

“What is Different Between These Datasets?”

A Framework for Explaining Data Distribution Shifts

Varun Babbar*

*Department of Computer Science
Duke University*

VARUN.BABBAR@DUKE.EDU

Zhicheng Guo*

*Department of Electrical and Computer Engineering
Duke University*

ZHICHENG.GUO@DUKE.EDU

Cynthia Rudin

*Department of Computer Science
Duke University*

CYNTHIA@CS.DUKE.EDU

Editor: Mingyuan Zhou

Abstract

The performance of machine learning models relies heavily on the quality of input data, yet real-world applications often face significant data-related challenges. A common issue arises when curating training data or deploying models: two datasets from the same domain may exhibit differing distributions. While many techniques exist for detecting such distribution shifts, there is a lack of comprehensive methods to explain these differences in a human-understandable way beyond opaque quantitative metrics. To bridge this gap, we propose a versatile framework of interpretable methods for comparing datasets. Using a variety of case studies, we demonstrate the effectiveness of our approach across diverse data modalities—including tabular data, text data, images, time-series signals – in both low and high-dimensional settings. These methods complement existing techniques by providing actionable and interpretable insights to better understand and address distribution shifts.

Keywords: dataset-differences, interpretability, dataset-comparison, data-analysis, explainable-AI, distribution-shift explanation

*These authors contributed equally to this work.

Table of Contents

1	Introduction	4
2	Related Works	5
2.1	Distribution Shift	5
2.2	Instance-level Explanations	5
2.3	Dataset-level Explanations	6
2.4	Synthetic Data	6
2.5	Prototype Learning	6
2.6	Rashomon Effect	6
3	The Dataset Explanation Framework	7
3.1	Overview of Methodology	7
3.2	Overview of Paper Structure	8
3.3	Explanations Based on Influential Examples	9
3.3.1	Introduction	9
3.3.2	Definitions of Relevant Importance Measures	10
3.3.3	Determining the Influential Examples	11
3.3.4	Case Study 1: Low Dimensional Tabular Data - Adult Dataset	12
3.3.5	Case Study 2: High Dimensional Tabular Data - HELOC Dataset	13
3.4	Prototype-Neighbourhood-Based Explanations	15
3.4.1	Introduction	15
3.4.2	Quantitative Comparison Between neighborhoods	16
3.4.3	Comparing Prototype Neighborhoods in \mathcal{D} and \mathcal{D}' in high dimensions	17
3.4.4	Case Study 1: Low Dimensional Tabular Data - Adult Dataset	20
3.4.5	Case Study 2: High-Dimensional Tabular Data - HELOC Dataset	21
3.4.6	Case Study 3: Time Series Medical Data - Cardiac Signals	23
3.5	Prototype-Summarization-Based Explanations	25
3.5.1	Introduction	25
3.5.2	Summarization Prototype Learning for Dataset Comparisons	25
3.5.3	Case Study 1: Time Series Data - Human vs Machine Audio	27
3.5.4	Case Study 2: Medical Image Data - Mammography Patient Population Dataset	30
3.5.5	Case Study 3: Image Data - Office-Home Dataset	32
3.6	A Brief Note on Comparing Natural Language Datasets	33
4	Discussion	33
4.1	Practical Guidance for Using Our Framework	33
4.2	Potential Failure Modes and Avenues for Future Research	34
5	Conclusion	36
6	Ethics note	36
7	Acknowledgement	36
A	Cardiac Signal Simulation Parameters	43
B	Evaluation of Prototypical Explanations	43
B.1	Prototype-Based Explanations for NSPD and NSDD	43
B.2	Choosing Partial Prototypes	44
B.3	Partial Prototype Feature Selection: Sensitivity Analysis	46
B.4	All learned summarization prototypes for the Office-Home Experiment	47
B.5	Robustness of Prototype-summarization-based explanations	48
B.6	Are Prototypical Neighborhoods Faithful for Vision and Signal Data?	51

C	Evaluation of Influential Example Explanations	56
C.1	Influential Example Explanations: Alignment	56
C.2	Influential Example Explanations: Robustness	57
C.3	Influential Example Explanations: Effect of Rashomon Set	59
D	Illustrating a Failure Mode of Kulinski and Inouye (2023)	60
D.1	Methodology	60
D.2	Case 1: Our explanation and Kulinski and Inouye (2023) is coherent	61
D.3	Case 2: Our explanation is coherent but Kulinski and Inouye (2023) is not	62
E	Text Data Difference Analysis with “<i>Exact Counts</i>”	62

1 Introduction

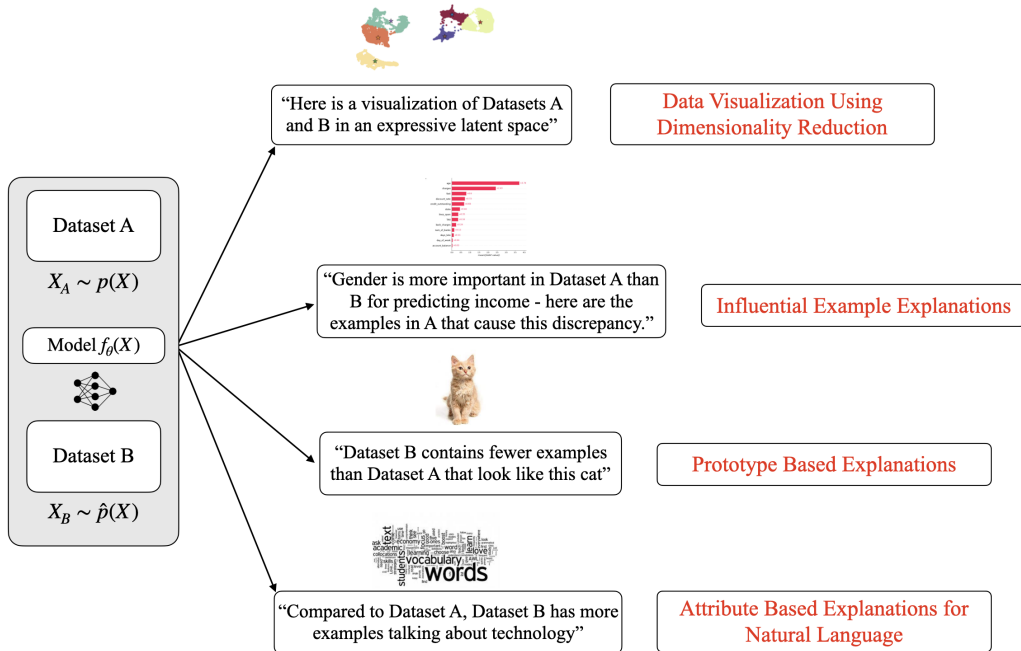


Figure 1: An illustration of our dataset explanation framework. This repertoire of explanations enables the user to gain insights on differences between distribution shifted datasets, with applicability across different modalities. Notably, these explanations do not all require machine learning models trained on the datasets.

Some of the most serious challenges facing the data revolution involves data itself: it is often hard to acquire, hard to share, hard to generate, and hard to troubleshoot. If we generate more data, how do we know it follows the same distribution as our original dataset? If we obtain datasets from different sources, how do we know what is different between them? These questions about data generation and comparisons are important: they arise when we generate medical datasets to protect patient privacy (Chen et al., 2021; Tucker et al., 2020; Giuffrè and Shung, 2023), generate larger synthetic datasets to augment small true datasets, study data from multiple related sources, or try to determine whether distribution shift has occurred (Guo et al., 2022; Chirra et al., 2018; Gao et al., 2022; Yang et al., 2023). Thus, it is important to be able to understand the *differences between datasets*.

Most previous works in this direction studied distribution shift, focusing on detecting whether or not distribution shift has occurred, as well as detecting differences in statistical features between datasets (e.g., mean, median, and variance, etc.) We claim that knowing whether changes have occurred is not good enough, nor is viewing the data through a few basic statistical measurements such as Wasserstein distance and KL divergence. Understanding the true nature and extent of the changes can help human operators make more informed and effective decisions.

In this work, we propose an explainable AI framework for examining and comparing the differences between two distribution shifted datasets, providing detailed and actionable information. We provide approaches for several data modalities, including high-dimensional complex data, with examples in audio, time series signal, image, and text data. Our framework is summarized in Figure 1. It encompasses a variety of explanation types, including prototype explanations (e.g., “Dataset B contains fewer examples that look like this”), explanations that involve feature importance (“these examples are why feature K is more important for Dataset A than B ”), and explanations that compare interpretable attributes of natural language datasets. Most explanations are accompanied by visualizations that allow users to examine high-dimensional data and samples. Note that we are not aiming to provide an *exhaustive* list of methods, as there are an infinite number of ways one could examine the difference between two datasets, and sorting through these could easily be overwhelming; instead, we aim for a small set of good methods that will suffice in most cases.

In Figure 2, we illustrate the distinction between traditional explainable AI (XAI) approaches and our specific task. Existing XAI methods primarily focus on elucidating the reasoning behind a specific model’s decisions on an individual sample basis, as depicted in the left examples of Figure 2. However, such methods

are inherently tied to model behavior and are not well-suited for explaining dataset-level differences. Since dataset differences do not necessarily depend on a specific model, analyzing them purely through model-based explanations can be limiting. In contrast, our goal is to develop model-agnostic methods that provide dataset-level explanations—capturing systematic differences directly from the data itself — as illustrated in the rightmost examples of Figure 2.

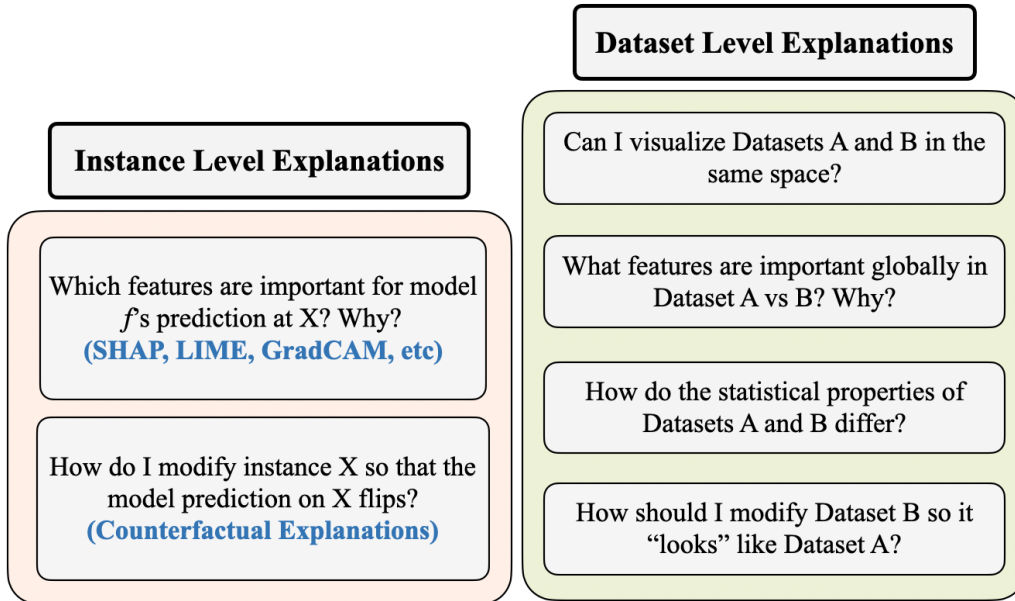


Figure 2: Highlights of the differences between explanations at the instance-level vs. those at the dataset-level.

2 Related Works

We introduce and discuss several related previous works to this study.

2.1 Distribution Shift

Our study is adjacent to distribution shift analysis, though our focus is broader: we do not focus on any specific type of distribution shift (such as covariant shift, Sugiyama et al. 2007, or label shift, Zhang et al. 2013), rather we focus on the changes between datasets with no particular assumptions on the type of shift. Previous efforts have largely focused on the detection and analysis of the shifts (Sun et al., 2021; Jang et al., 2022; Yang et al., 2021a), and the improvement of model generalizability to alleviate the effects of distribution shift (Hendrycks et al., 2021; Jang et al., 2022; Sun et al., 2020; Shen et al., 2021). However, to the best of our knowledge, most works have not explored explaining distribution shifts in a human-understandable manner. The closest work to ours from this literature is possibly that of Zhang et al. (2023), who proposed an approach to attribute model performance changes due to distribution shifts based on Shapley values (Shapley, Lloyd S., 1953). We focus more broadly on explaining differences between datasets, with no requirements of prior knowledge or a task-related model.

2.2 Instance-level Explanations

The conventional instance-level explanation literature has largely focused on post-hoc analysis, i.e., analysing a prediction from a trained model. Some well-known work (Ribeiro et al., 2016, 2018) has focused on learning simpler explanation functions that approximate the model around the neighbourhood of a point. The output of these functions is a score for each feature representing its contribution to a given prediction. The feature importance-based explanation literature has also examined methods that compute the gradient of the prediction with respect to the input (Lundberg and Lee, 2017; Selvaraju et al., 2017; Simonyan et al., 2013; Smilkov et al., 2017; Sundararajan et al., 2017a). Another line of research focuses on counterfactual

explanations (Yang et al., 2022; Ustun et al., 2019; Antoran et al., 2021; Upadhyay et al., 2021), which provide changes to a given instance so that the model flips its prediction (or, in the case of Antoran et al. (2021), becomes certain of its prediction). As far as we can tell, none of these types of approaches can be applied to explaining the difference between two datasets; instead, they all explain a model.

2.3 Dataset-level Explanations

To the best of our knowledge, there is very limited literature on dataset-level explanations. The most relevant work on dataset-level explanations is that of Kulinski and Inouye (2023), which uses optimal transport (Peyré and Cuturi, 2019) maps to explain mean shifts in distributions of the datasets (or individual clusters). The user is provided with the original clusters and the transported clusters and can visually inspect the difference between the two to derive insights. However, their method focuses exclusively on mean shifts between clusters and requires both datasets to be of the same size, which can be limiting (see Section D in the appendix for an example). Shin et al. (2022) provides dataset-level explanations for graph classification tasks by comparing examples in a dataset to salient sub-graph prototypes frequently observed in the dataset. Zhu et al. (2022) introduce natural language explanations for visual datasets. In particular, for each attribute or class in the dataset, the explanation consists of the K most salient image samples in dataset \mathcal{D} , their shifted versions in dataset \mathcal{D}' , and a natural language description of their differences. This method depends on having a 1-to-1 correspondence between items in \mathcal{D} and \mathcal{D}' that are not usually available.

For textual data, Elazar et al. (2023) explores properties of several large-scale text corpora to uncover insights on the relative presence of attributes such as toxicity, level of contamination, and n-gram statistics.

2.4 Synthetic Data

One major application of our work lies in explaining the difference between real data and synthetic datasets. Most frameworks for evaluating real and synthetic data focus on the statistical properties of the datasets or evaluate the quality of the generative models. Livieris et al. (2024) construct an evaluative framework for synthetic data generating models, providing metrics that quantify statistical faithfulness. However, while these metrics are very useful, they only provide a limited picture of the synthetic dataset. Neto et al. (2024), on the other hand, create annotated attributes for known real and synthetic face recognition datasets and compare the data along these attributes. While this emulates our philosophy of providing interpretable dataset explanations, their main findings are specific to the domain of face recognition (including the attributes they picked) and may not be directly applicable to other kinds of datasets. In this work, we show that our approaches are general enough to uncover underlying intricacies of synthetic data that distinguish it from real data, such as the quality of cluster substructures and properties of influential groups of datapoints.

2.5 Prototype Learning

In recent years, ProtoPNet (Chen et al., 2019a), a type of prototype network, has been introduced as an inherently interpretable neural network capable of providing explanations through case-based reasoning for its predictions. Specifically, ProtoPNet has been built into a popular framework where images are classified by comparing specific parts of an image to prototypical parts associated with each class. Subsequent developments have expanded on the original ProtoPNet algorithm (Chen et al., 2019a), focusing primarily on enhancing the components of ProtoPNet itself (Donnelly et al., 2022; Nauta et al., 2021b; Rymarczyk et al., 2022, 2021; Wang et al., 2023; Ma et al., 2024; Wang et al., 2021a; Nauta et al., 2021a), refining the training regimen (Rymarczyk et al., 2023; Nauta et al., 2023; Willard et al., 2024), or adapting ProtoPNets for high-stakes applications (Yang et al., 2024; Barnett et al., 2021; Choukali et al., 2024; Wei et al., 2024; Barnett et al., 2023). Although we utilize the underlying prototype learning mechanism, our focus differs significantly from traditional applications in the prototype learning literature. These methods have the ultimate goal of performing classification given the underlying task. Our proposed approach aims to compare distribution shifted datasets, and can operate on both labeled and unlabeled datasets. This fundamental shift highlights one of the unique challenges and goals of our methodology.

2.6 Rashomon Effect

Our research leverages the Rashomon Effect to evaluate feature importance. This phenomenon is the existence of multiple, diverse models that achieve similar predictive performance for the same task (Breiman, 2001). The Rashomon Effect presents both challenges and opportunities: it gives rise to predictive multiplicity –

where different models yield varying predictions for the same instance (Marx et al., 2020; Watson Daniels et al., 2023a; Kulynych et al., 2023; Hsu and Calmon, 2022; Watson Daniels et al., 2023b). Also, the set of high-quality models for a given dataset can disagree on which variables are important (Fisher et al., 2019; Dong and Rudin, 2020; Smith et al., 2020). Interestingly, compiling all these good models yields something better than what can be obtained with any single model – a robust, model-agnostic method for assessing variable importance, called the Rashomon Importance Distribution (Donnelly et al., 2023). In this study, we focus on the Rashomon set, the collection of highest-performing models, to construct reliable and robust feature importance measures that help distinguish between datasets.

