visualize and 2D planes can be drawn for arbitrary combinations of axes, resulting in $n \cdot (n-1)/2$ possible combinations. For more than a single-digit number of features, the amount of plots would be overwhelming.

To solve the visual overload problem, data scientists typically employ dimensionality reduction algorithms. In these algorithms, *n*D data points are projected into 2D by optimizing a stress criterion. Even though these algorithms are designed to approximate the distribution of data points and the focus of this paper is exploring a continuous decision function, the 2D representation can be used to explore the decision space [AVBD*12, ERH*19]. Geometrically, the projection embeds a hyperplane in input space for the linear case and a hyper-surface for the non-linear case. This embedding can be evaluated by inversely projecting 2D points to input space creating a densely sampled representation [EAS*21] with small remaining uncertainty about *n*D values in-between samples.

3.2. Desiderata

After a comprehensive review of existing work on decision boundary maps and counterfactual explanations in Section 2, we distilled the following requirements for an interpretable explanation with a decision boundary map:

- *R1: Convey distances in decision space*. The aim of a decision map is to convey the range of scenarios that keep or flip a decision. Therefore, the visual distances of test samples to the decision boundary on the map need to be comparable, i.e. there should be a monotonic relationship between visual and actual distance to the boundary. As this is hard to achieve for high-dimensional datasets [REHT19], focusing on specific instances is sufficient for counterfactual reasoning. The distance measure should reflect the expected distance between inputs, which in our case is Euclidean distance.
- *R2: Favour likely alterations*. Reciting the goal of counterfactuals to provide an explanation via an expressive comparison, not all counterfactuals are equally helpful as explanations. A decision boundary that is reached via likely changes is more realistic and therefore more helpful than one with unlikely changes [DPB*19, PSSR*20, CRSPG19]. The distances in the embedding should reflect the likelihood of a change in reality.
- *R3: Show a close decision boundary*. Reducing dimensionality is always a trade-off between many possible optimization criteria. The embedding can only cover a small sub-set of the high-dimensional space. Therefore, the optimization of the embedding should focus on providing explanatory value in the sense that the shown decision boundary actually is close to the explained instance [WMR17, LLM*18, DMBB20]. Note that suitable counterfactual reasoning just needs an actionable close boundary, not necessarily the mathematically closest one [MST20].

3.3. Embedding construction

On the basis of the requirements *R1–R3*, we now construct a suitable embedding. The construction process is illustrated on the synthetic example in Figure 1d. Since plotting an *n*-dimensional input space is infeasible, we explain the reduction from *n*D to 2D by reducing from 2D to 1D. A 1D line in Figure 1 is, therefore, analogous to a 2D hyperplane residing in *n*D.

A major decision for creating an embedding with dimensionality reduction is whether to stay linear or allow non-linear distance transformations. As described in Section 2, in non-linear decision boundary maps distances between embedding points and decision boundaries in input space are not matching [REHT19], regardless of the projection technique [ERT19]. Therefore, non-linear embeddings are violating *R1*. On the contrary, the axis of linear projection techniques are based on linear feature combinations and therefore fulfil *R1*.

From the plethora of linear transformations, we choose PCA [Pea01] to build our embedding visualization also utilized by OptMap [ERHT21] in a related scenario. In contrast to OptMap, we focus on explaining classifier outputs instead of optimization paths in the domain space of real-valued functions. Further, in OptMap, the PCA is trained from samples on a regular grid, while *R2* leads us to base it on the distribution of real data instances. This training data is not restricted to the training or test dataset of the model and can be any realistic distribution of samples as long as it is not strongly biased.

We choose PCA over more discriminant methods like LDA [Fis36] for two reasons: (1) PCA-axes are optimized to capture the most variance in the data. Assuming a higher variance in a feature means that it is more likely to change, the embedding implicitly accounts for feature variability (*R2*). Since this assumption only perfectly holds for a statistical dataset as in Figure 1, we provide additional tools to control feature mutability in Section 4.5. (2) Since we standardize all features before PCA, the analysis is an eigenanalysis of the correlation matrix. Thus, the covariance of two axes signals the correlation between the features. As a result, points in our map adhere to the linear feature correlation in the dataset (*R2*). While we cannot expect a dataset to only contain linear correlation, the assumption usually holds when restricting to local neighbourhoods. Hence, we restrict PCA training samples to a set of nearest neighbours in *n*D.

Constructing a PCA-hyperplane is not guaranteed to find a decision boundary at all, especially not a close one. To satisfy *R3*, we force the hyperplane orientation towards a close decision boundary by restricting the training samples to a relevant local sub-set. Therefore, we find a balanced set of nearest neighbours to the explained point, where half the samples share the predicted class and the other half is predicted to be in another class. The neighbours are determined in one ball tree for each class on Euclidean distance of standardized feature values. The default of 100 diverse neighbours, 10–15% of the data, worked best in our experiments, but is adjustable in the interface header.

The hyperplane constructed by the principal components pc_1 and pc_2 is traditionally anchored in the mean of the training data at (0,0). Because the aim is to explain the decision space around an explained instance i, we shift the PC-hyperplane into i. Figure 1d (purple line) provides a sketch of our proposed embedding. The mapping from a point (x, y) in embedding space to feature space is computed with:

$$Inv(x, y) = [x, y][\vec{pc}_1, \vec{pc}_2] + i \tag{2}$$

As this mapping is computationally simple and can be vectorized, the embedding is sampled once per pixel (x, y) (Figure 1: along the purple line) and the corresponding points in input space $Inv(x, y)$ are classified in the model. The predictions form a multi-variate PDP coloured by the most probable class. The neighbours used for training are projected into the plane as coloured circles to identify practically occurring feature combinations, though these could be omitted to reduce complexity for casual users.

4. Design

In this section, we describe a framework for exploration of local decision boundaries through the previously derived sampling techniques. As our tool allows instance-based explanations through counterfactuals, we name it CoFFi (COunterFactualFInder). First, we describe requirements for such a visualization and introduce the interface. Afterwards, we describe the design decisions of each component in the order of a typical workflow.

4.1. Design overview

The review of related work in Section 2 revealed design requirements for a holistic analysis of decision boundaries, which can be ordered from overview to detail.

- *DR1*: Show the overall class separability and data distribution [ERT19, SGH15] as well as the focused instance's placement therein [SHH20] to give an overview.
- *DR2*: Allow univariate sensitivity analysis to support sparse and simple explanations [WMR17, KS20, KPN16].
- *DR3*: Provide interpretable, direct analysis of the decision function under multi-dimensional changes (Section 3.2) to support multi-variate explanations.
- *DR4*: Retain flexibility to account for uneven feature model importance [RFT18] or mutability [SF20].

To make the framework usable in practice, we adhere to three additional design goals.

- *DR5*: Independence of the underlying machine learning model. With the ever-changing search for superior model architecture, fitting a visualization technique on a specific model type severely limits its usability. As a black-box approach, we only require a *predict* function and sample data, thus are model-agnostic.
- *DR6*: Applicability to common machine learning data. The exploration of decision boundaries for texts and images is currently still restricted to exemplary counterfactual generation [CYX*20, DCL*18, GWE*19], so we focus on tabular data with a number of dimensions displayable as a list, i.e. less than 30 [CMQ20, WPB*19, GHYB20, KPN16].
- *DR7* :Applicability to a variety of model outputs. In this paper, we demonstrate it on binary- and multi-classification problems. Although we have not applied our tool to regression models yet, they fit into our visualization by setting the boundary threshold t to the target regression value.

Our interface, shown in Figure 2, is split into four major components: The *topology view* (a), the *partial dependence view* (b), the *embedding view* (c) and the *feature selection* (d). The components each fulfil one of the listed requirements. The topology view grants insight in the separability of the dataset (*DR1*). The partial dependence view shows the univariate analysis of the local decision behaviour (*DR2*). The embedding view extends the exploration to multi-variate space (*DR3*). The feature selection provides a guided way of reducing the search space (*DR4*). Additionally, data points can be selected and compared in a data table (e). All components are linked and update to the current selection. A prototype of the presented framework is implemented using Python, Bokeh and Panel [Rud] and is available on GitHub at https://github.com/Jan-To/COFFI.

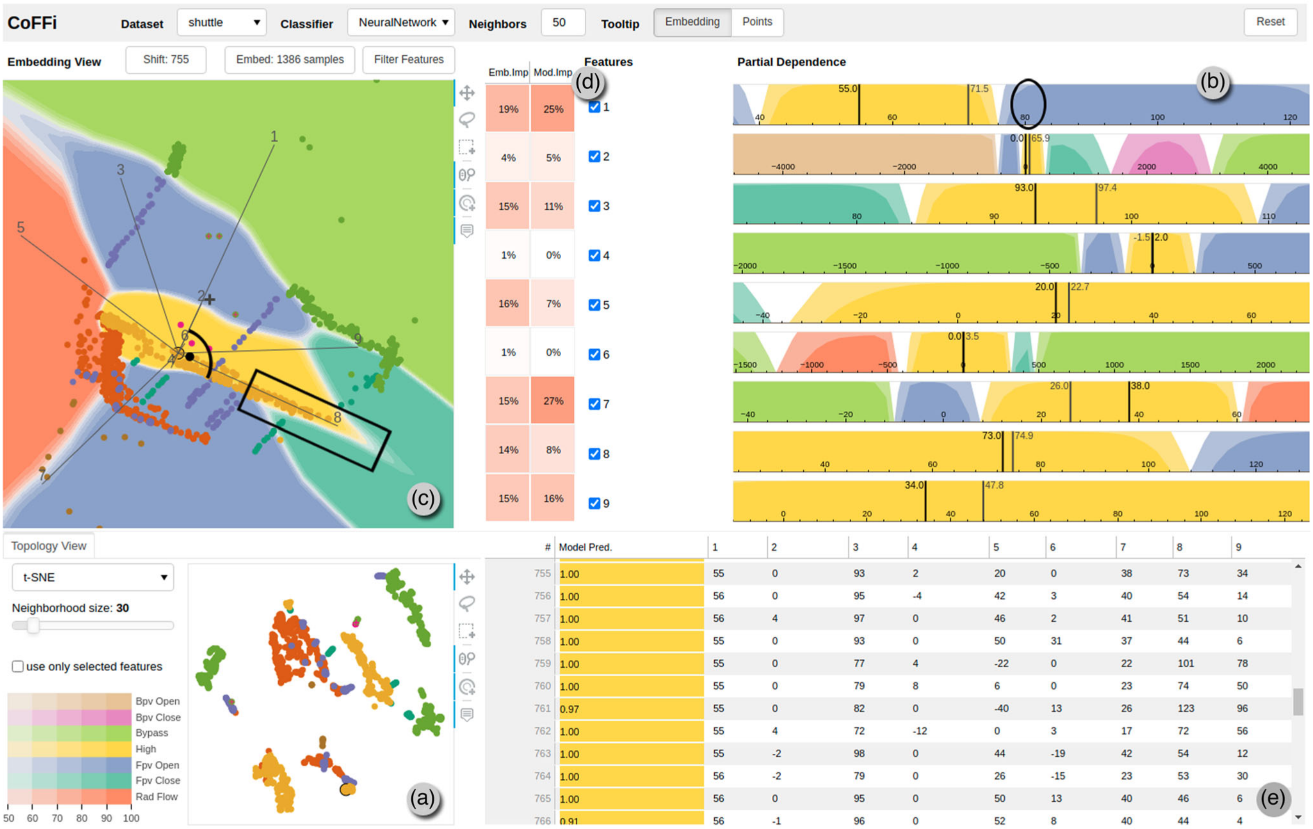
The interface is explained on the example of a space shuttle dataset from NASA [DG17] shown in Figure 2, where nine radiator sensor measurements are given and a fully connected neural network with three 100-neuron-layers is used to predict the corresponding radiator position with 99% accuracy. The dataset contains 58,000 instances over seven classes. To reduce overplotting, we limit the visible scatter glyphs to maximal 400 instances per class, but full data can be used for computation while remaining interactive.

4.2. Topology view

The exploration of decision boundaries in high-dimensional space is only sensible if we know what we are looking for. For a clustered, clearly separable data manifold as in Figure 2, we can expect the model to form similarly simple decision boundaries. For a complex, hardly separable data manifold as in Figure 3, the decision boundary could be equally complex. Hence, the first component to look at after loading in a dataset and a trained prediction model is the topology view.

The topology view is designed to give an overview of data clusters and class separability. Each instance of the provided sample data is rendered as a point in a scatterplot created with non-linear dimensionality reduction. The points are coloured by the predicted class with misclassified samples having a cross added in the colour of the ground truth class, if available.