3 The Dataset Explanation Framework

3.1 Overview of Methodology

In this paper, we aim to illuminate the differences between distribution shifted datasets \mathcal{D} and \mathcal{D}' consisting of features X and possibly labels Y , i.e., $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$ and $\mathcal{D}' = \{(X'_i, Y'_i)\}_{i=1}^{N'}$. Y is not always required - our framework contains methods to deal with both supervised and unsupervised data. Our primary assumption in this work is that \mathcal{D} and \mathcal{D}' belong to the same domain (e.g., both consist of animal images), but other properties of the datasets and their corresponding task models may differ, such as feature and class distributions, (latent) cluster structure, and model performance metrics. While these aspects of datasets are relatively easy to capture, what is not trivial is producing actionable insights into dataset differences. For instance, using our explanation framework, we can reveal that \mathcal{D}' lacks examples of a certain archetype that are more prevalent in \mathcal{D} , how structural properties of the datasets differ, and certain intrinsic biases in either dataset that cause differing feature importances between \mathcal{D} and \mathcal{D}' .

Our pipeline for exploring the differences between datasets is illustrated in Figure 3, which includes dimension reduction for data visualization, as well as three novel algorithms:

- Influential example-based explanations are discussed in Section 3.3. These explanations help uncover subgroups in datasets \mathcal{D} and \mathcal{D}' that cause differences in their feature importances. Applicable for supervised data.
- Prototype-based explanations. See Sections 3.4 and 3.5. These explanations help compare local neighborhoods between datasets \mathcal{D} and \mathcal{D}' . They are accompanied by comparative visualizations using dimensionally reduction methods. They can be used for both supervised and unsupervised data.
- Large Language Model (LLM)-based explanations using interpretable attributes, see Section 3.6. Can be used to compare text corpora.

The first two explanations involve generating, analysing, and comparing salient samples and their features in either dataset. The final explanation involves creating interpretable attributes for each dataset and examining the dataset in terms of those attributes.

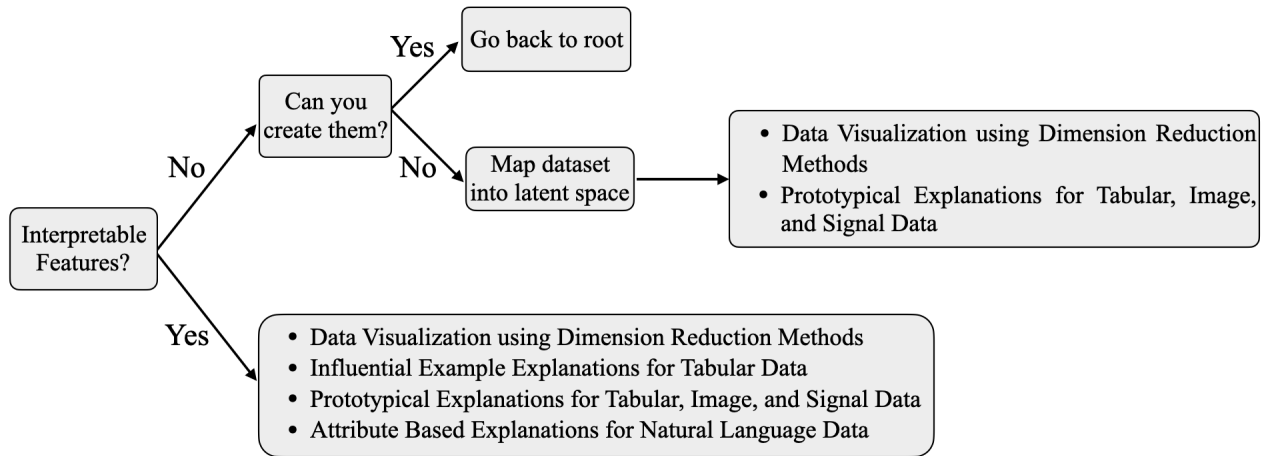


Figure 3: Pipeline for our explanation framework. We provide explanation methods that are applicable across many data modalities. Some of our methods leverage interpretable features and explain dataset differences in terms of those features. When the features are uninterpretable (e.g., individual tokens in natural language), one could potentially create proxy attributes that are interpretable and explain the datasets in terms of those attributes, or use prototypical explanations and dimension reduction projections. We use PacMAP Wang et al. (2021b) as the dimension reduction method in this paper, as it currently offers state of the art performance.

3.2 Overview of Paper Structure

Different data modalities and tasks require different types of explanations. For instance, using influential example-based explanations is appropriate for tabular data, where the feature values are interpretable. However, for image and signal data with non-interpretable features (e.g., a pixel value or a signal value at time t), feature importance would not be as interpretable for humans. In the following subsections, we describe each method in our framework and provide several supporting case studies of different tasks and data modalities utilizing the proposed approaches.

3.3 Explanations Based on Influential Examples

3.3.1 INTRODUCTION

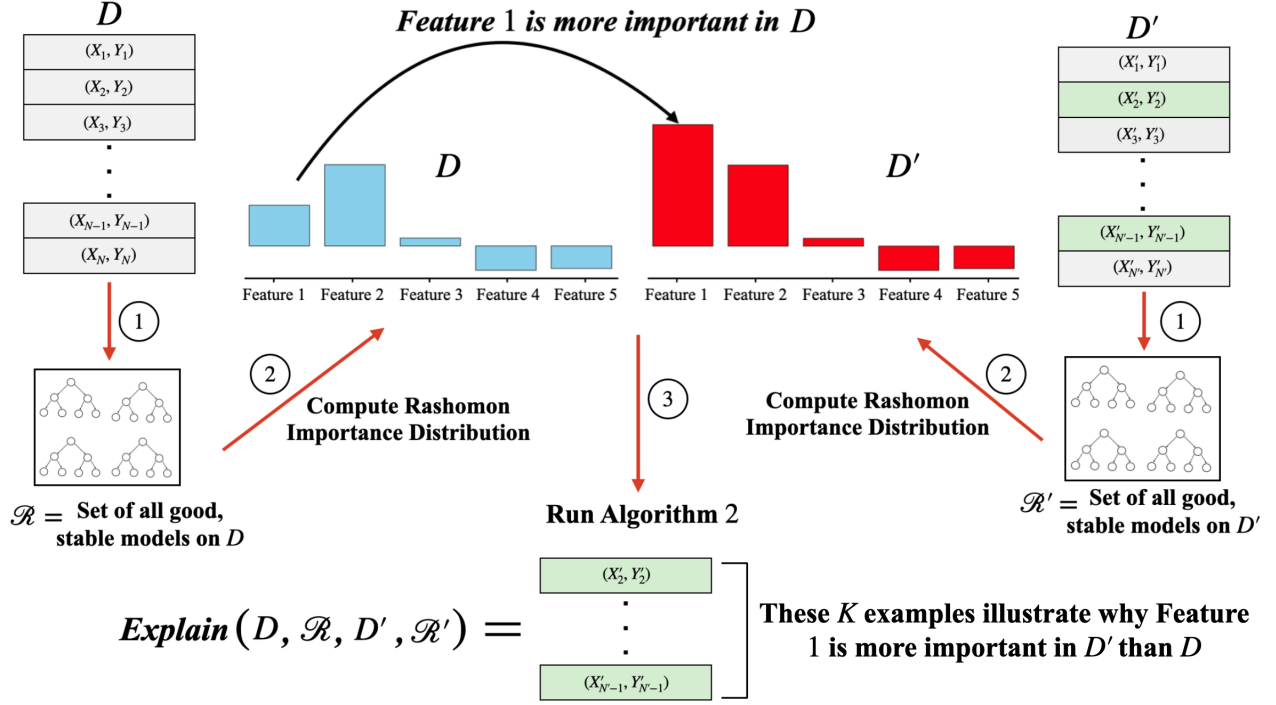


Figure 4: An illustration of influential example explanations. Given datasets \mathcal{D} and \mathcal{D}' and a feature importance metric, our explanation gives us K influential examples (the user can choose whether they are from \mathcal{D} or \mathcal{D}') that are most responsible for the feature importances being different between \mathcal{D} and \mathcal{D}' . These feature importances are computed from the set of all nearly optimal, stable decision trees (where stability means that the model is also nearly optimal for perturbations of \mathcal{D}) – we show how to compute these below. A practitioner can uncover specific patterns that distinguish these K examples – we illustrate this in Sections 3.3.5 and 3.3.4.

This section examines explanations that take into account differences between datasets by considering *which features are intrinsically important in both datasets relative to the underlying task*. An intrinsically important feature is one whose importance for the underlying data distribution remains stable across multiple well-trained models and perturbations of the dataset. Donnelly et al. (2023) show that not considering this model-agnostic representation of feature importance can cause researchers to arrive at multiple equally valid – yet contradictory – conclusions about the data. After determining the intrinsic importance of features, we ask the question: *Given Datasets \mathcal{D} and \mathcal{D}' , which K examples from Dataset \mathcal{D}' should I remove so that the intrinsic importance of features in both datasets for the underlying task are as similar as possible?* To the best of our knowledge, this is a novel way of looking at two datasets while taking into account an underlying task (e.g., classification). To determine intrinsic feature importances for a labeled dataset, we employ the Rashomon Importance Distribution (RID) framework of Donnelly et al. (2023). Donnelly et al. (2023) show that this method, which quantifies the importance of a feature across the set of all good models in a class, results in feature importances that are highly robust to dataset perturbations. Given a dataset \mathcal{D} , a hypothesis class \mathcal{F} , regularization strength λ , and tolerance ϵ , the Rashomon set \mathcal{R} is defined as the set of all models in \mathcal{F} whose empirical losses are within ϵ of the minimum empirical loss (Semenova et al., 2022):

$$\mathcal{R}(\mathcal{D}, \epsilon, \mathcal{F}, \lambda) = \{f \in \mathcal{F} : \ell(f, \mathcal{D}, \lambda) \leq \min_{f' \in \mathcal{F}} \ell(f', \mathcal{D}, \lambda) + \epsilon\}, \text{ where} \quad (1)$$

$$\ell(f, \mathcal{D}, \lambda) = \frac{1}{|\mathcal{D}|} \sum_{Z=(X,Y) \in \mathcal{D}} L(Z, f) + \lambda R(f) \quad (2)$$

is the regularized empirical loss for the dataset, with loss function L and model sparsity $R(f)$ (e.g. the number of leaves in a decision tree). Our intrinsic variable importance will average a variable importance metric over Rashomon sets constructed on bootstrap samples.

3.3.2 DEFINITIONS OF RELEVANT IMPORTANCE MEASURES

Before we introduce the method to compute the intrinsic feature importances, we first define the following terms:

Definition 1 (Local Feature Importance Measure – LFIM) *Given a predictor f from a hypothesis class \mathcal{F} and a dataset \mathcal{D} with M features, a local feature importance measure is a function $\phi(f, X, Y) : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^M$ that outputs a vector representing the relative contribution of each feature towards the output prediction Y for a specific input X . A lot of work has been devoted to the development of faithful feature importance measures (Ribeiro et al., 2016; Lundberg and Lee, 2017; Donnelly et al., 2023) – in principle, any of these can be used in our explanation framework. We assume that this feature importance measure is a property of the dataset and the model in question.*

Definition 2 (Global Feature Importance Measure – GFIM) *Given a predictor f from a hypothesis class \mathcal{F} and a dataset \mathcal{D} with M features, a global feature importance measure $\phi_g(f, \mathcal{D}) : \mathcal{F} \times \mathcal{D} \rightarrow \mathbb{R}^M$ will provide a similar vector as an LFIM, except that it represents the predictive power of each feature in the entire dataset. In this paper, we consider GFIM to be the average LFIM vector across all examples in a dataset, i.e., $\phi_g(f, \mathcal{D}) = \mathbb{E}_{(X,Y) \in \mathcal{D}}[\phi(f, X, Y)]$.*

Definition 3 (Local Intrinsic Feature Importance Measure – LiFIM) *Given a dataset \mathcal{D} with M features, a local intrinsic feature importance measure $\phi(X, Y, \mathcal{D}) : \mathcal{X} \times \mathcal{Y} \times \mathcal{D} \rightarrow \mathbb{R}^M$ for an example $(X, Y) \in \mathcal{D}$ computes the importance of each feature in (X, Y) by aggregating the LFIMs of well-trained, stable models in \mathcal{F} . This involves computing Rashomon sets of bootstrapped samples from \mathcal{D} , storing models associated with each set and aggregating their LFIMs. The precise technique is detailed below in this section.*

Definition 4 (Global Intrinsic Feature Importance Measure – GiFIM) *Given a dataset \mathcal{D} with M features, a global feature importance measure $\phi_g(\mathcal{D}) : \mathcal{D} \rightarrow \mathbb{R}^M$ will provide a similar vector as an LiFIM, except that it represents a holistic summary of the intrinsic predictive power of each feature across an entire dataset. In this paper, we consider GiFIM to be the average of LiFIMs across all examples in a dataset, i.e., $\phi_g(\mathcal{D}) = \mathbb{E}_{(X,Y) \in \mathcal{D}}[\phi(X, Y, \mathcal{D})]$.*

Under the framework of Donnelly et al. (2023), we can compute the LiFIM and GiFIM of models in the following manner:

- Bootstrap the dataset \mathcal{D} B times.
- For each bootstrapped dataset \mathcal{D}_i , compute its Rashomon set $\mathcal{R}(\mathcal{D}_i, \epsilon, \mathcal{F}, \lambda)$. For decision trees, this can be done using TreeFARMS (Xin et al., 2022).
- Compute the LFIMs of each example under each model in each Rashomon set using any method in literature (here, we use SHAP of Lundberg and Lee, 2017). Under computational constraints, a random sample of models from each Rashomon set can also be used.
- The LiFIM $\phi(X, Y, \mathcal{D})$ for an example $(X, Y) \in \mathcal{D}$ is computed by taking the mean (over bootstraps and Rashomon sets) of feature importances. That is:

$$\phi(X, Y, \mathcal{D}) = \frac{1}{B} \sum_{i=1}^B \frac{1}{|\mathcal{R}(\mathcal{D}_i, \epsilon, \mathcal{F}, \lambda)|} \sum_{f \in \mathcal{R}(\mathcal{D}_i, \epsilon, \mathcal{F}, \lambda)} \phi(f, X, Y). \quad (3)$$

If a model appears more than once across different Rashomon sets, this results in that model’s feature importance vector having a larger contribution to the final LiFIM.

- The GiFIM $\phi_g(\mathcal{D})$ for the dataset \mathcal{D} is the average of LiFIMs in the dataset, i.e. $\phi_g(\mathcal{D}) = \mathbb{E}_{(X,Y) \in \mathcal{D}}[\phi(X, Y, \mathcal{D})]$.

In this paper, given \mathcal{D} and \mathcal{D}' , the influential example explanation provides the following information to the user: *A set of K examples (from either \mathcal{D} or \mathcal{D}' that), if removed from the dataset, would align the GiFIMs of \mathcal{D} and \mathcal{D}' the most.* Concretely, let $\phi_g(\mathcal{D})$ and $\phi_g(\mathcal{D}')$ be the GiFIMs on \mathcal{D} and \mathcal{D}' . We aim to find the set S of K examples $S = \{(X_{[1]}, Y_{[1]}), \dots, (X_{[K]}, Y_{[K]})\}$ in \mathcal{D}' such that $d(\phi_g(\mathcal{D}), \phi_g(\mathcal{D}' \setminus S))$ is minimized, where $d(\cdot, \cdot)$ is the Euclidean distance metric between two vectors. That is, \mathcal{D} and $\mathcal{D}' \setminus S$ will have more aligned intrinsic feature importances. Figure 4 illustrated the underlying intuition behind these explanations. We now explain how we obtain these K influential examples.

3.3.3 DETERMINING THE INFLUENTIAL EXAMPLES

In order to provide influential example explanations, we first define the notion of *influence* for a test loss function.

Definition 5 (Influence Function for Test Loss (Koh and Liang, 2017)) *Given the following:*

- training and test datasets $\mathcal{D}_{train} = \{Z_i^{train} = (X_i^{train}, Y_i^{train})\}_{i=1}^{N_{train}}$, and $\mathcal{D}_{test} = \{Z_i^{test} = (X_i^{test}, Y_i^{test})\}_{i=1}^{N_{test}}$,
- a trained, parameterized model $m_\theta(x)$,
- the minimizer of the training loss: $\hat{\theta} = \operatorname{argmin}_\theta \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} L(Z_i^{train}, m_\theta)$,
- the empirical test loss $L_{test}(m_{\hat{\theta}}) = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} L(Z_i^{test}, m_{\hat{\theta}})$,

an influence function for training point $(X_i^{train}, Y_i^{train})$ estimates the theoretical change in the test loss $L_{test}(m_{\hat{\theta}})$ if the model m_θ is trained using $\mathcal{D}_{train} \setminus (X_i^{train}, Y_i^{train})$. By applying techniques from Koh and Liang (2017), we can write this as:

$$I(Z_i^{train}, \mathcal{D}_{test}, m) = \sum_{j=1}^{N_{test}} \frac{1}{N_{test}} \nabla_\theta L(Z_j^{test}, m_{\hat{\theta}})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(Z_i^{train}, m_{\hat{\theta}}). \quad (4)$$

where $H_{\hat{\theta}}$ is the Hessian of the parameters θ evaluated at $\theta = \hat{\theta}$. This is essentially an approximation of the following form:

$$I(Z_j^{train}, \mathcal{D}_{test}, m) \approx L_{test}(m_{\hat{\theta}}) - L_{test}(m_{\hat{\theta}_{-Z_j^{train}}}) \quad (5)$$

where

$$\hat{\theta}_{-Z_j^{train}} = \operatorname{argmin}_\theta \left(\left(\frac{1}{N_{train}} \sum_{i=1}^{N_{train}} L(Z_i^{train}, m_\theta) \right) - \frac{1}{N_{train}} L(Z_j^{train}, m_\theta) \right) \quad (6)$$

is the set of parameters that minimize the loss on all training examples except Z_j^{train} .