Two state-of-the-art non-linear dimensionality reduction techniques—t-SNE and UMAP—are available to plot the data points based on their feature values. The positioning of the data points approximates the intrinsic manifold of the dataset, while the colour distribution indicates the class separability. A dataset with feature-wise distinct classes shows in a clear separation into uni-coloured clusters as in Figure 2 (orange, yellow and green), while the other classes seem harder to differentiate. In our experiments,



CoFFi Dataset shuttle ▾ Classifier NeuralNetwork ▾ Neighbors 50 Tooltip Embedding Points Reset

Embedding View Shift: 755 Embed: 1386 samples Filter Features

Emb.Imp	Mod.Imp	Features
19%	25%	☑ 1
4%	5%	☑ 2
15%	11%	☑ 3
1%	0%	☑ 4
16%	7%	☑ 5
1%	0%	☑ 6
15%	27%	☑ 7
14%	8%	☑ 8
15%	16%	☑ 9

Topology View

t-SNE ▾

Neighborhood size: **30**

□ use only selected features

#	Model Pred.	1	2	3	4	5	6	7	8	9
755	1.00	55	0	93	2	20	0	38	73	34
756	1.00	56	0	95	-4	42	3	40	54	14
757	1.00	56	4	97	0	46	2	41	51	10
758	1.00	55	0	93	0	50	31	37	44	6
759	1.00	55	0	77	4	-22	0	22	101	78
760	1.00	55	0	79	8	6	0	23	74	50
761	0.97	55	0	82	0	-40	13	26	123	96
762	1.00	55	4	72	-12	0	3	17	72	56
763	1.00	55	-2	98	0	44	-19	42	54	12
764	1.00	56	-2	79	0	26	-15	23	53	30
765	1.00	56	0	95	0	50	13	40	46	6
766	0.91	56	-1	96	0	52	8	40	44	4

Figure 2: *Overview of our interface on neural network prediction the radiator position of a space shuttle from nine sensor measurements. The decision boundaries of sample 755 are explained. The topology view (a) shows clear cluster distribution. The partial dependence view (b) shows possible predictions under single parameter change. The embedding view (c) shows possible prediction under statistically likely multi-linear changes. The attributed feature importance (d) in the model mostly lines up with the variance in the embedding. Thus, the embedding can be assumed to show a representative explanation of decision boundaries. Specific points can be searched and selected in a data table (e). The black annotations in (c) illustrate the conclusions drawn in Section 4.4.*

t-SNE proved better for finding clusters, while UMAP excelled at capturing the intrinsic manifold. Nevertheless, the outcome of both algorithms strongly depends on their hyperparameters, so we included sliders to experiment until a satisfying setting is found.

The downside of these non-linear embeddings is that most of the information about global distances between points is lost. Conclusions can only be drawn about local similarity. Due to the focus on immediate locality and the resulting distortion in the non-linear embedding, we frequently experienced points that are distant in the plot but within each other's nearest neighbours. In preliminary versions, we augmented the scatterplot with inverse mappings of non-linear dimensionality reduction as in [SHH20]. However, in our experiments, the background embedding did not provide additional information about class separability compared to just the scatterplot. For that reason, we chose to not add a background colouring here.

The aim of this workflow is to explain the decision boundary for a specific data point. If this point is novel, it can be added in the data table. If we are searching for a point of interest, a single different-coloured dot or a colourful glyph-border in the topology view provides hints about atypical instances. In the shuttle example, an instance of the central class High is chosen, demonstrating the possible variety of adjacent classes. A click selects the instance and updates the other components to this point.

4.3. Partial dependence view

The most easily understood way to describe decision boundaries is by illustrating univariate behaviour. In the context of model predictions, this approach is typically called partial dependence analysis [Fri01]. For each dimension, the class prediction is observed with one feature altering while the other dimensions are fixed. By design, partial dependence perfectly captures the univariate behaviour around the examined instance. As partial dependence is a well-established technique that still marks the state-of-the-art for inspection of feature-model relationships [HHC*19, KPN16, WPB*19], we include them with a new look. Partial dependence is typically plotted as line charts [WPB*19, HHC*19, CLG*15, Fri01] or colourmaps [KPN16]. We choose to use a hybrid of colourmaps and line/area charts called horizon chart [Few08], which combines the advantages of both previous visualization methods.

Horizon charts are vertically condensed and colour coded area charts. The area is segmented into ideally two horizontal bars [HKA09] which are colour-coded and shifted over each other. The colour component facilitates highlighting regions of interesting value range, while the slope conveys an accustomed way of reading gradients as well as the possibility to read exact values. The lowered space constraints of horizon charts over line charts allows us to show significantly more dimensions at once than previous methods [WPB*19, HHC*19, CLG*15].

Here, horizon graphs are used to steer attention to features the model is sensitive about. In that way, they capture the feature specific volatility in an instance's prediction. At the same time, the range of possible predictions for individual feature changes is presented. Thereby, insight over a plethora of hypotheticals is generated without actually having to try them out by hand.

The message conveyed by horizon graphs depends on the axis ranges. The x-axis should cover all possibly occurring feature values, hence we set it to cover the value range of the training set. The vertical baseline is the decision boundary threshold $t = 0.5$, since our goal is to put emphasis on the pivot point between predicted classes. The vertical axis is set to 25% prediction change, so that over two positive bands, the horizon covers the prediction range between 50% and 100% per prediction per class. The classes each are colour-coded with more confident classifications coloured with richer colours. We can, therefore, read the predicted class under all individual feature changes without any manual work.

The feature values of the currently selected data point are indicated with black lines, while the currently explored counterfactual is shown with grey lines, which we will learn about in the next section. For point 755 selected in Figure 2 none of the sensors measure an extreme value. Therefore, the neural network predicts 100% class High, which can be read from the full height of the saturated yellow area at the black lines. The chart provides hypothesis testing such as if sensor 1 had been 80 and all other sensor were the same, the model would have predicted 99% Fpv Open.

Categorical features are visualized as discretized versions of the horizon charts, which look like stacked bar charts. Ordinal categories are converted to numerical features. Due to the PCA analysis of Section 3.3, nominal data are supported through one-hot encoding up to 30 total dataset dimensions.

4.4. Embedding view

The previous evaluation relies on the assumption that features are changing independently from another. In practice as well as in the case of a space shuttle radiator, this assumption may be wrong. For measurements of one sensor to change, the radiator moves its position, which inevitably changes the measurements of other sensors. To convey the decision boundary under reasonable changes, an explanation needs take these dependencies into account. Therefore, we extended our analysis to multi-variate changes with respect to the underlying covariance in the Embedding View.

The Embedding View shows the prediction probabilities on a linear hyperplane based on the inspected point. The necessary cutting plane is generated as introduced in Section 3.3. As a continuous bijective mapping between the plane and the parameter space exists, a sample point in parameter space can be created at every position. A dense regular grid is sampled on the cutting plane to create a smooth visual impression. For each sample, the class probabilities are predicted in the machine learning model. From these probabilities, we create a map of the model predictions on the hyperplane where each pixel is coloured by the most probable class. The saturation is increased with certainty of the prediction. Hence, decision boundaries will show as white areas or flips in hue.

A grey biplot [Gab71] of the high-dimensional axes is added to indicate how feature values change while moving in the sampled plane. We centre the biplot in the focused data instance and orient the axes in positive feature direction. Since the hyperplane is created on centred and normalized training data, the covariance of two axes in the biplot signals the correlation between them. We can assume that axes in similar directions are positively correlated, axes in opposite directions are negatively correlated and the other axes are not correlated. Further, the axes scaling attends to the variance of each feature in the training data, which is encoded in the relative length of the axes.

The coloured map lets us draw conclusions about multi-variate decision boundaries. In our linear embedding feature, values increase linearly in the direction of their respective axis. Moreover, the respective feature value does not change when moving orthogonal to an axis. Hence, a decision boundary that is linear and orthogonal to an axis, relies on this feature to cross a certain threshold. The extension of said boundary signals the generalizability with regard to the other features, which change when moving along the boundary. Decision boundaries that form curves rely on a non-linear combination of features. We enforce an 1:1 ratio of the plot's own x- and y-axis so that when a variable strongly influences a decision boundary, it is visible as its axis being (close to) orthogonal to that boundary. The lower part of the decision boundary between High (yellow) and Fpv Close (teal) in Figure 2c is orthogonal to the feature axes 1 and 2, so one of them or both are the deciding factor here. The linear boundary ends in a sharp curve when one of the similar-oriented features 5, 8 or 9 reach a certain value. The partial dependence view hints towards a further exploration of the interaction with sensor 5 as it shows a teal range on the lower end. The necessary workflow is described in Section 4.5.

To annotate the embedding, the training instances are projected into that plane and shown as circles in the plot. The same colouring schema is used as in the topology view which consists of slightly more saturated class colours to make circles distinguishable from the embedding colours. The sample points work as an indicator for the spatial distribution of real data as well as for the descriptiveness of the embedding. Regions where samples share their colour with the embedding are regions where the cutting plane is illustrating the behaviour of the model on real data well. Regions where the colours do not match signal that the cutting plane does not generalize to these points. In the example embedding of Figure 2, the Fpv Open, High and Bypass match with the linear embedding, while the projection is inferior for the other classes.

A point of interest is focused by clicking on it in any view, increasing its size. The embedding and Partial Dependence View can then be updated to the selected point with the 'Shift'-Button. Since the samples are orthogonally projected onto the linear embedding, shifts necessarily are orthogonal to the hyperplane and the embedding slices the parameter space in parallel. Thus, with the same training data, only the embedding colours change, while the axes orientation and the location of the sample points stay the same.

On startup and inital selection of a datapoint, the embedding is based on a (shifted) PCA of all available instances (cf. Figure 1d black lines, Figure 2c, Figure 3a) for overview and consistency. However, based on the approach of Section 3.3, the training data of the embedding should be reduced to the nearest neighbours to better adhere to the local manifold around an instance. This is achieved with the *Embed*-Button, which then reflects the current number of training samples and also works on a custom selection drawn in the embedding or topology plot. All non-selected samples are rendered transparent, as their projection to the embedding is less meaningful. In our experiments, the closest 100 nearest neighbours provided a good balance between locality and generalizability, thus, we use this number throughout our examples, one of which can be seen in Figure 3. A further benefit of having a linear embedding is its rapid computation, which mainly depends on the number of dimensions for neighbourhood search and PCA. In our experiments, the projection and inverse mapping required under 10 ms, while the model evaluation of 300×300 pixels took under 0.3-s scaling linearly with pixel count and ML model evaluation time. Hence, it is possible to zoom and pan interactively in the embedding, exploring regions of interest in more detail.

Lastly, the embedding can be probed at any position by hovering, generating an explanation for the prediction. Clicking in the embedding creates a grey cross that marks the probed position. Simultaneously, the partial dependence view shows the feature values of the inversely projected point as grey lines. This serves two purposes. First, precise readings of feature values at the decision boundaries and beyond can be taken. Second, the comparison with the black markers of the focused point reveals the necessary changes that can serve as a counterfactual explanation. In Figure 2, we selected a multi-variate counterfactual beyond the Fpv Open boundary that was not evident from individual feature changes in the partial dependence view. By freely choosing points along the decision boundary, the user can explore a plethora of possible counterfactual explanations visually. Thus, the explanation can be steered to personal preference without knowledge about coding, machine learning models or dimensionality reduction other than PCA. The steering of the exploration is extended in the next section.

4.5. Feature selection

When analysing the embedding of a high-dimensional reduction, keeping track of all the feature interactions quickly becomes overwhelming. A user may not even be interested in explanations that require many features to change, especially features he can not influence. At the same time, it is obvious that a model can be more sensitive to some features than to others. Consequently, the embedding should be adjusted to better capture the model's 'view' on the parameters and the user's personal flexibility.

In the feature selection component (Figure 2d), features can be disregarded for the visual sensitivity analysis by fixing them to the value of the focused point. The embedding view is then read as: 'What are the predictions of likely changes to the focused instance under the assumption that unchecked features do not change?' In case, the user has no preferences on which features are immutable or irrelevant, we provide guidance on how to filter features for an expressive embedding.

As the embedding should capture the model sensitivity, we evaluate its quality by the accordance of the feature influence in the *embedding* compared to in the *model* shown in an annotated heatmap in Figure 2d. Embedding influence is computed over the length of the feature vectors spanned by PCA. As model-agnostic feature importance is still a topic of active research, model sensitivity is computed by the permutation importance measure [Bre01, FRD19] based on implementation availability. We restrict model importance measure to the training data of the current embedding, thereby keeping the measures comparable and adjusting to local differences. In the end, both measures are normalized for easier comparison. The embedding quality can be assessed by comparing the influence of features in embedding and model, and improved by adjusting the embedding to better mirror the model.

5. Case Study

In this section, we demonstrate how the proposed design can be used to explore decision boundaries and draw conclusions through a case study on a real-world high-dimensional dataset.

The diabetes diagnosis of members of the Indian Pima tribe is to be automated. The dataset is pre-selected to contain only females above 21 year old [RA15]. The dataset contains 768 samples with eight numerical input features. The output variable signals whether the patient was diagnosed with diabetes mellitus (orange) or not (green). We train a random forest classifier with 100 internal trees and achieve 80% cross-validation accuracy.

We consider a specific female 667 (f667) who is missing the insulin measure, but has average values for all other features, indicating no clear class affiliation. The model predicts f667 to be healthy but she actually is diabetic. For the model to be used in practice, both the female herself and the model developer have interest in why this error happened. We provide an explanation by exploring the local decision boundary around the sample point.

After seeing no clear class separation in the topology view in Figure 3e, we select f667 by clicking it and shifting to (a+d). The partial dependence view (d) updates to show the model's response under feature changes in one dimension. The green areas left of the black instance-lines in (d) indicate that a lower pregnancy, glucose, BMI or age value lowers the prediction for diabetes. Apart from this expected behaviour, we mark multiple unexpected patterns in Figure 3d that raise suspicion about model robustness. First, the chances to be healthy would have been significantly improved had the female been older than 55 years. Second, a higher glucose value from 111 mg/dl up to 140 mg/dl would improve her prediction even though the healthy range is anything under 140 mg/dl according to the American Diabetes Association. Lastly, the model would predict her as a diabetes patient would her BMI be just one higher, but an BMI increase to between 32 and 42 would have improved her prediction.