Algorithm 1 Influential Example Dataset Difference Explanations Based on Feature Importance

Require: $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$, $\mathcal{D}' = \{(X'_i, Y'_i)\}_{i=1}^{N'}$

- 1: Let $\mathcal{D}_\phi = \{(\phi(X, Y, \mathcal{D}), 1) \text{ if } (X, Y) \in \mathcal{D} \text{ else } (\phi(X, Y, \mathcal{D}'), 0), \forall (X, Y) \in \mathcal{D} \cup \mathcal{D}'\}$ be the dataset of LiFIMs and corresponding labels computed from both datasets \mathcal{D} and \mathcal{D}' (using Donnelly et al. (2023))
 - 2: Train a logistic regression model $m_\theta(X)$ to classify \mathcal{D} vs \mathcal{D}' using the dataset \mathcal{D}_ϕ
 - 3: Scores = \emptyset
 - 4: **for** each example $Z' \in \mathcal{D}_\phi$ **do**
 - 5: $s_{Z'} = I(Z', \mathcal{D}_\phi, m_\theta)$ ▷ This is computed using Equation 4
 - 6: Add $s_{Z'}$ to Scores
 - 7: **return** The K examples in \mathcal{D}' with the highest $s_{Z'}$ in Scores
-

Algorithm 1 finds the examples that are most detrimental to the performance of the discriminator (i.e., have the highest positive influence value $I(Z', \mathcal{D}', m_\theta)$). Because the discriminator learns to distinguish

between \mathcal{D} and \mathcal{D}' based on their respective feature importance measures, removing the examples found by our algorithm will make the remaining feature importances look more indistinguishable. That is, once we find the set $S \in \mathcal{D}$ of examples to remove, $d(\phi_g(\mathcal{D}), \phi_g(\mathcal{D}' \setminus S))$ will become smaller – we also demonstrate this through empirical studies later. Knowledge of these influential examples can be valuable to the end user, not only to precisely understand the properties of ‘culprit’ examples that make \mathcal{D} and \mathcal{D}' different, but also to design ways to remediate this difference by generating or removing certain examples.

3.3.4 CASE STUDY 1: LOW DIMENSIONAL TABULAR DATA - ADULT DATASET

The Adult dataset contains demographic information from the 1994 US Census database. In particular, each data point corresponds to information on age, sex, education levels, marital status, race, and occupation of an individual. The underlying task is to predict if the annual income of the individual is $\geq \$50K$. In this section, we aim to explain the difference between Adult male and females. This analysis sheds light on potential biases in the dataset, which may affect model predictions and subsequently influence decision-making processes.

Dataset \mathcal{D} \mathcal{D} corresponds to the dataset of all males, but with the same subset of features as Kulinski and Inouye (2023) – age, education, and income. The income feature is encoded as 1 if the annual income is $\geq \$50k$ and 0 otherwise.

Dataset \mathcal{D}' This is the dataset of all females, preprocessed in the same manner as \mathcal{D} . Thus, we will be examining differences between the two “sex” datasets. We followed the procedure as outlined in Section 3.3 for the Adult male (\mathcal{D}) and female datasets (\mathcal{D}'). We identified $N = 50$ influential examples in \mathcal{D}' and examined their characteristics – these correspond to only $\approx 1\%$ of the dataset. In order to use GOSDT (Lin et al., 2020) decision trees, we first binarised the age and education-num features by thresholding, using the method of McTavish et al. (2022).

For visualization purposes, Figure 5 shows non-binarised features and the respective influential examples.

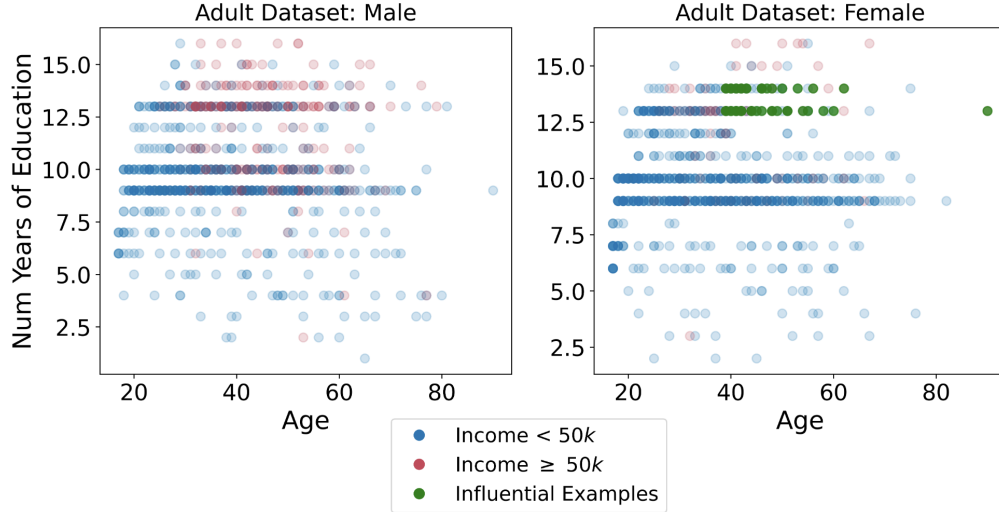


Figure 5: Visualization of the Adult male (\mathcal{D}) and female datasets (\mathcal{D}') with the influential examples for \mathcal{D}' overlaid. Our explanation aims to show that these examples are a big reason why the feature importances in the female dataset are different from the male dataset - hence, we only highlight influential examples in the female dataset and ‘fix’ the male dataset. These influential examples are seen to be localised to a specific part of the feature space. In particular, they are examples of young to middle-aged women with many years of education. We place this into context in the analysis below.

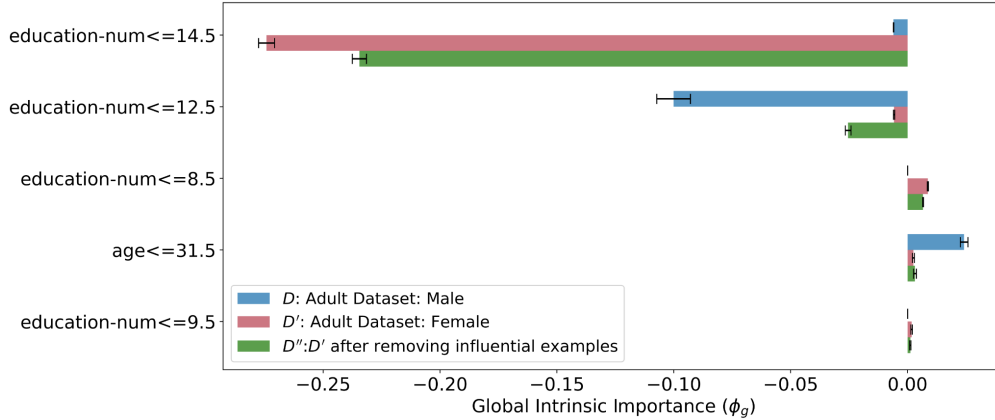


Figure 6: Global intrinsic feature importances for Adult males (\mathcal{D}), Adult females (\mathcal{D}'), and Adult females after removing 20 influential examples (i.e., \mathcal{D}''). The task is to predict whether the annual income is $\geq 50k$ from binarised age and education features. We show features whose importances in \mathcal{D}'' are most aligned. In particular, note how the blue and the green bars in the plot (corresponding to \mathcal{D} and \mathcal{D}'') are closer than the blue and red bars (resp. \mathcal{D} and \mathcal{D}'). Compared to men, women who have less than 14.5 years of education are disproportionately more likely to have lower income – this is the most affected feature. However, removing a small number of influential examples from the female dataset closes this gap – we discuss the implications of this below.

Dataset	Age	Num Education Years	# Income $\leq 50k$	# Income $\geq 50k$
Adult male: \mathcal{D}	39.86 ± 0.42	10.10 ± 0.08	709	291
Adult female: \mathcal{D}'	36.13 ± 0.44	9.91 ± 0.07	899	101
Influential Examples in \mathcal{D}'	46.12 ± 1.21	13.38 ± 0.07	38	12

Table 1: Mean value of the features (\pm standard error) **Age** and **Num Education Years** alongside the class balance of \mathcal{D} , \mathcal{D}' , and the influential examples. We see that the average influential examples all have similar characteristics – they include older women who are highly educated but are mostly not commanding a high income. Removing instances of these examples better aligns the intrinsic importances of features in the male and female datasets.

Given the above information in Table 1 and Figure 6, we can posit one dataset explanation: *Women who have less than 14.5 years of education are more likely to have lower income than men. This association is driven in large part due to a few highly educated (~ 13.4 years), middle-aged women, most of whom are not earning well.* Thus, analysing the properties of influential examples in datasets can not only uncover insights as to why \mathcal{D} and \mathcal{D}' differ in their intrinsic feature importances for the given task, but also highlight specific biases that may exist within the data.

3.3.5 CASE STUDY 2: HIGH DIMENSIONAL TABULAR DATA - HELOC DATASET

This dataset, which was used in the Explainable Machine Learning Challenge, contains information from the credit reports of around 12000 people. In particular, it contains features relating to trade characteristics (e.g., total trades, overdue trades, etc), consolidated risk indicators (external risk estimate, longest delinquency period, etc), and miscellaneous indicators (e.g., length of credit history). The task is to predict whether an applicant for a loan will repay it back within 2 years. Following Kulinski and Inouye (2023), we generate two separate datasets corresponding to low risk and high risk individuals. This is done by splitting the HELOC dataset on the variable `ExternalRiskEstimate`.

Dataset \mathcal{D} This is the low risk dataset. Concretely, $\mathcal{D} = \{(X, Y) | \text{ExternalRiskEstimate}(X) \leq 70\}$. `ExternalRiskEstimate` is a black-box metric computed by external agencies that estimates the risk of defaulting. We chose to split the data on this feature because it is likely that there is a distribution shift between individuals with high and low `ExternalRiskEstimate`.

Dataset \mathcal{D}' This is the high risk dataset. $\mathcal{D}' = \{(X, Y) | \text{ExternalRiskEstimate}(X) > 70\}$.

We now attend to influential example-based explanations for HELOC. We use the Rashomon Importance Distribution (RID) Donnelly et al. (2023) as the feature importance measure (see Section 3.3.1 for details). As with Section 3.4.4, we first binarized the features in \mathcal{D} and \mathcal{D}' using threshold guessing McTavish et al. (2022) as this is required as input to GOSDT and the RID framework. Let $\phi_g(\mathcal{D})$ and $\phi_g(\mathcal{D}')$ be the global intrinsic feature importance measures (GiFIM) for the datasets \mathcal{D} and \mathcal{D}' respectively (see Definition 4).

- We first identified the $N = 50$ *influential* examples in \mathcal{D}' using Algorithm 1. Because \mathcal{D}' is of size ≈ 4500 , these influential examples correspond to only $\approx 1\%$ of the dataset. Figure 7 shows these examples highlighted in the original dataset.
- We then removed these examples from the dataset \mathcal{D}' . Call this new dataset \mathcal{D}'' .
- Lastly, we recomputed the LiFIMs and GiFIMs on \mathcal{D}'' .

We now show the resulting feature importances in Figure 8. We then look at the features whose importances were most affected by this removal.

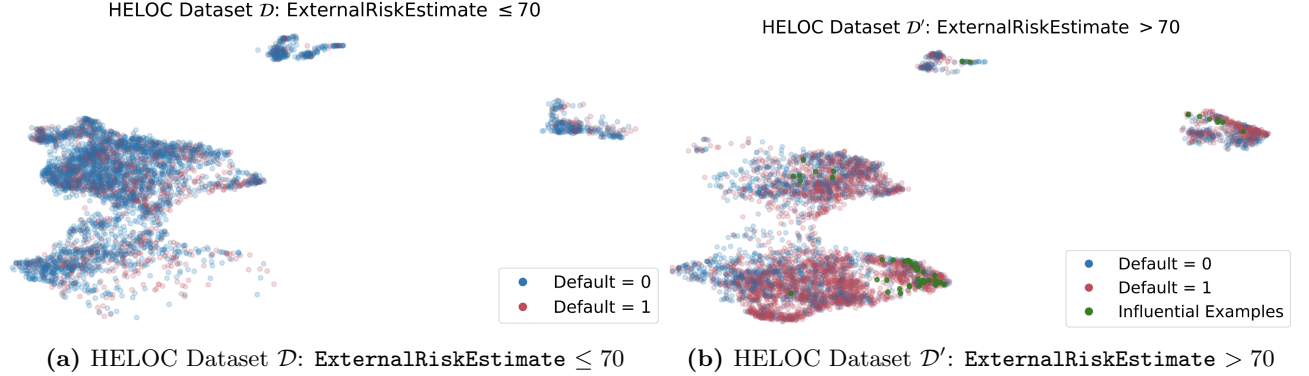


Figure 7: PaCMAP (Wang et al., 2021b) projection of HELOC datasets \mathcal{D} and \mathcal{D}' in a common 2-D space, but with the influential examples for \mathcal{D}' overlaid. The most influential examples are seen to be localised to a specific part of the feature space. From Section 3.3.1, these are the examples that, if removed from \mathcal{D}' , would most likely align the feature importances of \mathcal{D} and \mathcal{D}' . We examine this further below.

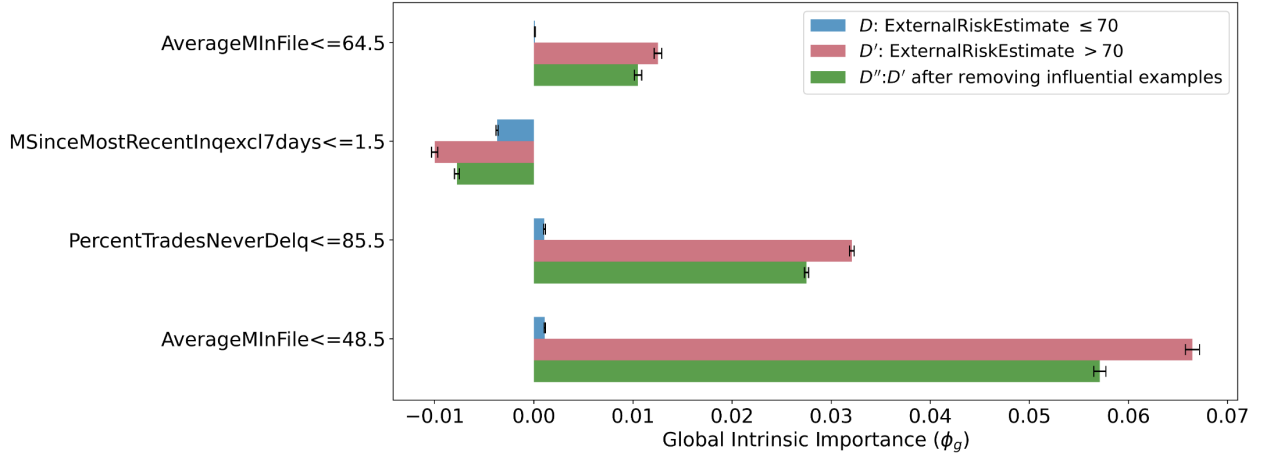


Figure 8: Global intrinsic feature importances for datasets \mathcal{D} , \mathcal{D}' , and \mathcal{D}' after removing influential examples (i.e., \mathcal{D}''). **AverageMinFile** = Length of credit history. **MSinceMostRecentInq** = Months since most recent credit inquiry. **% TradesNeverDelq** = % of non-delinquent trades. We show binarised features (e.g. **AverageMinFile** ≤ 64.5) that have the greatest change in feature importance between \mathcal{D}'' and \mathcal{D}' . In particular, note how the blue and the green bars in the plot (corresponding to \mathcal{D} and \mathcal{D}'') are closer than the blue and red bars (resp. \mathcal{D} and \mathcal{D}'). We examine the properties of examples removed from \mathcal{D}' to see why this is the case (see Table 2.)

Dataset	AverageMinFile	MSinceMostRecentInq	%TradesNeverDelq	# Default = 0	# Default = 1
\mathcal{D}	67.00 \pm 0.47	0.1 \pm 0.07	91.01 \pm 0.21	1746	3566
\mathcal{D}'	86.76 \pm 0.46	0.70 \pm 0.09	97.10 \pm 0.08	3390	1169
Influential Examples in \mathcal{D}'	106.92 \pm 4.99	NaN	99.16 \pm 0.28	6	44

Table 2: Average value \pm standard error of some original (non-binarised) important features and number of examples of each class (Default = 0 and Default = 1) in \mathcal{D} , \mathcal{D}' , and the influential examples. We see that the influential examples correspond to individuals with high **AverageMinFile** and **%TradesNeverDelq** and no known recent inquiry (**MSinceMostRecentInq** is NaN – these are given a special value of -8 in the dataset). This corresponds to individuals with longer credit histories who have almost no delinquent trades and no credit inquiries on their profile. Despite these positive indications, most of these individuals have defaulted on their loans in the last 2 years (44 out of 50 samples with Default = 1).