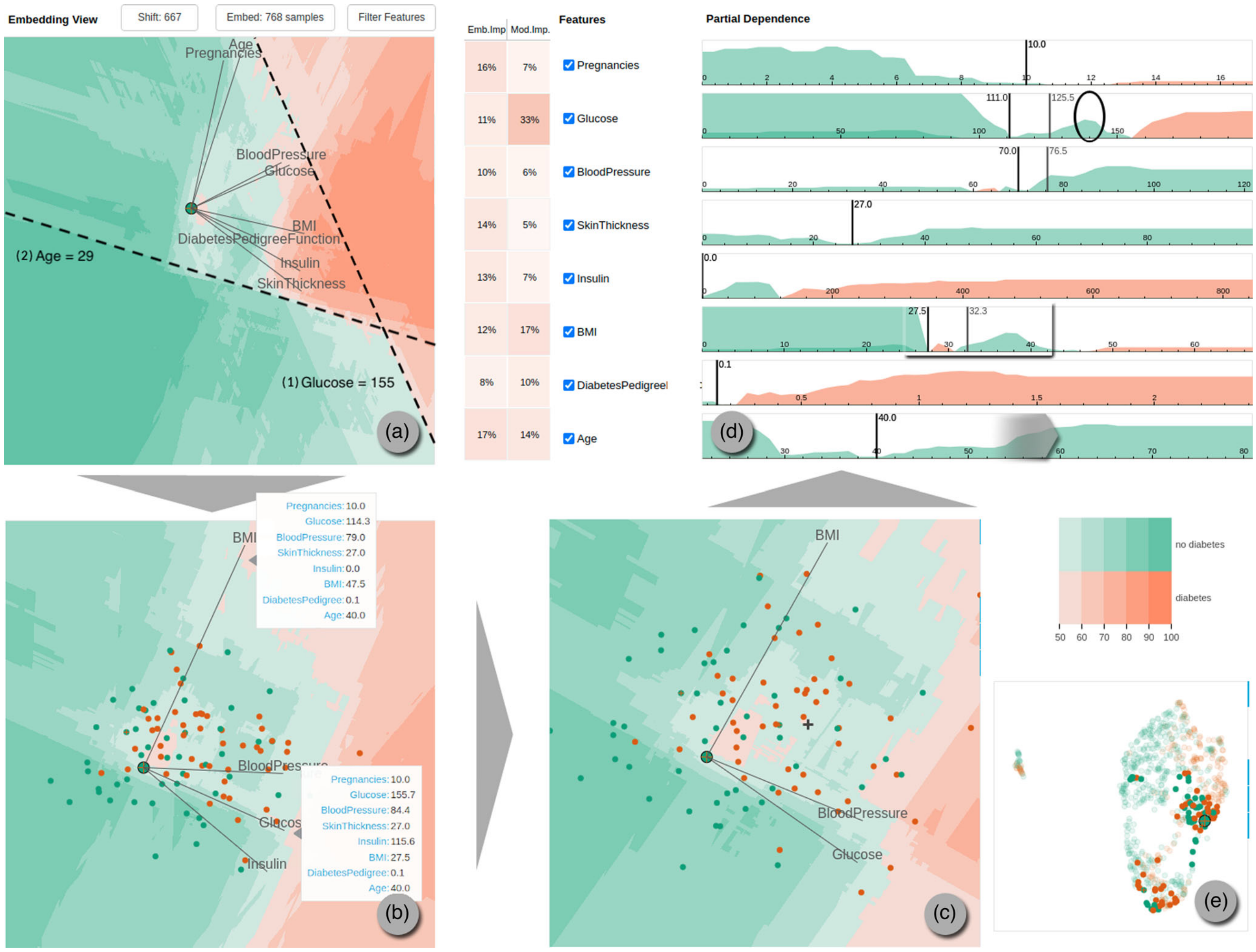


Figure 3: *Workflow of local analysis of f667 in the diabetes dataset. (a) Inspecting global map. (b) Restricting to local neighbourhood and fixing the values of immutable and less-influencial features. (c) Further reduction of axis for closer inspection of boundary artefacts. An unusual counterfactual is found. (d) Comparing the multi-dimensional counterfactual with the expected sensitivity under individual feature changes. (e) Topology overview shows no class separation on non-linear manifold.*

It is likely that this is a model artefact from overfitting to a biased training set. We will find out in the following exploration.

While one-dimensional hypotheses are insightful, the medical measures considered here are unlikely to be changed independently. Figure 3a shows the global linear embedding shifted to be centred at our sample. We can, therefore, explore the prediction results under multi-dimensional alterations to f667's values respecting the global correlation and variance. The axes show three correlated groups of features: pregnancies and age, glucose and blood pressure, and the rest. While we cannot explain the last group, the first two are expected since females with more children are generally older, and glucose increases the blood pressure. It is, therefore, sensible for our embedding to assume that these features change proportionally.

Decision boundaries orthogonal to the respective axes indicate that (1) glucose levels above 155 mg/dl lead to higher prediction even if the blood pressure rises accordingly and regardless of the other features; (2) being younger than 29 with less pregnancies helps reducing the diabetes chance, but is overruled by Glucose values. The exact values of embedding points can be probed with tooltips in the embedding and grey lines in the partial dependence view, demonstrated in Figure 3b and c+d, respectively.

There is also a small orange patch close to f667, which corresponds to the same phenomenon in one-dimensional BMI sensitivity described before. All points on the hyperplane within this patch are predicted to have diabetes. The important difference to a well-formed decision boundary here is that this model behaviour is not generalizable. An algorithmic counterfactual explanation would have been: 'If your BMI was slightly higher, you would have been correctly diagnosed with diabetes'. While this is correct, it leads to the wrong assumption. To overturn her AI-doctor, f667 gains weight and increases her BMI by three. She still is rejected even though she followed the explanation. With our visual inspection, she would have known that this behaviour does not generalize to higher values. In our multi-dimensional linear map, the small size of the patch immediately shows that this behaviour is not applicable for similar feature combinations. It can, therefore, be assumed that the patch is a model artefact from training data that contained an abnormal amount of diabetes patients with similar values as f667. We investigate this further in the next paragraph.

Until now the linear embedding is chosen to approximate the global covariance. However, a dataset can have non-linear associations between variables, so we localize the embedding. We restrict training data of the projection to the nearest 50 neighbours per class. She further cannot readily change skin thickness, pedigree function, age or number of pregnancies. Therefore, we mark these features as immutable as described in Section 4.5. The resulting plot after both changes is shown in Figure 3b. Through the central orange patch, we can see that the class shift by increasing BMI only applies if the other feature values are within a short range. By probing the embedding at the closest points in the orange main area, we formalize a generalizable counterfactual explanation: f667 needs to change her glucose level above 155 mg/dl or her BMI above 47.

Lastly, we demonstrate that multi-dimensional probing is not implicitly achieved by adding individual partial dependencies. We fix the insulin value in the example above to get to Figure 3c. Clicking in an orange area in the embedding view marks the counterfactual with a grey cross and updates the grey lines in the partial dependence view (d). Each individual change reduces the likelihood for diabetes, so their combination is expected to reduce it even further. However, changing all three at once results in flipping the prediction towards diabetes. The need to assume such interactions vanishes with our linear embedding visualization as the likely ones are presented already.