We can now compare the two datasets by considering the properties of influential examples in Table 2 and the GiFiMs of important features in Figure 8. The dataset difference explanation therefore tells us the following: *The binary features **TradesNeverDelq** ≤ 85.5 , **AverageMinFile** ≤ 48.5 , **AverageMinFile** ≤ 64.5 , and **MSinceMostRecentInq** ≤ 1.5 are considered to be unusually important in the higher risk dataset \mathcal{D}' compared to \mathcal{D} . However, this is in large part due to a few individuals in \mathcal{D}' who mostly defaulted on their loan in the last 2 years despite having $\approx 99\%$ non-delinquent trades, longer credit history, and no recent credit inquiries.*

3.4 Prototype-Neighbourhood-Based Explanations

3.4.1 INTRODUCTION

Given two datasets \mathcal{D} and \mathcal{D}' , prototype-based explanations compare these datasets using a set of prototypical samples $P = \{p_1, p_2, p_3, \dots, p_n\}$. Each of these prototypes is considered to be a meaningful and faithful representation of its neighboring samples when \mathcal{D} and \mathcal{D}' are projected to a latent space. By comparing the neighborhood sample distribution between \mathcal{D} and \mathcal{D}' , we could provide insights into the differences between two datasets. There are multiple ways to create prototypes:

- First, we can choose prototypes manually with domain knowledge. We show an example of this for explaining the difference between males and females in the Adult dataset in Section 3.4.4, i.e., in Figures 11 and 12.

- Second, cluster centers from clustering methods such as k -means as the cluster centers (similar to Kulinski and Inouye, 2023) can be seen as prototypes. This is illustrated in Section 3.4.5 to explain the difference between low and high risk examples in the HELOC dataset, i.e., in Figures 13 and 14.
- Third, prototypes and their surrounding latent space can be learned in a neural network in a supervised and end-to-end fashion, where the encoder f , prototype set P , and the final classifier layer are the learnable-components. We use this approach for explaining the differences between real and synthetic PPG data and human and machine generated audio. For this last approach, we adapt ProtoPNet (Chen et al., 2019b) and its variant (Barnett et al., 2023) to project both \mathcal{D} and \mathcal{D}' into the same latent space of the learned encoder, as illustrated in Figure 9b. ProtoPNet tends to have similar accuracy to its non-interpretable counterparts despite being trained to use case-based reasoning, thus providing assurance of the quality of the learned latent space from a performance perspective. In this latent space, we make quantitative comparisons between the learned prototypes P and their neighborhoods in \mathcal{D} and \mathcal{D}' .

3.4.2 QUANTITATIVE COMPARISON BETWEEN NEIGHBORHOODS

Once the prototypes corresponding to \mathcal{D} are generated, we can use two metrics to analyze the differences between \mathcal{D} and \mathcal{D}' :

Definition 6 (Neighboring Sample Proportion Difference – NSPD) *The neighboring samples for prototype p_i are defined as the samples that have p_i as their closest prototype. The neighboring sample distribution difference for p_i is calculated as the difference between the percentage of p_i ’s neighboring samples in \mathcal{D} and the percentage of p_i ’s neighboring samples in \mathcal{D}' .*

Definition 7 (Neighboring Sample Distance Difference – NSDD) *The neighboring sample distance difference for p_i is calculated as the difference between the average neighboring sample distance to p_i in \mathcal{D} and the average neighboring sample distance to p_i in \mathcal{D}' . The distance between a sample’s feature and a prototype in the latent space is calculated using cosine distance.*

We can compute these differences either in the original feature space or project the prototypes to a latent space using a learned encoder. Figures 9a and 9b illustrate the process of obtaining prototypes in latent space. In Section B.1 in the appendix, we examine how adjusting the number of prototypes influences the balance between the explanation’s complexity and its faithfulness. We also discuss in Section 4.1 practical justifications for the number of prototypes in the explanations, as it is an important design choice.

We show an example of two toy examples with high NSPD and low NSDD in Figure 9g, another pair of toy examples with high NSDD but low NSPD in Figure 9h, and a pair of toy examples with both low NSDD and low NSPD in Figure 9i to illustrate Def. 6 and Def. 7 in practice. In addition to quantitative comparisons, users can also inspect each prototype and perform visual comparisons with samples in \mathcal{D} and \mathcal{D}' . We show examples of this throughout the paper.

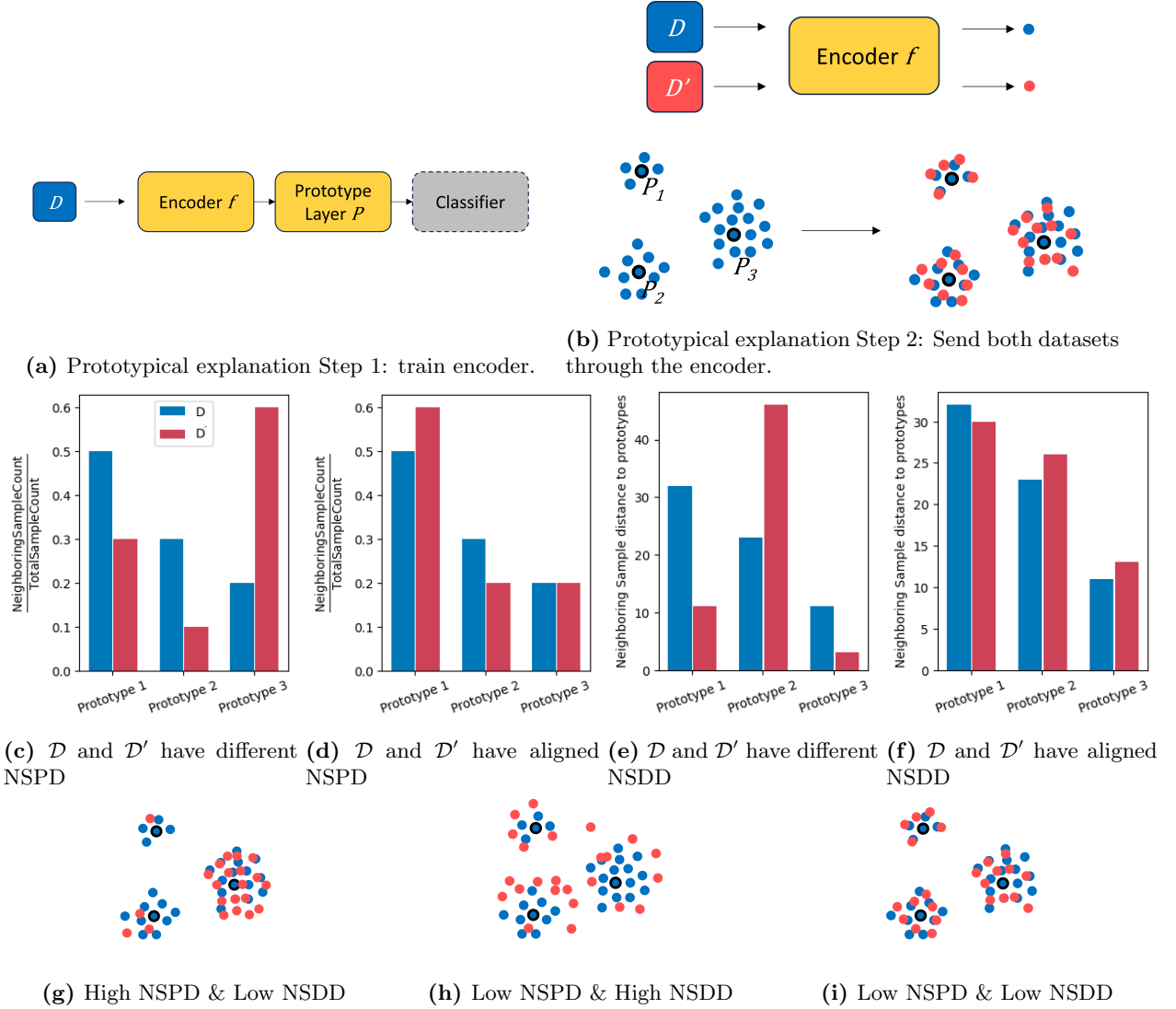


Figure 9: In Step 1, we follow the approach of Barnett et al. (2023) to learn both an encoder and a set of faithful prototypes. In Step 2, both \mathcal{D} and \mathcal{D}' are encoded by the learned encoder in Step 1. The encoded features of samples in \mathcal{D}' are projected into the sample latent space of f , the encoder learned on \mathcal{D} . The samples in \mathcal{D}' are compared against the prototypes P in the same latent space, users can see which parts of \mathcal{D}' are not evenly distributed in the latent space. In (c), (d), (e), (f), we show four examples of P-1 and P-2 evaluation results. The NSPD and NSDD are based on Definitions 6 and 7 respectively.

3.4.3 COMPARING PROTOTYPE NEIGHBORHOODS IN \mathcal{D} AND \mathcal{D}' IN HIGH DIMENSIONS

For tabular datasets with a large number of features, it is useful to use only a subset of relevant variables within the NSDD and NSPD calculation. As we will see, using a good subset will allow a high-quality approximation of the full NSDD and NSPD (see Section B.2 in the appendix), with a much sparser feature set. Our partial prototypical explanation provides the NSDD and the NSPD for the prototype along with the K most relevant features of the prototype for the user to focus on. The notion of a relevant feature is based on two desiderata: *value stability* and *rank stability*.

Definition 8 (Value Stability) *The K chosen features must vary less around the prototype neighborhood, i.e., given a prototype X_p from dataset \mathcal{D} and K chosen feature indices $\{m_1, \dots, m_k\}$, we want to ensure $\mathbb{E}_{X' \in \mathcal{D}' | d(X', X_p) \leq \delta} \left[d(X_p[m_1, \dots, m_k], X'[m_1, \dots, m_k]) \right]$ is small, where \mathcal{D} is a distance metric that can compare vectors of the same dimension (e.g., ℓ_2 , ℓ_1 , or distances that use inner products).*

If the dataset is labeled, we can optionally one-hot encode the labels and append them to the example vector before computing distances. While we chose examples in \mathcal{D}' that are in an overall δ neighborhood of the prototype, there may be some features whose values in the neighborhood vary less than others. Thus, if we choose only K features due to interpretability constraints, we are best off choosing important features whose values are most stable in the neighborhood. One important clarification: for the NSDDs and NSPDs of partial prototypes to remain approximately similar to the original prototypes, we want to preserve the structure of the prototype neighborhood as much as possible. Selecting more features will preserve neighborhood structure better but will lead to a loss in interpretability – this tradeoff is illustrated in the appendix (Section B).

Definition 9 (Rank Stability) *If the datasets are labeled, the K features selected should capture as much of the true model behavior as possible, i.e., they should be important for the **prototype in \mathcal{D}** and similarly important for its **neighbors in both \mathcal{D} and \mathcal{D}'** .*

This helps the end user reason about neighboring sample distribution and distance differences only in terms of features that are equally important for both datasets. To this end, we generate the LiFIM $\phi(X, Y, \mathcal{D})$ using the Rashomon Importance Distribution (RID) method that can return a vector containing intrinsic feature importance scores for each feature in $(X, Y) \in \mathcal{D}$. The equivalent LiFIM for \mathcal{D}' is $\phi(X, Y, \mathcal{D}')$. We further break down rank stability into two components below: To enable this, we propose a score for each feature, and we will use the top scoring features within the partial prototype explanation:

- **Definition 10 (Rank Difference Penalty)** *If feature j is deemed to be very important for the prototype $(X_p, Y_p) \in \mathcal{D}$ (according to the local intrinsic importance $\phi(X, Y, \mathcal{D})$), but this feature is not so important for prototype neighbors in either \mathcal{D} or \mathcal{D}' , it is assigned a high penalty score – this feature is less likely to be one of the K selected. This penalty therefore penalizes the relative rank differences in the importance of feature j in predicting the label for a prototype in \mathcal{D} and its neighbors in \mathcal{D}' and \mathcal{D} .*
- **Definition 11 (Absolute Rank Penalty)** *The above mechanism could result in features which are less important for both the prototype and its neighborhood being selected (as only the relative rank difference is penalized). However, the chosen features should be important for both the prototype and the neighborhood. The absolute rank penalty aims to ensure that a chosen feature that has low rank difference penalty is also an important feature.*

As these forces can be opposing, we propose a score function for each feature that is based on a user-defined tradeoff between rank stability and value stability. Given feature j , datasets \mathcal{D} and \mathcal{D}' , an example $(X', Y') \in \mathcal{D}'$, and LiFIM $\phi(X', Y', \mathcal{D}')$, let $U_j^\phi(X', Y', \mathcal{D}') = \text{rank}(\phi(X', Y', \mathcal{D}') [j])$ be the rank of the importance of the feature (i.e., if j is the 3^{rd} most important feature, then $U_j^\phi(X', Y', \mathcal{D}') = 3$). Then, the scoring function for feature j given example $(X', Y') \in \mathcal{D}'$ and prototype $(X_p, Y_p) \in \mathcal{D}$ is:

$$\begin{aligned}
 s_j(\mathcal{D}, \mathcal{D}', X_p, Y_p, X', Y') = & c_1 \underbrace{\left(\left| U_j^\phi(X_p, Y_p, \mathcal{D}) - U_j^\phi(X', Y', \mathcal{D}') \right| \right)}_{\text{Rank Difference Penalty}} + c_2 \underbrace{\left(0.5 U_j^\phi(X_p, Y_p, \mathcal{D}) + 0.5 U_j^\phi(X', Y', \mathcal{D}') \right)}_{\text{Absolute Rank Penalty}} \\
 & \underbrace{\hspace{10em}}_{\text{Rank Stability}} \\
 & + c_3 \underbrace{\left| X_p[j] - X'[j] \right|}_{\text{Value Stability}}
 \end{aligned} \tag{7}$$

The same scoring function can be defined for an example $(X, Y) \in \mathcal{D}$. Algorithm 2 then sums up scores across both datasets for each feature and prototype.

where the user can choose parameters c_1 , c_2 , and c_3 to weigh the relative importance of each desideratum. This naturally induces a tradeoff between value stability and rank stability, which is illustrated in Figure 30 in the appendix.

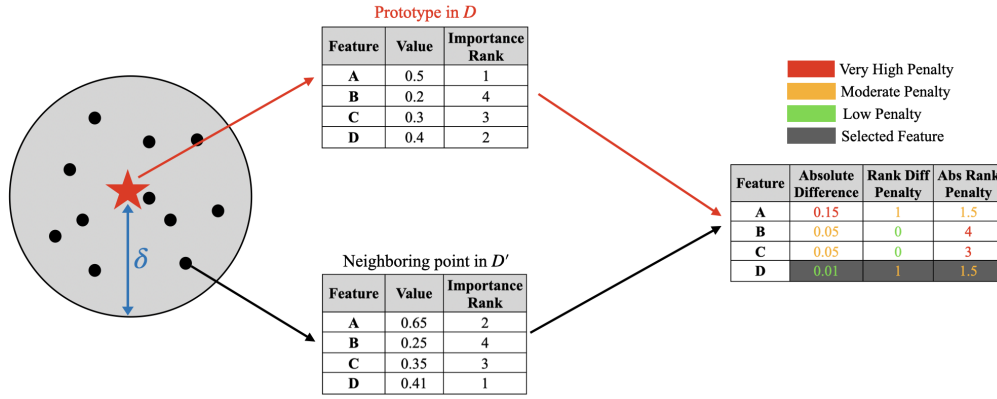


Figure 10: Simple example illustrating our interpretable partial prototype feature scoring procedure for $K = 1$ (i.e. choosing the best feature). Say we only consider two points to compute the feature scoring function – the prototype in \mathcal{D} and a point in \mathcal{D}' in the δ neighborhood of the prototype. We now compare the feature values and the feature importance values of each feature for both points: Feature A ’s value differs a lot between the points compared to other features, and it is relatively important for predicting labels for both points. Features B and C are less important for both the prototype and the variable, and they do not differ as much between the two feature tables. Feature D is very stable in value, is relatively important for prediction, and has only a moderate difference in rank between the prototype and the neighbor. Our scoring procedure therefore chooses feature D as the partial prototype because it is reliably important for both \mathcal{D} and \mathcal{D}' .