6. Comparison with Related Work

To assess the quality of our embedding technique, we compare it to state-of-the-art decision boundary maps, iLAMP [AVBD*12] and iNN [ERH*19]. We conduct our benchmarks on the breast [DG17], diabetes, robot [FBVV09] and shuttle dataset. To simulate interactive use of our tool CoFFi, we also implemented a naive feature filtration where only features with more than average feature importance are kept. Note that this case is just to show the potential of the tool, since a direct comparison to full-feature maps is unfair. We base our analysis on the general map desiderata *R1–R3* in Section 3.2.

Our map is explicitly defined to convey linear distances based on likelihood of change as discussed in Section 3.3 and therefore fulfils *R1*. On the other hand, Rodrigues *et al.* [REHT19] confirms that the distortion problems of non-linear embeddings translate to non-linear decision maps.

As a measure to favour likely alterations (*R2*), we assume that a likely alteration is a close one. We compute the closest shown counterfactual per sample, as if a user clicked on the closest differently coloured pixel in each of the approaches. In our approach, the average L1-distance in feature space, which is the common measure for counterfactual distances [WMR17], is lower than with iLAMP and iNN in all our experiments (Table 1 $\overline{d_{shown}}$).

Table 1: *Comparison of decision boundary maps iLAMP [AVBD*12], iNN [ERH*19], CoFFi and filtered CoFFi. Shown are the average L1-distances to closest shown counterfactual in normalized feature space $\overline{d_{shown}}$ and Pearson correlation between shown and optimized counterfactual from alibi ρ. The best results of the all-feature approaches are marked in bold.*

		iLAMP	iNN	CoFFi	fil. CoFFi
Breast [DG17]	$\overline{d_{shown}}$	12.41	12.30	**11.92**	5.58
	ρ	0.14	0.14	**0.47**	0.29
Diabetes [RA15]	$\overline{d_{shown}}$	7.56	7.18	**5.07**	3.84
	ρ	−0.07	−0.05	**0.37**	0.48
Robot24 [FBVV09]	$\overline{d_{shown}}$	22.60	22.33	**14.97**	5.61
	ρ	0.03	0.04	**0.17**	0.21
Shuttle [DG17]	$\overline{d_{shown}}$	9.96	9.90	**7.57**	3.46
	ρ	0.16	**0.19**	0.10	0.16

To confirm *R3*, we test how well the shown decision boundary is actually matching with the closest decision boundary. As finding the closest decision boundary is NP-hard, we rely on the closest counterfactual found through optimization as a baseline, which we compute with *alibi* [KLVC21]. In a perfect embedding, the shown counterfactuals should coincide with the optimized ones. Therefore, we compute the Pearson correlation coefficient between the counterfactual distances found through optimization and the embeddings (Table 1 ρ).

In the experiments, our approach found closer counterfactuals on average and in most scenarios, it also showed higher correlation with the ‘optimal’ counterfactuals than previous non-linear approaches. At the same time, our approach relies on a simpler concept with lower computation time. Detailed plots of the benchmark results can be found in Figure 1 of the supplementary material.

7. Discussion and Conclusion

Previous work has shown that decision boundaries provide expressive explanations of individual black-box classifier decisions. Until now, decision boundaries are described by either few counter-examples, discriminating projections of a sparse test dataset, or annotated maps of univariate or non-linear manifolds. We combine the three approaches and present a visual analytics framework for exploring high-dimensional decision boundaries on local and linear maps. Our case study demonstrates that simple, complex and malformed decision boundaries can be conveyed, while explicit probing reveals personalized multi-dimensional counterfactuals with context. Thus, we overcome previous trade-offs between generalizability, explicitness, dimensionality and interpretability. As our method does not require a specific model architecture, it is now possible to gain dense multi-dimensional insight into any classification decision function without extrapolation from examples or accounting for hidden distortion.

Our approach remains with limitations regarding scalability, interactivity and accessibility that are planned directions for future work. As in previous work, visualizing too many sample points in a scatterplots leads to overplotting. While the full dataset can be used for computation and instance-selection, only a sub-set of points should be scatter-plotted. Though we successfully experimented with up to 30 input dimensions in our list, we suggest to move the feature selection process to a plot representation [RFT18] or a recommender system. Due to the limited number of perceived colours [War12], only about eight output-classes can be inspected at once and more classes would need to be grouped. While the proposed projection is significantly faster to compute than previous maps, creating a dense map still depends on the excessive probing of the decision function ($\mathcal{O}(resolution)$). Hence, the interactivity of any map depends on a rapid model evaluation. Finally, the orientation of our hyperplane is fixed via dataset correlation. The customization of explanations can be extended with adherence to the model by class discrimination [AZBZ18, FKM20, SGH15] or the user by movable axes with drag-and-drop [LT13]. Further evaluation of the accessibility also requires a user study with novices and experts, indicating the usefulness of an integration into an ML engineering pipeline in practice.

Acknowledgement

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—252408385—IRTG 2057.

References

[AVBD*12] Amorim E., Vital Brazil E., Daniels J., Joia P., Nonato L., Costa Sousa M.: iLAMP: Exploring high-dimensional spacing through backward multidimensional projection. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology* (Oct. 2012), pp. 53–62. https://doi.org/10.1109/VAST.2012.6400489.

[AVMC*15] Amorim E., Vital Brazil E., Mena-Chalco J., Velho L., Nonato L. G., Samavati F., Costa Sousa M.: Facing the high-dimensions: Inverse projection with radial basis functions. *Computers and Graphics 48* (2015), 35–47.

[AZBZ18] Abid A., Zhang M. J., Bagaria V. K., Zou J.: Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications 9*, 2134 (2018). https://doi.org/10.1038/s41467-018-04608-8.

[BPP*15] Barbosa A., Paulovich F., Paiva A., Goldenstein S., Petronetto F., Nonato L.: Visualizing and interacting with kernelized data. *IEEE Transactions on Visualization and Computer Graphics 22* (Jan. 2015), 1314–1325.

[Bre01] Breiman L.: Random forests. *Machine Learning 45*, 1 (Oct. 2001), 5–32. https://doi.org/10.1023/A:1010933404324.

[BSO15] Bussone A., Stumpf S., O'Sullivan D.: The role of explanations on trust and reliance in clinical decision support systems. In *Proceedings of the 2015 International Conference on Healthcare Informatics* (2015), pp. 160–169. https://doi.org/10.1109/ICHI.2015.26.

[BSR20] Barocas S., Selbst A. D., Raghavan M.: The hidden assumptions behind counterfactual explanations and principal reasons. In *FAT*'20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (New York, NY, USA, 2020), Association for Computing Machinery, pp. 80–89. https://doi.org/10.1145/3351095.3372830.