Algorithm 2 Partial Prototype-Based Explanations

Require: M , K , c_1 , c_2 , c_3 , δ , $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$, $\mathcal{D}' = \{(X'_i, Y'_i)\}_{i=1}^{N'}$, Prototype Learning Algorithm P , Feature Importance Function $\phi: \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{R}^{|\mathcal{X}|}$ based on RID Donnelly et al. (2023)

- 1: Determine the M most salient prototypes in \mathcal{D} using the prototype learning algorithm P
- 2: **for** Prototype: $Z_p = (X_p, Y_p)$ in the set of M learned prototypes **do**
- 3: $\mathcal{D}_\delta \rightarrow \{Z = (X, Y) \in \mathcal{D} | d(Z, Z_p) \leq \delta\}$ ▷ Examples in \mathcal{D} close to prototype Z_p
- 4: $\mathcal{D}'_\delta \rightarrow \{Z' = (X', Y') \in \mathcal{D}' | d(Z', Z_p) \leq \delta\}$ ▷ Examples in \mathcal{D}' close to prototype Z_p
- 5: $S \rightarrow \emptyset$
- 6: **for** Feature j in set of features **do**
- 7: $s_{\mathcal{D}} \rightarrow \mathbb{E}_{(X,Y) \in \mathcal{D}_\delta} [s_j(\mathcal{D}, \mathcal{D}, X_p, Y_p, X, Y)]$ ▷ Equation 7 for Dataset \mathcal{D}_δ - the average score for the neighbors in \mathcal{D}
- 8: $s_{\mathcal{D}'} \rightarrow \mathbb{E}_{(X',Y') \in \mathcal{D}'_\delta} [s_j(\mathcal{D}, \mathcal{D}', X_p, Y_p, X', Y')]$ ▷ Equation 7 for Dataset \mathcal{D}'_δ - the average score for the neighbors in \mathcal{D}'
- 9: $s_{total} = s_{\mathcal{D}} + s_{\mathcal{D}'}$
- 10: Append score s_{total} to S
- 11: Choose the array indices $[m_1, \dots, m_K]$ in S with the K lowest scores. These are the K chosen features.
- 12: $X_p^{partial} \rightarrow X_p[m_1, \dots, m_K]$.
- 13: **return** M partial prototypes, each with K features

We note that having a feature importance function is not strictly necessary for the scoring mechanism and may only be used if the dataset is labelled. Otherwise, one can simply set the parameters c_1 and c_2 to 0 and work only with the value stability desiderata. In Section 3.4.5 of this paper, we will demonstrate examples of partial prototypes for a few real-world tabular datasets. In the appendix (Section B.2), we also share recommendations for choosing an appropriate value of K . In particular, a large value of K will provide the user with a larger prototype vector, making it less interpretable but more expressive. However, a very small value of K may not necessarily preserve the NSDDs and NSPDs, degrading the quality of the explanation.

3.4.4 CASE STUDY 1: LOW DIMENSIONAL TABULAR DATA - ADULT DATASET

In this section, we construct prototypical explanations for the Adult dataset, employing the NSPD and NSDD methods. The setup is the same as in Section 3.3.4 - we are comparing Adult male and female datasets. This example is only three-dimensional (so we do not require complex dimension reduction), and prototypes will be chosen in a simple heuristic manner based on feature percentiles and depth-2 decision trees. To construct prototypes, we first defined 3 categories of education levels: lower, medium, and high. These correspond to the 10th, 50th, and 90th percentiles of education years in the male dataset \mathcal{D} . Note that we could have also chosen the female dataset for constructing prototypes – there is nothing inherently special about our choice here. We categorised age in the same manner as education. 9 prototypes were then constructed, corresponding to all possible combinations of education level and age. To construct an income feature, we trained a shallow decision tree classifier on \mathcal{D} to predict if income $\geq \$50k$ from age and education level. Each prototype was then passed through this decision tree and the tree’s prediction (the majority vote in the leaf) was used as the income feature for the prototype.

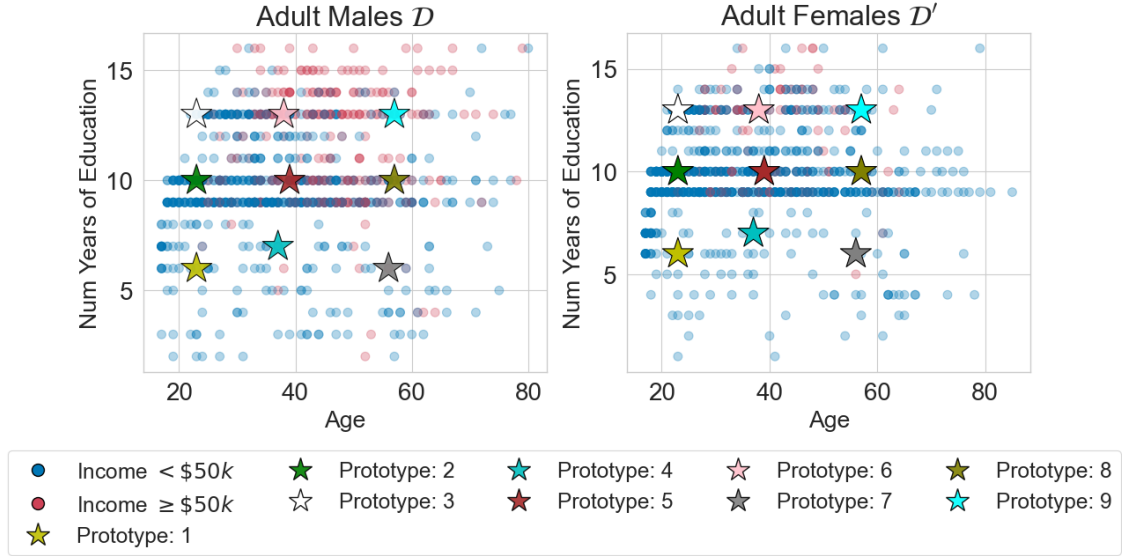


Figure 11: Visualization of the Adult datasets for males (\mathcal{D}) and females (\mathcal{D}') in 2-D space, where only two features – Age and Num Years of Education – are considered. The colors of each point correspond to its class label.

	Feature 1	Feature 2	Feature 3
Prototype 2	Age 23	# Education Years 10	Income $\geq \$50k$ 0
Prototype 4	Age 38	# Education Years 7	Income $\geq \$50k$ 0
Prototype 6	Age 40	# Education Years 13	Income $\geq \$50k$ 1
Prototype 8	Age 59	# Education Years 7	Income $\geq \$50k$ 0

Table 3: A few prototypes from the Adult male dataset. The NSPD and NSDD for both datasets are computed using Euclidean distance metric over the normalized version of the datasets and the prototypes. We perform normalization of both the prototype and the datasets by using the average and standard deviation of Age and #Education Years from the Male dataset. The binary income feature is not normalized.

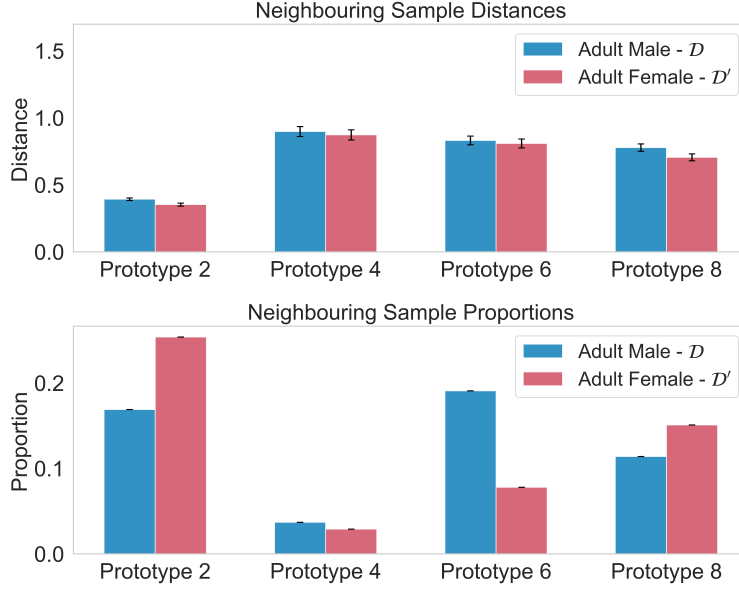


Figure 12: NSPD and NSDD for Adult male (\mathcal{D}) and female (\mathcal{D}') datasets. Both datasets have similar average distance to the prototypes, but the proportions of examples in the neighborhood of a given prototype are very different. For instance, there are a disproportionately high number of male examples in the neighbourhood of Prototype 6.

A visualization of the Adult male and female datasets is seen in Figure 11. We can now interpret the NSPD and NSDD for the datasets in terms of these prototypes. To facilitate comparison with Kulinski and Inouye (2023), consider the prototype corresponding to middle aged individuals with a bachelor’s degree who earn more than \$50k (i.e., education = 13, age = 38, income = 1). This is marked as Prototype 6 in Table 3. Figure 12 shows that there are comparatively fewer examples of this archetype in the female dataset than in the male dataset. An explanation is therefore: *Compared to the male dataset, the female dataset contains fewer individuals who have a bachelors degree, are middle aged, and earn a high income.* Similar comparisons can be made for other prototypes.

3.4.5 CASE STUDY 2: HIGH-DIMENSIONAL TABULAR DATA - HELOC DATASET

We now construct prototypical explanations for the HELOC dataset. Our datasets \mathcal{D} and \mathcal{D}' are the same as in Section 3.3.5. Keeping \mathcal{D} as the reference dataset, we define the prototypes to be the cluster centers in \mathcal{D} obtained after K-means clustering on the high dimensional space and projecting them to a lower dimensional space using PaCMAP (Wang et al., 2021b).

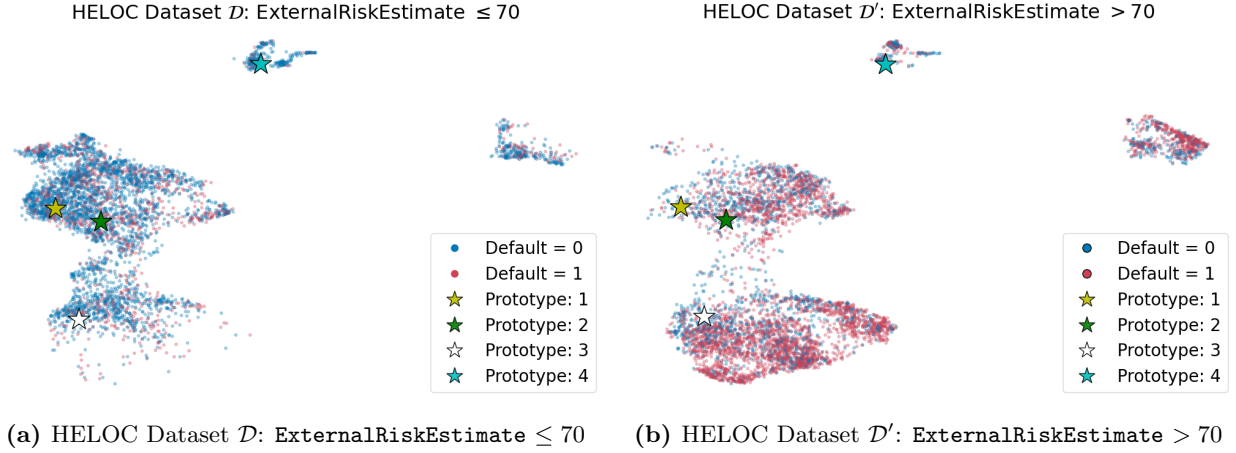


Figure 13: PaCMAP projection of HELOC datasets \mathcal{D} and \mathcal{D}' in a common 2-D space. The color of each point corresponds to its class label. The same prototypes that were learned on \mathcal{D} (left) are being visualized on \mathcal{D}' (right). Both datasets are normalized using the mean and standard deviation of features from \mathcal{D} .

To generate these PaCMAP projections, we combined \mathcal{D} and \mathcal{D}' , ran PaCMAP on this combined dataset, and plotted the lower dimensional datasets separately. The PaCMAP visualizations serve as explanations on their own; because PaCMAP preserves the global and local structure of datasets (Wang et al., 2021b), visualizing them on a common projected space enables us to understand the cluster structure and relative shifts qualitatively. Even a bird’s eye view of the datasets using PaCMAP provides us with very useful information. First, both datasets have similar structures in the feature space, implying that their features are likely to take on the same range of values. Another indication is the larger presence of people who defaulted on their loan in the higher risk dataset \mathcal{D}' (i.e. class 1 labels).

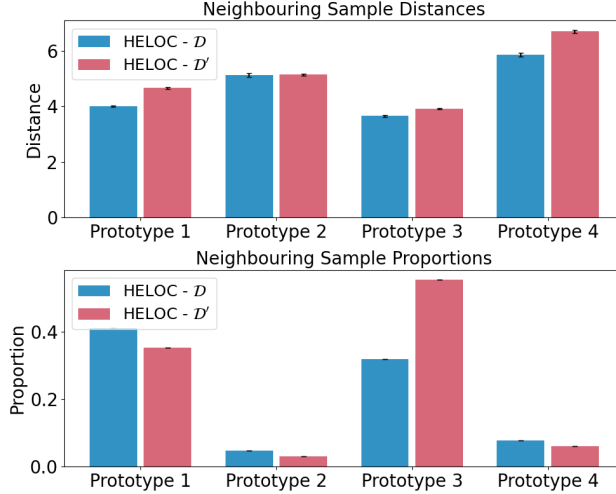


Figure 14: NSPD and NSDD for the HELOC datasets \mathcal{D} and \mathcal{D}' . The distance metric used is the Euclidean distance in high dimensional space. Because PaCMAP is structure-preserving (Wang et al., 2021b), the distance metrics in low and high-dimensional space will be very similar. \mathcal{D} contains fewer examples that are close to Prototype 3 compared to \mathcal{D}' , but the average distance to the prototype is similar. Similar types of conclusions can be made for other prototypes. This analysis enables the user to focus on certain neighbourhoods where \mathcal{D} and \mathcal{D}' are most different.

	Feature 1	Feature 2	Feature 3	Feature 4
P1	NumRevolvingTradesWBalance 3	PercentInstallTrades 0	NumInstallTradesWBalance 1	NumInqLast6M 1
P2	MSinceOldestTradeOpen 1	NumTrades60Ever2DerogPubRec 1	NumTrades90Ever2DerogPubRec 97	NumRevolvingTradesWBalance 5
P3	NumSatisfactoryTrades 1	NumTrades60Ever2DerogPubRec 1	MSinceOldestTradeOpen 7	PercentInstallTrades 0
P4	NumTotalTrades 0	NumInqLast6M 2	MSinceMostRecentInqexcl7days 2	MaxDelqEver 18

Table 4: Understanding the $K = 4$ most salient features for each prototype in \mathcal{D} . We can interpret this jointly with Figures 14 and 13. Here is a dataset-level explanation in terms of Prototype 3: the dataset of individuals with lower **ExternalRiskEstimate** (i.e., \mathcal{D}) has a lower proportion of individuals with approximately the following profile:

- Num Satisfactory Trades = 1
- 1 trade more than 60 days past due
- 7 Months since last trade
- No installment trades

Similar interpretations can be made for other prototypes.

Given the prototypes in \mathcal{D} , the explanation compares the NSPD and NSDD of \mathcal{D} and \mathcal{D}' for these prototypes. From Figure 14, we can also analyze a small subset of salient features for a prototype (aka the partial prototype) to understand the properties of the prototype and its neighbourhood in \mathcal{D} and \mathcal{D}' in an interpretable manner.

3.4.6 CASE STUDY 3: TIME SERIES MEDICAL DATA - CARDIAC SIGNALS

Cardiac signals are essential in clinical diagnostics and disease screening. The advancement of machine learning and deep learning has facilitated numerous studies to automate cardiac disease detection, further improving reliability and efficiency. However, the scarcity of large open-access datasets poses a challenge for practitioners and machine learning researchers. Given this context, the need for accurate and high-quality synthetic cardiac data becomes imperative. In this experiment, we aim to showcase our method by comparing synthetic data against real-world data and derive actionable items to improve synthetic data generation.

- **Dataset \mathcal{D} :** Photoplethysmography (PPG) was chosen as a representative form for time series medical signals due to its rising popularity in recent years as the medium for heart monitoring on wearable devices. In this study, we chose the Stanford PPG dataset, which was collected from subjects wearing smartwatches while performing regular daily activities Torres-Soto and Ashley (2020). Using the dataset’s signal quality labels, we sampled a subset of 16,058 25-second signals that contain a relatively small amount of noise, each accompanied by an atrial fibrillation (AF) or non-atrial fibrillation (non-AF) label. During preprocessing, signal amplitudes were normalized into the 0-1 range and re-sampled to have 2400 timesteps in the 25 seconds time frame. We show samples of real PPG signals in Figure 38a.
- **Dataset \mathcal{D}' :** A popular PPG processing and simulation tool, neurkit2 (Makowski et al., 2021), was used to generate a synthetic PPG dataset containing 3,000 30-second signals for this study. Synthetic signals were created with the addition of varying levels of signal noise and artifacts to mimic realistic conditions. A detailed description of the simulation parameters can be found in appendix Section A; we also show a few generated synthetic PPG signals in Figure 38a. Signal amplitudes were normalized to the 0-1 range and resampled to have 2400 timesteps.

Forming the explanation The comparisons were conducted using the prototypical explanation method introduced in Section 3.4.2. A 1D-ResNet-34 model is used as the encoder. To accommodate the relatively small D dataset size, we first pre-trained the encoder using a multitask approach. The encoder was trained to optimize both a signal reconstruction MSE loss as part of an autoencoder, and the cross-entropy loss for the AF detection classification task (AF vs. non-AF classification). The pre-trained encoder was then used to train the prototype learning model following the approach in previous work (Barnett et al., 2023).

Quantitative comparison between prototypical neighborhoods We are able to visualize the projection of encoded samples in both \mathcal{D} and \mathcal{D}' . We can observe the difference in coverage of \mathcal{D}' samples to the \mathcal{D} samples in the latent space. The learned prototypical samples of \mathcal{D} are shown in Figure 15. We calculate the quantitative difference using the NSPD and NSDD metrics defined in Section 3.4.2, and the results are shown in Figure 16. From these results, we conclude that the synthetic data generator does generate samples similar to those of the real dataset \mathcal{D} in terms of latent space distance; however, there is a discrepancy between the number of certain types of signals generated in the synthetic dataset and in the real dataset. This conclusion is supported by the fact that the NSDD is relatively small, *indicating a similarity in features related to AF classification between generated signals and the prototypes comparable to that between the real samples and the prototypes* (shown in Figure 16); in addition, we observe large NSPDs for prototypes 1, 2 and 4, indicating that there are *insufficient samples similar to prototype 1 and 4*, and *too many samples similar to prototype 2*. By inspecting the learned prototypes, we could potentially improve the realism and quality of the generated signals by introducing more variable and organic noise corruptions similar to those in prototypes 1 and 4, in addition to those in Neurokit 2 (Makowski et al. (2021)).

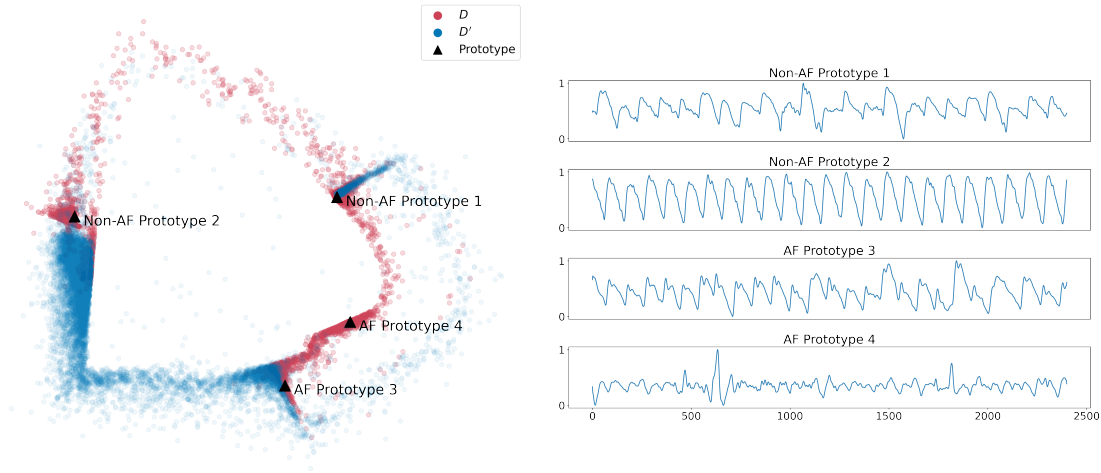


Figure 15: Visualization of the projections of encoded samples in both \mathcal{D} and \mathcal{D}' in the same latent space, including the learned prototypes.

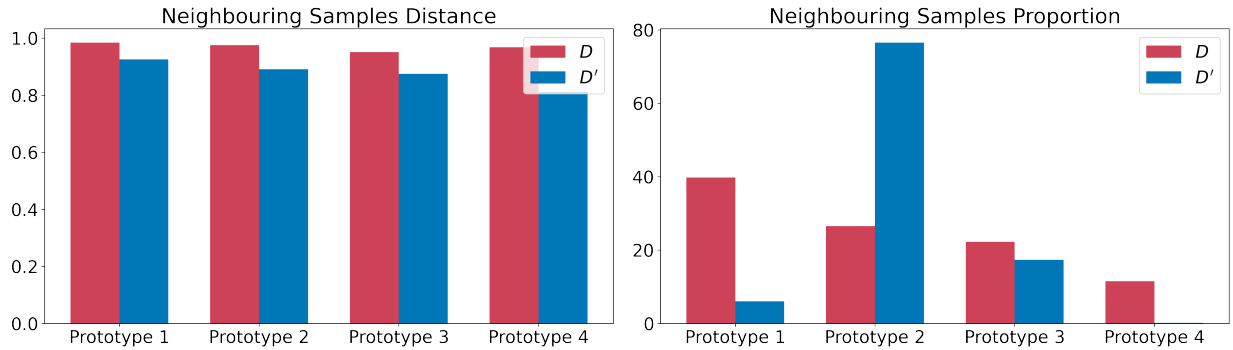


Figure 16: NSPD and NSDD comparison between \mathcal{D} , \mathcal{D}' sample features for all learned prototypes. The two datasets differ in that there are very few samples in \mathcal{D}' that are sufficiently similar to Prototypes 1 and 4 in \mathcal{D} ; More samples that are similar to Prototype 2 than that in \mathcal{D} .

3.5 Prototype-Summarization-Based Explanations

3.5.1 INTRODUCTION

In this section, we propose another type of prototypical explanation that is different from the neighbourhood-based-explanation method introduced in Section 3.4. Previously, we learned prototypes in *only one* of the two datasets. Here, we learn prototypes in *both* \mathcal{D} and \mathcal{D}' using a modified version of ProtoPNet (Chen et al., 2019a). The ensuing explanation involves directly comparing prototypes unique to \mathcal{D} and \mathcal{D}' , providing insights into the differences between the datasets. As we will show, this approach is especially useful for visual and signal-based datasets where differences are not easily discernible through direct inspection. These prototypes represent key samples from \mathcal{D} and \mathcal{D}' that best explain the distinctions between the datasets.

To identify these prototypes, we construct a binary classification task where samples from \mathcal{D} are labeled as 1 and samples from \mathcal{D}' as 0. A ProtoPNet or a similar prototype learning network is then trained to distinguish between the two datasets, learning n_p prototypes for \mathcal{D} and $n_{p'}$ prototypes for \mathcal{D}' . These prototypes encapsulate the unique and distinguishing characteristics of each dataset, enabling users to analyze differences without examining a large number of samples. Figure 17 illustrates this process. In subsequent sections, we provide examples demonstrating how prototype summarization explanations can effectively highlight distribution shifts using a small set of representative samples.

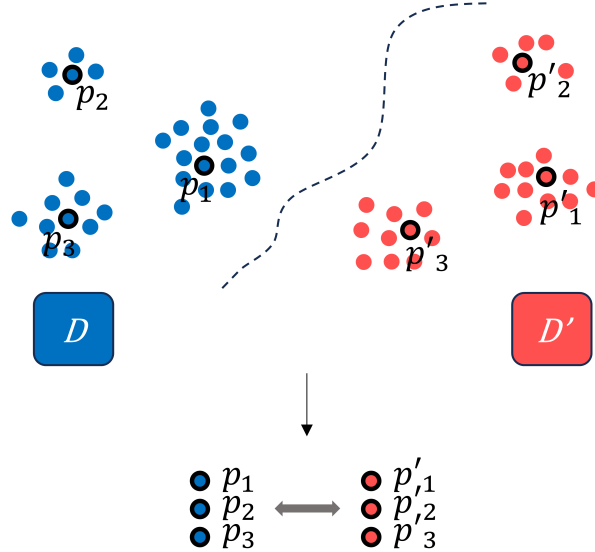


Figure 17: An illustration of the Prototype-Summarization-Based Explanations. First a prototype learning model is trained to classify between dataset \mathcal{D} and \mathcal{D}' . The learned prototypes p_1, p_2, p_3 from dataset \mathcal{D} can be used as a summarization of its neighbouring samples and to be compared against prototypes p'_1, p'_2, p'_3 learned in dataset \mathcal{D}' , thus forming an explanation.

3.5.2 SUMMARIZATION PROTOTYPE LEARNING FOR DATASET COMPARISONS

In general, the feature extractor in ProtoPNet is trained by optimizing the following loss terms:

$$\min_{\omega_g} (\text{Cross Entropy} + \lambda_c \ell_{\text{clst}} + \lambda_s \ell_{\text{sep}}), \text{ where} \quad (8)$$

$$\ell_{\text{clst}} = \frac{1}{n} \sum_{i=1}^n \min_{j: \text{class}(\mathbf{p}_j) = y_i} s(f_{\omega_f}(\mathbf{x}_i), p_j), \quad \ell_{\text{sep}} = -\frac{1}{n} \sum_{i=1}^n \min_{j: \text{class}(\mathbf{p}_j) \neq y_i} s(f_{\omega_f}(\mathbf{x}_i), p_j). \quad (9)$$

Here, ℓ_{clst} ensures that each training sample is close to a prototype of its class, while ℓ_{sep} pushes samples away from prototypes of other classes. Additionally, ℓ_{ortho} , an optional term, encourages prototype diversity.

The task of comparing the differences between datasets, which requires the learning of summarization prototypes, differs significantly from the existing literature’s prototype learning task – classification. Instead of associating samples with object classes, samples are labeled based on dataset membership. Although

existing loss functions could work for datasets with few or single-semantic concepts (where the object label \approx dataset membership label), they will falter for datasets containing diverse concepts (concepts could be objects, subcategories, art styles, etc.) In such cases, the model would reduce the loss trivially by aligning all samples regardless of their difference in concepts or subcategories within a dataset to the same prototypes, compromising the summarization power.

To address this, we propose an alternative clustering loss:

$$\ell_{\text{clst}} = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{p_k \in P} [s(f_{\omega_f}(\mathbf{x}_i), p_k)] - \min_{j: \text{class}(\mathbf{p}_j) = y_i} s(f_{\omega_f}(\mathbf{x}_i), p_j). \quad (10)$$

This modified loss function encourages a greater separation between the most similar prototype and the average similarity to all prototypes within the same dataset. By doing so, it ensures that each sample strongly aligns with a single prototype while avoiding trivial alignment with others. This adjustment improves prototype summarization for datasets with diverse underlying concepts.

Additionally, to further improve the summarization power of the prototypes and coherence in the latent space, we also introduce a novel prototype affinity-based contrastive learning loss to aid the learning of the model and prototypes without explicit supervision. This learning target does not utilize any object or concept labels for contents within the dataset.

The contrastive prototype loss is designed to align similar samples while distinguishing dissimilar samples. We construct two augmented views from each input image, calculate the similarity matrices from each view's latent feature to each prototype $P, Q \in \mathbb{R}^{n \times m}$, where n is the number of samples and m is the number of prototypes. The similarities are then normalized into an prototype affinity distribution using softmax function with a temperature scaling factor T (we default to $T = 0.07$):

$$p_{ip} = \frac{\exp(P_{ip}/T)}{\sum_{k=1}^m \exp(P_{ik}/T)}, \quad q_{ip} = \frac{\exp(Q_{ip}/T)}{\sum_{k=1}^m \exp(Q_{ik}/T)}, \quad (11)$$

where p_{ip} and q_{ip} represent the affinity of sample i to prototype p in the two respective views, and k indexes over the m prototypes.

We measure how aligned two prototype affinity distributions are using cross-entropy. The pairwise cross-entropy between two samples i and j is defined as:

$$\text{CE}(P_i, Q_j) = - \sum_{k=1}^m p_{ik} \log(q_{jk}). \quad (12)$$

The pairwise cross-entropy matrix is then computed as:

$$\text{Pairwise_CE}_{ij} = \text{CE}(P_i, Q_j) = - \sum_{k=1}^m p_{ik} \log(q_{jk}). \quad (13)$$

Finally the loss term is defined as follows:

$$\ell_{\text{contrast}} = - \frac{1}{n} \sum_{i=1}^n \log \left(\frac{\exp(\text{Pairwise_CE}_{ii})}{\sum_{j=1}^n \exp(\text{Pairwise_CE}_{ij})} \right). \quad (14)$$

Here, Pairwise_CE_{ii} represents the similarity between the same sample across the two views, while Pairwise_CE_{ij} for $i \neq j$ represents the similarity between different samples. Since we want to increase the similarity between two positive pair images' prototype affinity distribution by minimizing ℓ_{contrast} , we remove the negative sign in Equation 13 when calculating ℓ_{contrast} .

This formulation encourages high similarity between matching pairs with the same concepts while penalizing similarity to other samples with different concepts. By using this loss term, the model will learn a more coherent neighborhood, thus increasing the summarization power and faithfulness of the learned prototypes.

The ℓ_{contrast} and ℓ_{clst} are optional for training the summarization-based prototypes on single-semantic datasets, but it is necessary and beneficial to use these loss terms for cases with complex concepts and subcategories involved in either dataset.

Quantitative metric for prototype coverage We introduce an evaluation metric to quantify how well the learned set of summarization prototypes covers the whole dataset. For each sample, we consider it covered by prototype set, if its similarity to any one of the prototypes is higher than a certain threshold. For a prototype, we use the X th percentile ($X \in \{0-100\}$) of all samples’ similarities to this prototype as the cutoff threshold (i.e., similarity above threshold indicates a sample is covered by the prototype). We calculate the percentage of the covered samples in both datasets \mathcal{D} and \mathcal{D}' for each threshold, and derive an area-under-the-coverage-curve (**AUCC**) as the final coverage score for a particular learned set of summarization prototypes for \mathcal{D} and \mathcal{D}' . An example coverage curve and its area-under-the coverage-curve value is shown in Figure 18. The maximum AUCC score is 100, and the minimum is 0.

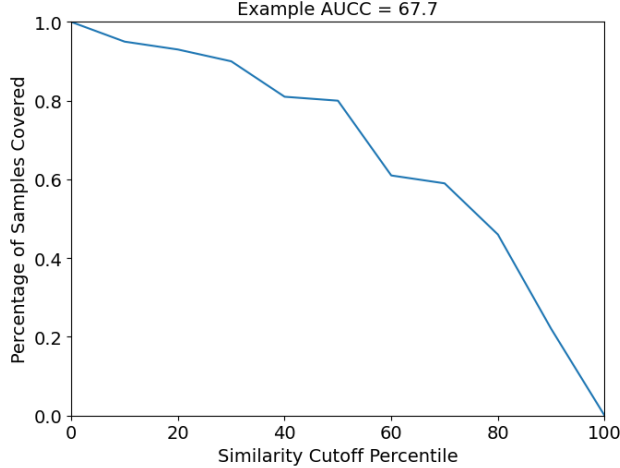


Figure 18: An example of the coverage curve and area-under-the-coverage-curve. The higher the AUCC, the more representative the given set of prototypes are of the entire dataset. However, while having too many prototypes gives higher coverage, it increases explanation complexity. We illustrate the tradeoff between explanation interpretability and prototype representativeness in Section B.1

3.5.3 CASE STUDY 1: TIME SERIES DATA - HUMAN VS MACHINE AUDIO

In this section, we compare the following two datasets:

- **Dataset \mathcal{D}** We use the human emotional speech audio dataset RAVDESS (Livingstone and Russo, 2018), which contains 928 audios of 24 different human speakers speaking two statements with a range of emotions. Statements contain “Kids are talking by the door,” 02 = “Dogs are sitting by the door.” Audios labeled neutral, happy, sad, and angry were included in this experiment. Figure 19 shows a human audio example.
- **Dataset \mathcal{D}'** For this study, we leveraged the Coqui TTS (Eren and Team, 2021) to generate AI audio. The AI audio is generated using 58 AI speakers, and includes the same set of emotions as the human in \mathcal{D} , speaking the same two statements. We generated 864 machine-generated audio signals. Figure 19 shows a machine-generated audio example.

Forming the explanation We compute the “Prototype-summarization-based explanations” described in Section 3.5. To do this, we trained a binary human vs. machine-generated audio prototype-based classifier by fine-tuning the pretrained HUBERT audio classification model (Yang et al., 2021b). 1434 audios were used for training and 358 audios were used for evaluations. During training, each audio was first sliced into 4 equal-length continuous segments and fed into the network, and each prototype is a 0.5-second segment of the audio signal. For each input audio, we use its most similar segment to each prototype to represent the affinity of the audio to the prototype. This means we can pinpoint the specific differences between human audio and machine audio, rather than just comparing entire audio signals, which is less informative.

Comparing human and machine audio prototypes It is difficult for a human to tell the difference between the generated and real audio examples, thus we did not know in advance whether there were any differences between them. We show the learned prototypes in Figure 19 - we learned four and three unique prototypes respectively for \mathcal{D} and \mathcal{D}' . These comparisons immediately provide insight into the difference

between the human and machine-generated datasets. Specifically, our results indicate that *humans tend to wait before starting to speak, whereas the machine audio starts right away*. A second observation we can make is the *machine audio waveform has highly periodic patterns where peak-to-peak intervals remain almost constant* throughout the audio piece; we can also see the *machine audio signal amplitude always changes gradually as opposed to human audio*, where there may appear more sudden amplitude changes (e.g., jagged contours of human prototype waveforms). We attribute the the second observation to human nature; human tends to speak with varying speed, loudness, and pitch, whereas synthetic audio always maintains the same pace throughout the whole speech in a more monotonic tone. The model’s insights lead immediately to ways to improve the machine-generated audio to make it more akin to human voice: (1) add in a random wait period before the machine speaks, (2) add frequency and amplitude distortion to the machine audio.