[CCWH08] Caragea D., Cook D., Wickham H., Honavar V.: *Visual Methods for Examining SVM Classifiers*. Springer Berlin Heidelberg, 2008, pp. 136–153. https://doi.org/10.1007/978-3-540-71080-6_10.

[CLG*15] Caruana R., Lou Y., Gehrke J., Koch P., Sturm M., Elhadad N.: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD'15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), Association for Computing Machinery, pp. 1721–1730. https://doi.org/10.1145/2783258.2788613.

[CMQ20] Cheng F., Ming Y., Qu H.: DECE: Decision explorer with counterfactual explanations for machine learning models. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1. https://doi.org/10.1109/TVCG.2020.3030342.

[CRSPG19] Chapman-Rounds M., Batt U., Schulz M., Pazos E., Georgatzis K.: FIMAP: Feature importance by minimal adversarial perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence. 35*, AAAI Press (Palo Alto, CA, USA, 2021), 11433–1441.

[CYX*20] Chen L., Yan X., Xiao J., Zhang H., Pu S., Zhuang Y.: Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020).

[DCL*18] Dhurandhar A., Chen, P.-Y., Luss R., Tu, C.-C., Ting P., Shanmugam K., Das P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems* (vol. *31*). Curran Associates, Inc. (Red Hook, NY, USA, 2018), pp. 592–603.

[DG17] Dua D., Graff C.: UCI machine learning repository (2017). https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle).

[DMBB20] Dandl S., Molnar C., Binder M., Bischl B.: *Multi-objective Counterfactual Explanations. Lecture Notes in Computer Science*, 2020, pp. 448–469. https://doi.org/10.1007/978-3-030-58112-1_31.

[DPB*19] Dhurandhar A., Pedapati T., Balakrishnan A., Chen P.-Y., Shanmugam K., Puri R.: Model agnostic contrastive explanations for structured data. *arXiv preprint* (2019). https://doi.org/10.48550/arXiv.1906.00117.

[EAS*21] Espadoto M., Appleby G., Suh A., Cashman D., Li M., Scheidegger C. E., Anderson E., Chang R., Telea A.: Unprojection: Leveraging inverse-projections for visual analytics of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, *01* (Nov. 2021), 1. https://doi.org/10.1109/TVCG.2021.3125576.

[ERH*19] Espadoto M., Rodrigues, F. C. M., Hirata, N. S. T., Hirata, Jr. R., Telea A. C.: Deep learning inverse multidimensional projections. In *Proceedings of the EuroVis Workshop on Visual Analytics (EuroVA)* (2019), T. V. Landesberger and C. Turkay (Eds.), The Eurographics Association. https://doi.org/10.2312/eurova.20191118.

[ERHT21] Espadoto M., Rodrigues F., Hirata N., Telea A.: OptMap: Using dense maps for visualizing multidimensional optimization problems. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 1: IVAPP* (2021), INSTICC, SciTePress, pp. 123–132. https://doi.org/10.5220/0010288501230132.

[ERT19] Espadoto M., Rodrigues F., Telea A.: Visual analytics of multidimensional projections for constructing classifier decision boundary maps. In *Proceedings of the International Conference on Information Visualization Theory and Applications* (Jan. 2019), vol. *10* pp. 28–38. https://doi.org/10.5220/0007260800280038.

[FBVV09] Freire A., Barreto G., Veloso M., Varela A.: Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *Proceedings of the 2009 6th Latin American Robotics Symposium, LARS 2009* (Nov. 2009), pp. 1–6. https://doi.org/10.1109/LARS.2009.5418323.

[Few08] Few S.: Time on the horizon (Jan. 2008). http://www.perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf.

[Fis36] Fisher R. A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 2 (1936), 179–188.

[FKM20] Fujiwara T., Kwon O. H., Ma K. L.: Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 45–55.

[FRD19] Fisher A., Rudin C., Dominici F.: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research 20*, 177 (2019), 1–81.

[Fri01] Friedman J. H.: Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*, 5 (2001), 1189–1232.

[Gab71] Gabriel K. R.: The biplot graphic display of matrices with application to principal component analysis. *Biometrika 58*, 3 (1971), 453–467.

[GHYB20] Gomez O., Holter S., Yuan J., Bertini E.: Vice: Visual counterfactual explanations for machine learning models. In *IUI'20: Proceedings of the 25th International Conference on Intelligent User Interfaces* (2020), Association for Computing Machinery, pp. 531–535. https://doi.org/10.1145/3377325.3377536.

[GWE*19] Goyal Y., Wu Z., Ernst J., Batra D., Parikh D., Lee S.: Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning, PMLR* (Long Beach, California, USA, June 2019), K. Chaudhuri and R. Salakhutdinov (Eds.), vol. *97*, pp. 2376–2384. http://proceedings.mlr.press/v97/goyal19a.html.

[HHC*19] Hohman F., Head A., Caruana R., DeLine R., Drucker S.: Gamut: A design probe to understand how data scientists understand machine learning models. In *Proceedings of the SIGCHI* (May 2019), ACM.

[HJ11] Hansen C. D., Johnson C. R.: *Visualization Handbook*. Elsevier, Amsterdam, 2011.

[HKA09] Heer J., Kong N., Agrawala M.: Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proceedings of the ACM Human Factors in Computing Systems (CHI)* (2009). http://vis.stanford.edu/papers/horizon.

[IMI*10] Ingram S., Munzner T., Irvine V., Tory M., Bergner S., Müller T.: Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the 2010 IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 3–10. https://doi.org/10.1109/VAST.2010.5652392.

[JM15] Jordan M. I., Mitchell T. M.: Machine learning: Trends, perspectives, and prospects. *Science 349*, 6245 (2015), 255–260.

[JZF*09] Jeong D. H., Ziemkiewicz C., Fisher B., Ribarsky W., Chang R.: IPCA: An interactive system for PCA-based visual analytics. *Computer Graphics Forum 28* (June 2009), 767–774.

[KLVC21] Klaise J., Looveren A. V., Vacanti G., Coca A.: Alibi explain: Algorithms for explaining machine learning models. *Journal of Machine Learning Research 22*, 181 (2021), 1–7.

[KPN16] Krause J., Perer A., Ng K.: Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (May 2016), Association for Computing Machinery, pp. 5686–5697. https://doi.org/10.1145/2858036.2858529.

[KS20] Keane M. T., Smyth B.: Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In *Case-Based Reasoning Research and Development*. I. Watson and R. Weber (Eds.) Springer International Publishing, Cham (2020), pp. 163–178.

[LC87] Lorensen W. E., Cline H. E.: Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics 21*, 4 (1987), 163–169.

[Lip90] Lipton P.: Contrastive explanation. *Royal Institute of Philosophy Supplement 27* (1990), 247–266.

[LJLH19] Liu Y., Jun E., Li Q., Heer J.: Latent space cartography: Visual analysis of vector space embeddings. In *Proceedings of the Computer Graphics Forum (Proc. EuroVis)* (2019). http://idl.cs.washington.edu/papers/latent-space-cartography.