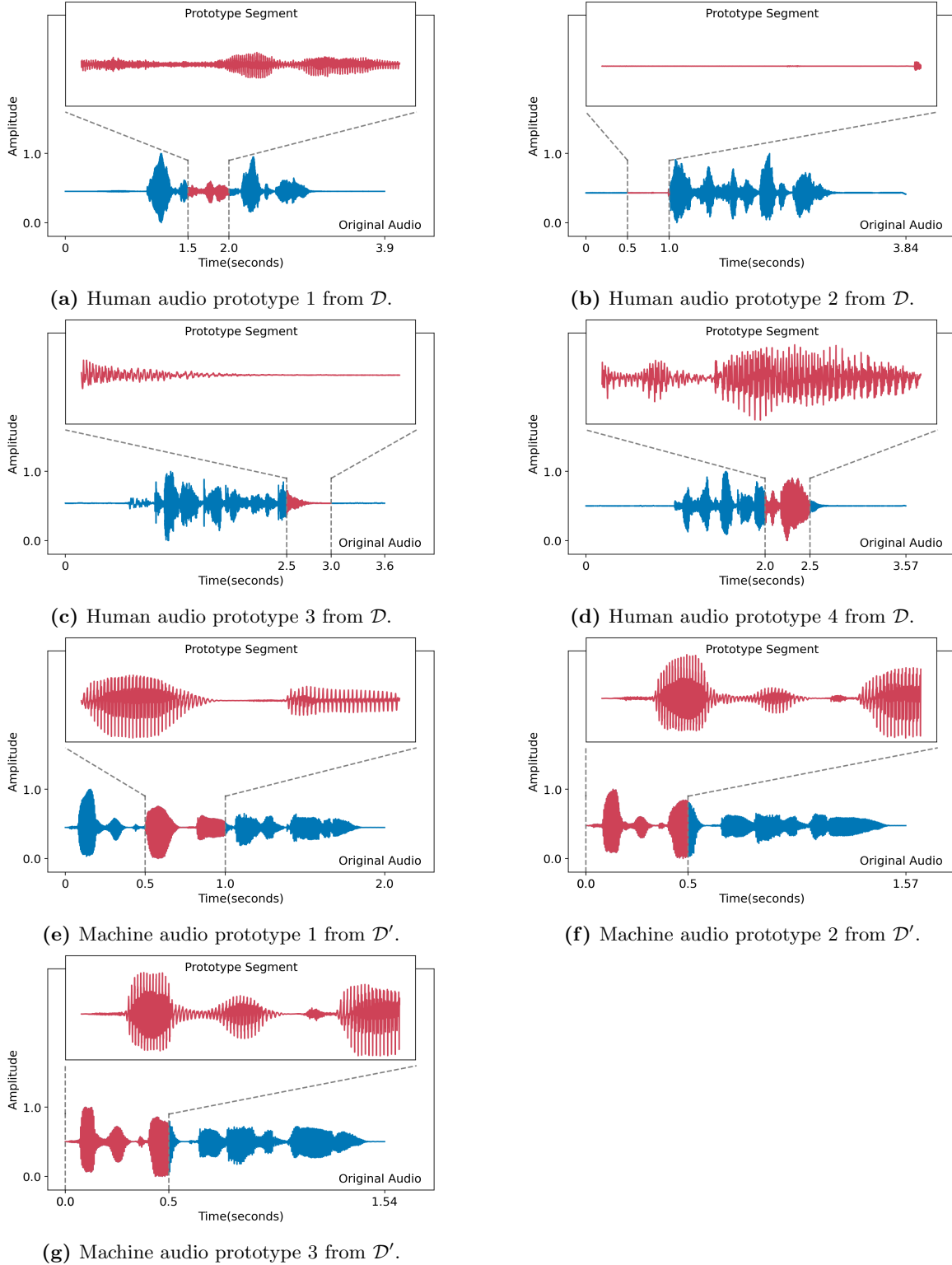


Figure 19: Learned audio segment prototypes from \mathcal{D} and \mathcal{D}' . The learned prototypes grasp the most obvious human audio characteristic, human tends to wait before speaking, whereas AI starts speaking right away. Some of the learned human prototypes represent the silent waiting period in human audio. The humans also tend to speak with varying speed, loudness, and pitch as opposed to the machine’s paced and monotonic speech, reflected by the constant peak-to-peak interval in machine audio and the very gradual changes in machine audio amplitudes.

Coverage evaluation We evaluate the coverage quality of the learned set of prototypes using the AUCC score introduced in Section 3.5. The summarization prototype has AUCC of 87.1. The coverage curve is shown in Figure 20. The learned latent space for \mathcal{D} and \mathcal{D}' is shown in Figure 21 on the right.

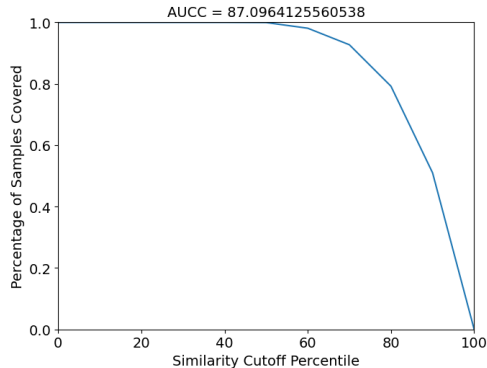


Figure 20: Coverage curve and AUCC score for the human-vs-machine audio case study.

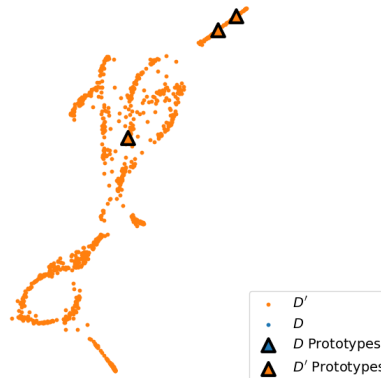


Figure 21: Visualization of the learned latent space for \mathcal{D} and \mathcal{D}' .

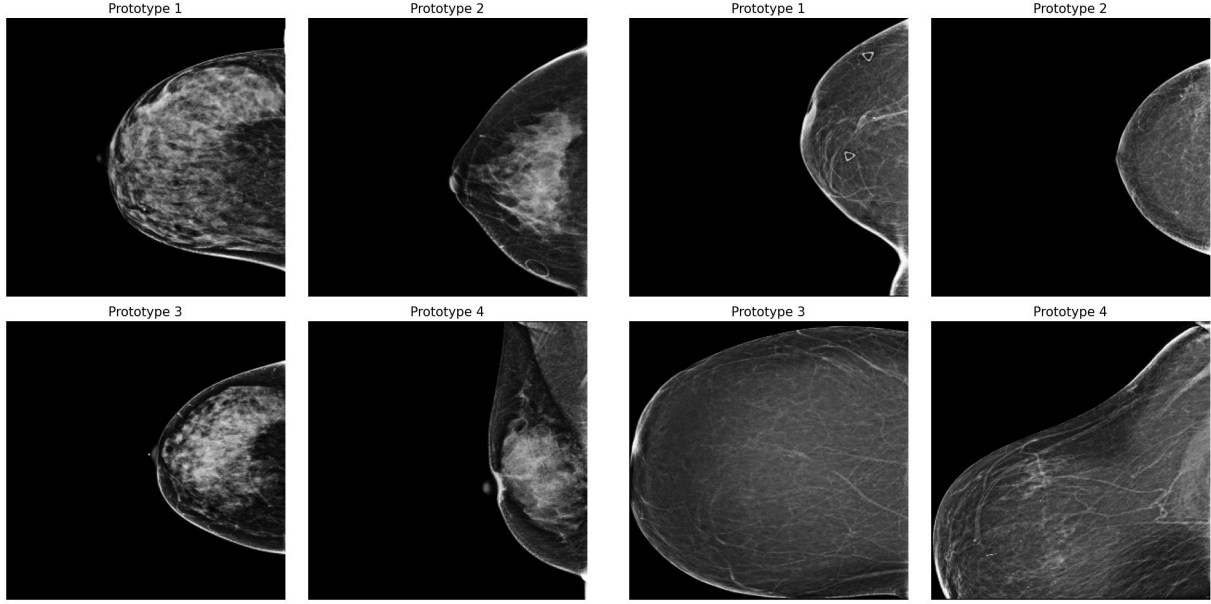
3.5.4 CASE STUDY 2: MEDICAL IMAGE DATA - MAMMOGRAPHY PATIENT POPULATION DATASET

In this case study, we assess the effectiveness of our proposed method in a realistic application, focusing on identifying differences between mammograms from two distinct patient populations. Specifically, we simulate a real-world scenario where users deploy models to analyze mammograms of women with varying tissue density distributions — a challenge commonly encountered when comparing premenopausal and postmenopausal patients or younger and older individuals. Premenopausal or younger patients often exhibit denser breast tissue, whereas postmenopausal or older patients often present with less dense tissue (Kim et al., 2020). This dataset evaluation step is crucial before deploying a clinical breast cancer risk detection model across different patient populations.

Dataset \mathcal{D} and \mathcal{D}' We use the publicly available EMBED dataset (Jeong et al., 2023). To simulate premenopausal and postmenopausal patient populations, we construct two datasets, \mathcal{D} and \mathcal{D}' , by randomly sub-sampling from EMBED. Dataset \mathcal{D} comprises 27,224 mammograms from 8,456 patients with dense breast tissue (density category *three* in EMBED) and 21,675 mammograms from 7,841 patients in density category *two*. Dataset \mathcal{D}' includes 27,224 mammograms from 2,715 patients with less dense tissue (density category *one*) and 21,675 mammograms from 7,797 patients in density category *two* (medium density). All mammograms were preprocessed to remove clinical markers and aligned such that the breast tissue faces left.

Forming the explanation For this task, we implemented the “Prototype-summarization-based explanations” described in Section 3.5. We trained a binary \mathcal{D} vs \mathcal{D}' classifier using the VGG19 feature extractor as backbone and learn four prototypes for each dataset. 97798 mammograms were used for training, and 24450 mammograms were used for testing.

Result By examining the summarization prototypes shown in Figure 22, we identified tissue density as the primary difference between \mathcal{D} and \mathcal{D}' . In mammograms, brighter areas correspond to denser tissue. Additionally, we observed that less dense tissue is often associated with larger tissue size. Without our proposed method, human users would need to manually analyze the dataset, which is a labor-intensive and time-consuming task, to reach the same conclusions.



(a) The learned prototypes for dataset \mathcal{D} with denser tissue. (b) The learned prototypes for dataset \mathcal{D}' with less dense tissue.

Figure 22: An inspection of these prototypes in \mathcal{D} and \mathcal{D}' suggests that our method is able to successfully discover tissue density and size differences between the datasets.

Robustness of the explanation To examine the robustness of our explanation result, we repeat the explanation algorithm approach on bootstrapped versions of \mathcal{D} and \mathcal{D}' . Five bootstrapped datasets were constructed by resampling by patients with replacement. As shown in Figure 34 in Appendix Section B.5, we reach the same conclusion for all the bootstrapped datasets.

Coverage evaluation We again evaluate the coverage quality of the learned set of prototypes using the AUCC score. The coverage curve is shown in Figure 23. We also display the learned latent space for \mathcal{D} and \mathcal{D}' in Figure 24 and the two datasets and the prototypes are well separated even though they contain overlapping mammograms with density category *two* (i.e. medium density breasts).

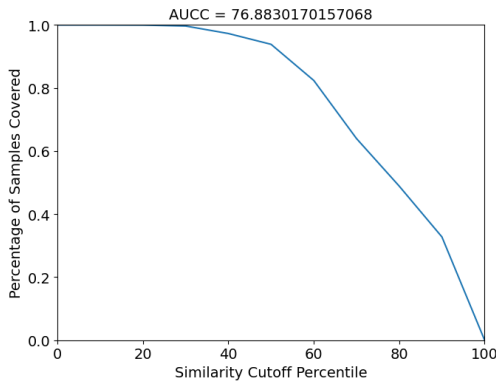


Figure 23: The coverage curve and area under the coverage curve for mammography case study.

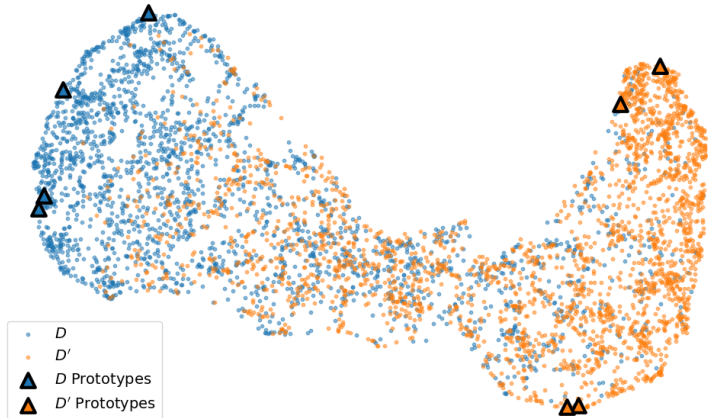


Figure 24: Visualization of the learned latent space for \mathcal{D} and \mathcal{D}' .

3.5.5 CASE STUDY 3: IMAGE DATA - OFFICE-HOME DATASET

Dataset \mathcal{D} and \mathcal{D}' For this case study, we use an established domain adaptation dataset – the Office-Home dataset Venkateswara et al. (2017). The dataset contains different styles of images from different objects that belong in office or home scenes. In this experiment, dataset \mathcal{D} contains office objects including “Table, Shelf, File_Cabinet, Printer, Calculator, Postit_Notes, Calendar, Fan, Monitor, Eraser, Folder, Pencil, Mouse, Push_Pin, Telephone, Trash_Can, Paper_Clip, Ruler, Computer, Desk_Lamp, Speaker, Pen, Scissors, Chair, Keyboard, Marker, Notebook, Clipboards, Laptop, Webcam,” whereas the \mathcal{D}' contains home objects including “ToothBrush, Bed, Oven, Lamp_Shade, Bottle, Flowers, Radio, Bike, Glasses, Flipflops, Alarm_Clock, Sneakers, Couch, Mop, Pan, Helmet, Kettle, Mug, Toys, TV, Drill, Spoon, Fork, Refrigerator, Batteries, Candles, Soda, Backpack, Exit_Sign, Curtains, Hammer, Sink, Bucket, Screwdriver, Knives.” \mathcal{D} contains 7101 images, and \mathcal{D}' contains 8487 images.

Forming the explanation In this showcase, we again use the “Prototype-summarization-based explanations” described in Section 3.5. Here, we train a binary \mathcal{D} vs \mathcal{D}' classifier with VGG-19 as the feature extractor. 12470 images were used for training, and 3118 images were used for testing. Since we have some rough prior knowledge of the content of the datasets (i.e., they contain several types of objects, but we do not know what objects), we opt for the alternative clustering term defined in Equation 10. We learn 200 prototypes for each dataset, around 6-8 prototypes for each object. We expected the algorithm to be able to capture the diverse art styles and account for the possibility of a large number of concepts/objects in each dataset.

Results In the retrospective evaluation, the proposed algorithm was able to discover all 30 office-scene objects and 34 out of the total 35 home-scene objects ($\approx 98\%$ of total objects). The learned prototypes also successfully captured a diverse set of the art styles of the objects in each dataset. We can observe that the datasets \mathcal{D} and \mathcal{D}' mainly differ in their object composition; from Figure 25, we see that \mathcal{D} contains office objects while \mathcal{D}' contains home scene objects (note that the full set of learned summarization prototypes are shown in Figure 31 and Figure 32 in Appendix Section B.4 and all prototype neighbourhoods shown in Figure 37). In addition, we can observe that there are several sets of art styles of the images; these art styles exist in both datasets, thus are not a differentiating factor between the two datasets. Based on the examination of the learned prototypes, our proposed method is able to discover the underlying differences between datasets without introducing incorrect explanations, even when there exists a large number of concepts in the two datasets.



(a) Some of the learned prototypes for dataset \mathcal{D} with office objects. (b) Some of the learned prototypes for dataset \mathcal{D}' with home objects.

Figure 25: An inspection of these prototypes in \mathcal{D} and \mathcal{D}' suggests that our method is able to successfully discover underlying differences between the datasets.

Coverage evaluation The AUCC score and the coverage curve are shown in Figure 26. The model achieved AUCC score of 95, implying that the learned prototypes are highly representative of the latent space. We visualize this learned latent space for \mathcal{D} and \mathcal{D}' in Figure 27.

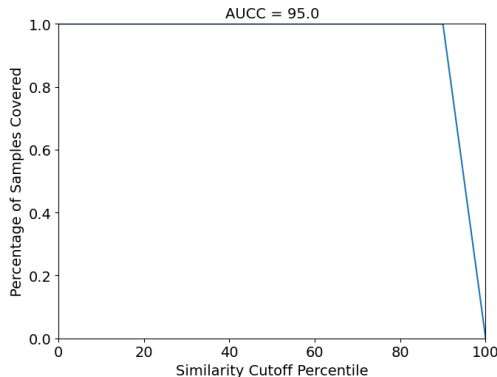


Figure 26: The coverage curve and area under the coverage curve for the office-home dataset case study.

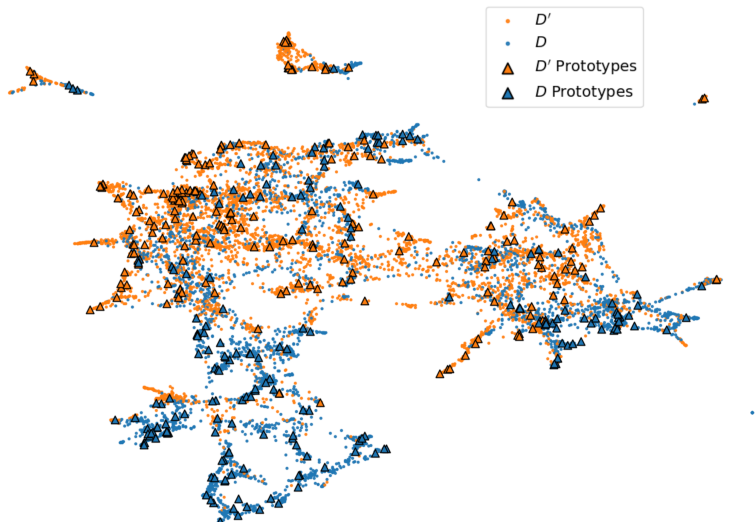


Figure 27: Visualization of the learned latent space for \mathcal{D} and \mathcal{D}' .