[LK19] Looveren A. V., Klaise J.: Interpretable counterfactual explanations guided by prototypes. *Machine Learning and Knowledge Discory in Databases. Research Track*, Springer International Publishing, Cham (2021), pp. 650–665.

[LLM*18] Laugel T., Lesot M.-J., Marsala C., Renard X., Detyniecki M.: Comparison-based inverse classification for interpretability in machine learning. In *Proceedings of the IPMU* (2018).

[LT13] Lehmann D. J., Theisel H.: Orthographic star coordinates. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2615–2624.

[MFH*21] Ma Y., Fan A., He J., Nelakurthi A., Maciejewski R.: A visual analytics framework for explaining and diagnosing transfer learning processes. *IEEE Transactions on Visualization and Computer Graphics 27*, 02 (Feb. 2021), 1385–1395.

[MHT18] M Rodrigues F. C., Hirata R., Telea A. C.: Image-based visualization of classifier decision boundaries. In *Proceedings of the 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)* (2018), pp. 353–360. https://doi.org/10.1109/SIBGRAPI.2018.00052.

[MM21] Ma Y., Maciejewski R.: Visual analysis of class separations with locally linear segments. *IEEE Transactions on Visualization and Computer Graphics 27*, 1 (2021), 241–253.

[Mol19] Molnar C.: Interpretable machine learning. a guide for making black box models explainable (2019). https://christophm.github.io/interpretable-ml-book/.

[MPNP21] Mazumdar D., Popolin Neto M., Paulovich F.: Random forest similarity maps: A scalable visual representation for global and local interpretation. *Electronics 10* (Nov. 2021), 2862.

[MQB18] Ming Y., Qu H., Bertini E.: Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics* 1 (2018). https://doi.org/10.1109/TVCG.2018.2864812.

[MST20] Mothilal R. K., Sharma A., Tan C.: Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (Jan. 2020). https://doi.org/10.1145/3351095.3372850.

[NM12] Nam J., Mueller K.: TripAdvisor(N-D): A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Transactions on Visualization and Computer Graphics 19* (Feb. 2012). https://doi.org/10.1109/TVCG.2012.65.

[NP21] Neto M. P., Paulovich F. V.: Explainable matrix—visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (Feb. 2021), 1427–1437.

[OEHT22] Oliveira. A., Espadoto. M., Hirata Jr,. R., Telea. A.,: SDBM: Supervised decision boundary maps for machine learning classifiers. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications—IVAPP* (2022), INSTICC, SciTePress, pp. 77–87. https://doi.org/10.5220/0010896200003124.

[Pea01] Pearson K.: LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[PSSR*20] Poyiadzi R., Sokol K., Santos-Rodriguez R., Bie T., Flach P.: Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AIES'20: AAAI/ACM Conference on AI, Ethics, and Society* (Feb. 2020), pp. 344–350. https://doi.org/10.1145/3375627.3375850.

[RA15] Rossi R. A., Ahmed N. K.: The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI* (2015). http://networkrepository.com.

[RD00] Rheingans P., DesJardins M.: Visualizing high-dimensional predictive model quality. In *Proceedings of the Visualization 2000. VIS 2000 (Cat. No.00CH37145)* (2000), pp. 493–496. https://doi.org/10.1109/VISUAL.2000.885740.

[REHT19] Rodrigues, F. C. M., Espadoto M., Hirata R., Telea A. C.: Constructing and visualizing high-quality classifier decision boundary maps. *Information 10*, 9 (2019). https://doi.org/10.3390/info10090280.

[RFT18] Rauber P. E., Falcão A. X., Telea A. C.: Projections as visual aids for classification system design. *Information Visualization 17*, 4 (2018), 282–305.

[RSG16] Ribeiro M. T., Singh S., Guestrin C.: Why should I trust you?: Explaining the predictions of any classifier. In *KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), Association for Computing Machinery, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778.

[RSG18] Ribeiro M. T., Singh S., Guestrin C.: Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* (2018).

[Rud] Rudiger P.: Panel—a high-level app and dashboarding solution for Python (2022). https://panel.holoviz.org/index.html. Accessed: 2022-02-09.

[SF20] Sokol K., Flach P.: One explanation does not fit all: The promise of interactive explanations for machine learning transparency. *KI - Künstliche Intelligenz 34* (Feb. 2020). https://doi.org/10.1007/s13218-020-00637-y.

[SGH15] Schulz A., Gisbrecht A., Hammer B.: Using discriminative dimensionality reduction to visualize classifiers. *Neural Processing Letters 42* (2015), 27–54.

[SHH20] Schulz A., Hinder F., Hammer B.: Deepview: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (July 2020), International Joint Conferences on Artificial Intelligence Organization, pp. 2305–2311. https://doi.org/10.24963/ijcai.2020/319.

[SSTea20] Schramowski P., Stammer W., Teso S., Brugger A., Herbert F., Shao X., Luigs H.-G., Mahlein A.-K., Kersting K.: Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence 2* (2020), 476–486.

[TGR20] Tsirtsis S., Gomez-Rodriguez M.: Decisions, counterfactual explanations and strategic behavior. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*, Curran Associates Inc., (Red Hook, NY, USA, 2020).

[TMF*12] Tatu A., Maass, F., Färber I., Bertini E., Schreck T., Seidl T., Keim D.: Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), pp. 63–72. https://doi.org/10.1109/VAST.2012.6400488.

[vWvL93] van Wijk J., van Liere R.: HyperSlice: Visualization of scalar functions of many variables. In *Proceedings IEEE Visualization'93* (1993), pp. 119–125.

[War12] Ware C.: *Information Visualization: Perception for Design. Morgan Kaufmann Series in Interactive Technologies* (3rd edition). Morgan Kaufmann, Amsterdam, 2012. http://www.sciencedirect.com/science/book/9780123814647.

[WFC*18] Wang Y., Feng K., Chu X., Zhang J., Fu C., Sedlmair M., Yu X., Chen B.: A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics 24*, 5 (2018), 1828–1840.

[WMR17] Wachter S., Mittelstadt B. D., Russell C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology (31/2)*, Harvard University Press, Cambridge, MA, USA (2018).

[WPB*19] Wexler J., Pushkarna M., Bolukbasi T., Wattenberg M., Viégas F. B., Wilson J.: The What-If Tool: Interactive probing of machine learning models. In *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), pp. 56–65. https://doi.org/10.1109/TVCG.2019.2934619.

[XYC*18] Xia J., Ye F., Chen W., Wang Y., Chen W., Ma Y., Tung A.: LDSScanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets. In *IEEE Transactions on Visualization and Computer Graphics24*, *1* (2018), 236–245.

[YRWG13] Yuan X., Ren D., Wang Z., Guo C.: Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics 19* (Dec. 2013), 2625–2633.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supporting Information

Supporting Video