3.6 A Brief Note on Comparing Natural Language Datasets

Text data are central to modern machine learning, driven by advances in large language models. Unlike other modalities, text data are inherently rich in information, multi-semantic, and context-dependent, producing an intractable number of dimensions for interpretation and explanation. For example, sentiment analysis relies on understanding emotional tone, while machine translation requires preserving contextual meaning across languages, and tasks like named entity recognition demand precise extraction of information.

Our framework proposes extraction and identification of relevant attributes that serve as different dimensions of comparison between two datasets. This formalizes existing work of Elazar et al. (2023), where large text corpora were analyzed by extracting summary statistics of different attributes from the data. We show an example of how we can compare datasets under this framework in Appendix Section E.

4 Discussion

In this work, we developed practical approaches to understand shifts in data distributions in an interpretable manner. Our approaches can be seen as constituents of a dataset explanation framework, whose taxonomy is outlined in Figure 3. Dataset differences, however, can take on many forms, and what classifies as an interpretable explanation for a shift can be highly context-dependent. We view our work as one of the early efforts towards creating a formal framework for interpretable explanations of dataset differences. To this end, we first provide practical guidance, highlight some caveats associated with our explanations below and propose directions for future work.

4.1 Practical Guidance for Using Our Framework

Which explanation method should I use? Here is our recommendation for how the framework should be used:

- The first thing we generally aim for is exploratory data analysis: visualizing the datasets using a state of the art dimensionality reduction (DR) technique. This will help the user understand high-level structural differences between the datasets. It is important to use a good DR technique so it faithfully captures both global and local structure of the data – we use PaCMAP currently for this purpose because its performance on global structure preservation tends to be more reliable than other methods (Huang et al., 2022).
- If the datasets are supervised and tabular, the next step would involve understanding which features are important for the underlying predictive task and how this differs between \mathcal{D} and \mathcal{D}' . Pinpointing specific groups of influential examples that are most responsible for this difference can help inform the

user about potential biases present in a dataset (e.g., the presence of highly educated women who were still low income in Section 3.3.4).*

- The next step in the pipeline, which is applicable for tabular, image, and signal datasets (both supervised and unsupervised), is to compare local neighborhoods in the datasets. To this end, using prototypical explanations or prototype summarization explanations will provide valuable insights for image and signal data. For high dimensional tabular data, using partial prototypical explanations is a better option since it provides greater interpretability; we need only explain datasets in terms of features most important in the prototype neighborhood.

How many prototypes should I choose? Each prototype should represent a neighborhood of samples with a homogeneous concept or feature. For tabular data, one might estimate the number of prototypes in a dataset by first eyeballing the data using its PaCMAP projection and understanding the cluster structure. For unstructured data such as images, audio, or signals, we recommend starting with a higher prototype count and then gradually reducing it by visualizing the latent space with PaCMAP (or other dimension reduction techniques), as well as observing the visualized prototypes to check if there are duplicates. A rough idea of the content of the datasets would be beneficial, but not required, in this process. The prototype explanation method supports any arbitrary number of prototypes for each of \mathcal{D} and \mathcal{D}' .

What are some hyperparameter choices for influential example explanations? An important design choice for influential example explanations is the size of the set of near-optimal models (i.e., the Rashomon set) used to compute RID (Donnelly et al., 2023). We demonstrate in the appendix (e.g., Figure 43) that once the set of near-optimal models across all bootstrapped Rashomon sets is large enough (i.e., $\mathcal{O}(10^2)$), the chosen group of influential examples become relatively stable.

Once the RID importance metric is computed, we now need to decide how many influential examples to consider to explain how \mathcal{D} differs from \mathcal{D}' . The first step in this endeavour is to consider the distribution of influences across all examples in dataset \mathcal{D} . Technically, all examples with influence > 0 could be assessed, as removing them would align feature importances of \mathcal{D} and \mathcal{D}' (recall that we train a discriminator to classify whether a feature importance vector is from \mathcal{D} or \mathcal{D}'). However, Koh et al. (2019) shows that while computed influences are highly correlated with the true difference in discriminator loss (which is corroborated in this work in Figure 39, examples with small positive influences may actually reduce discriminator loss in practice, which is the opposite of the intended explanation. Hence, it is advisable to choose a small group of examples from either dataset that exhibit the highest positive influences. In this work, for example, we chose the top 1% of examples for both Adult female and HELOC low-risk datasets.

What is the computational complexity / runtime of our methods? The computational time of a prototype learning model scales with the size of the dataset. Additionally, the choice of backbone is a critical factor: while more sophisticated backbones improve feature extraction capabilities, they also lead to longer training and inference times. In contrast, increasing the number of prototypes has a negligible impact on computational time, as the additional parameters are minimal compared to the overall model.

For influential example explanations, the primary computational challenges lie in computing the Rashomon set and deriving the associated feature importances. Fortunately, practical solutions exist to address these bottlenecks. Recent work has demonstrated efficient techniques to approximate or sample from the Rashomon set (Xin et al., 2022; Ciaperoni et al., 2024). Moreover, standard decision tree methods—such as CART, when configured with different initializations (e.g., varying depth budgets and regularization parameters)—can produce a well-performing ensemble of trees. Although this approach does not offer optimality guarantees, it can be effectively used to generate variable importance estimates. A detailed comparison between the method proposed in Donnelly et al. (2023) and this decision tree-based approach remains an interesting avenue for future research.

4.2 Potential Failure Modes and Avenues for Future Research

It is unclear how to evaluate dataset-level explanations: Compared to instance-level explanations, for which there exist several evaluation criteria, there is not yet a well-defined metric to assess the quality of dataset-level explanations. For instance-level explanations, Agarwal et al. (2022) has focused on the

*For image / signal data, it may be possible to use influential examples if the dataset can be mapped to a fully interpretable latent space (i.e., each dimension of the latent space is interpretable). This is because a crucial aspect of these explanations is the computation of feature importances – if each element of the feature importance vector is interpretable (which is not the case if the data is composed of pixels), then the explanations provided can become more actionable and useful for stakeholders.

evaluation of explanations such as SHAP (Lundberg and Lee, 2017), LIME (Ribeiro et al., 2016), and IntegratedGrad (Sundararajan et al., 2017b) along several axes such as performance, faithfulness, and stability. Other works such as that of Antoran et al. (2021) have evaluated counterfactual explanations through human subject experiments (Delaney et al., 2023). Analogous tasks for dataset-level explanations are unclear at this point, as there exists no standard evaluative framework. We have some tools to quantify the efficacy of our explanations and their associated tradeoffs (e.g., AUCC score and evaluation methods in Section B in the appendix) – we consider these to be starting points for future work.

One potential evaluation would show that taking action based on our explanations would make two datasets more similar, as we showed in Sections 3.3.4 and 3.3.5. Kulinski and Inouye (2023) performed such an evaluation for their distribution shift recourse method by showing that their explanation maps Dataset A to a data distribution that more closely resembles that of Dataset B. For our framework, we show a recourse-based evaluation for influential example explanations in the appendix of this paper, however, analogous recourse is not yet clear for other forms of explanations, e.g., prototype-based explanations. We could envision generating/collecting data around prototypes that are not represented in one of the two datasets, for example. Other evaluations could include the creation of benchmark datasets with known ground truth differences that future dataset explanations need to uncover. It is also not yet clear which explanation is most suitable for a given dataset or whether this is subjective and depends on the human. Figure 3 in the paper provides a starting point for choosing an explanation method, but this is based more on the modality of the dataset rather than any inherent property of the data itself (e.g., latent structure, dimensionality).

It is hard to determine dimensions along which to compare complex natural language datasets without knowing the task: Recent work (Elazar et al., 2023) devised a tool called WIMBD which can analyze text corpora to understand the content of large-scale datasets. This provides information on summary statistics of the dataset, including token distributions, personally identifiable information, and potential biases present within the text. An explanation framework focused on comparing natural language datasets could utilize this tool and compare high level summaries of the datasets along these axes. Our work focuses on a brief formalism of some ideas in their work, suggesting that mining attributes of language corpora and generating summary statistics along those attributes (e.g., sentiment, topic) can yield valuable comparative insights between corpora. However, it is not yet clear how (or if) we can design a generic framework for more nuanced comparison of text corpora, e.g., comparing writing styles, tone, and topic compositions, or mine appropriate attributes along which to compare corpora. This is an open direction for future work.

There is no guarantee of actionability: The actionability of insights and explanations generated from our analysis may vary based on the task. A generated insight can be very actionable if a difference between datasets can be directly fixed by tuning a parameter in the generative algorithm or changing the data sampling and collection strategy. While our explanation is most useful for understanding the differences, it might be harder to design algorithms to actually mitigate those differences for the datasets in question.

For high dimensional data with non-interpretable features, the quality of the explanation depends on the quality of the latent space: For prototype-based explanations derived from training discriminators (e.g., in Sections 3.5.5 and 3.5.4), the prototype distances are computed in the latent space. The underlying assumption is that distances in the latent space are meaningful. That is, if two examples are close in latent space, it is because these examples share some similar characteristics that are semantically meaningful and interpretable to humans. In our experiments, the learned summarization prototypes worked well to aid in explaining the differences between two datasets. However, they are not guaranteed to be absolutely faithful. Users should examine the detailed visualization of the neighbourhood before taking further action. For example, there are inconsistent and ambiguous groupings shown in Figure 37, where the neighboring samples contain different objects but the same art styles, leaving the actual difference up to interpretation. We also would like to note that even though PaCMAP is optimized to maintain both global and local structure, there is no absolute guarantee of the dimension reduction algorithm’s ability to maintain faithful structure in extremely high-dimensional cases. See Huang et al. (2022) for a review on trustworthiness of dimension reduction methods.

Guarantees of completeness: Prototype methods capture a *sufficient* set of differences between the two datasets but do not necessarily capture *all* differences between the datasets. In other words, there may be other ways the datasets differ that are not captured by a single prototype model. This may not be problematic in some cases, since we may only need to see the main differences. In our current framework, users could iteratively run our algorithm in an explain-then-mitigate loop to find out more differences. Although

fully delineating the differences between datasets may not be immediately necessary for practical purposes, it presents a promising avenue for future research.

Influential example explanations can only be used if local feature importance can be computed:

Some methods such as permutation importance and Gini-impurity provide global feature importances (GiFIMs) directly, with no information given on the importance of a feature for the prediction of a particular instance (LiFIM). Because our explanation technique relies on a) computing GiFIMs as an aggregate of LiFIMs, and b) computing influences from LiFIMs, methods that skip this step will not be compatible with our technique. Future work will seek to implement this explanation in scenarios where only a global feature importance metric is provided.

Influential example explanations may require access to a Rashomon set: In our version of influential example explanations, we were considering features that were important across the entire set of near optimal models (i.e., the Rashomon set). We also evaluated decision trees as the model class, however, other the method can easily be adaptable to other model classes. The only caveat is that it can be expensive to compute this set, especially for datasets with many features and with more complex model classes. In these situations, a useful proxy metric can be to examine an easily computable subset of representative models rather than the entire Rashomon set (e.g., collecting all empirical risk minimizers trained on bootstrapped versions of the data).

AUCC score: The AUCC score is a useful metric to evaluate the coverage of the prototypes. It is worth noting that the score could be trivially maximized when the model learns a trivial solution that aligns every single sample to prototypes regardless of content (even though we offer optional learning targets that regularize against this behavior) or when the user chooses an unreasonably large number of prototypes as the hyperparameter. The AUCC score should be interpreted along with the visualization of the prototype neighborhood and, optionally, the visualization of the latent space.

5 Conclusion

In conclusion, we developed an explainable AI paradigm for explaining the differences between any two datasets in an interpretable manner. The suite of approaches proposed in this work provides end users with insights and actionable clues to understand and mitigate the differences. With case studies and experiments that cover a variety of data modalities and common machine learning tasks, we demonstrate the comprehensiveness and adaptability of our methods. Our framework is most useful for detecting biases in synthetic data, understanding erroneous examples in specific regions of the input space, and exploring the impact of discrepancies on model performance. Our study could potentially improve machine learning algorithm robustness in the real world by allowing researchers to examine changing factors, enabling future studies to improve generative algorithms, and other data science / exploratory data analysis applications. We envision future work focusing more on evaluation and the HCI aspect of this research direction, aiming to understand how practitioners can benefit most from our framework.

6 Ethics note

This work could be used to help improve deepfake techniques and adverse / malicious machine signal, image, and tabular data generation in various domains.

7 Acknowledgement

We thank Jon Donnelly and Srikar Katta for helpful discussions and guidance on using Rashomon variable importance. We acknowledge support from NIH 1R01HL166233-01, NIH/NIBIB 5P41-EB028744, and NSF IIS-2130250.

References

Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. OpenXAI: Towards a Transparent Evaluation of Model Explanations.

- In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15784–15799. Curran Associates, Inc., 2022.
- Javier Antoran, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a CLUE: A Method for Explaining Uncertainty Estimates. In *International Conference on Learning Representations*, 2021.
- Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A Case-Based Interpretable Deep Learning Model for Classification of Mass Lesions in Digital Mammography. *Nature Machine Intelligence*, 3(12):1061–1070, 2021.
- Alina Jade Barnett, Zhicheng Guo, Jin Jing, Wendong Ge, Cynthia Rudin, and M. Brandon Westover. Interpretable Machine Learning System to EEG Patterns on the Ictal-Interictal-Injury Continuum, 2023.
- Leo Breiman. Statistical Modeling: The Two Cultures (With Comments and a Rejoinder by the Author). *Statistical Science*, 16(3):199–231, 2001.
- Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. *This Looks Like That: Deep Learning for Interpretable Image Recognition*. Curran Associates Inc., Red Hook, NY, USA, 2019a.
- Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.
- Richard J. Chen, Ming Y. Lu, Tiffany Y. Chen, Drew F. K. Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497, Jun 2021. ISSN 2157-846X.
- Prathyush Chirra, Patrick Leo, Michael Yim, B. Nicolas Bloch, Ardeshtir R. Rastinehad, Andrei Purysko, Mark Rosen, Anant Madabhushi, and Satish Viswanath. Empirical Evaluation of Cross-Site Reproducibility in Radiomic Features for Characterizing Prostate MRI. In Nicholas Petrick and Kensaku Mori, editors, *Medical Imaging 2018: Computer-Aided Diagnosis*, volume 10575 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 105750B, February 2018.
- Mohammad Amin Choukali, Mehdi Chehel Amirani, Morteza Valizadeh, Ata Abbasi, and Majid Komeili. Pseudo-Class Part Prototype Networks for Interpretable Breast Cancer Classification. *Scientific Reports*, 14(1):10341, 2024.
- Martino Ciaperoni, Han Xiao, and Aristides Gionis. Efficient Exploration of the Rashomon Set of Rule Set Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 478–488. ACM, 2024.
- Eoin Delaney, Arjun Pakrashi, Derek Greene, and Mark T. Keane. Counterfactual Explanations for Misclassified Images: How Human and Machine Explanations Differ. *Artificial Intelligence*, 324:103995, 2023. ISSN 0004-3702.
- Jiayun Dong and Cynthia Rudin. Exploring the Cloud of Variable Importance for the Set of All Good Models. *Nature Machine Intelligence*, 2(12):810–824, 2020.
- Jon Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable Protopnet: An Interpretable Image Classifier Using Deformable Prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10265–10275, 2022.
- Jon Donnelly, Srikar Katta, Cynthia Rudin, and Edward Browne. The Rashomon Importance Distribution: Getting RID of Unstable, Single Model-Based Variable Importance. *Advances in Neural Information Processing Systems*, 36:6267–6279, 2023.
- Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hanna Hajishirzi, Noah A. Smith, and Jesse Dodge. What’s In My Big Data?, 2023.

Gölge Eren and The Coqui TTS Team. Coqui TTS, Jan 2021.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. 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):1–81, 2019.

Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-distribution robustness via targeted augmentations. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

Mauro Giuffrè and Dennis L. Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *npj Digital Medicine*, 6(1):186, Oct 2023. ISSN 2398-6352.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv preprint arxiv:2301.07597*, 2023.

Lin Lawrence Guo, Stephen R. Pfohl, Jason Fries, Alistair E. W. Johnson, Jose Posada, Catherine Aftandilian, Nigam Shah, and Lillian Sung. Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. *Scientific Reports*, 12(1):2726, Feb 2022. ISSN 2045-2322.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

Hsiang Hsu and Flavio Calmon. Rashomon Capacity: A Metric for Predictive Multiplicity in Classification. *Advances in Neural Information Processing Systems*, 35:28988–29000, 2022.

Haiyang Huang, Yingfan Wang, Cynthia Rudin, and Edward P Browne. Towards a Comprehensive Evaluation of Dimension Reduction Methods for Transcriptomic Data Visualization. *Communications Biology*, 5(1):719, 2022.

Sooyong Jang, Sangdon Park, Insup Lee, and Osbert Bastani. Sequential Covariate Shift Detection Using Classifier Two-Sample Tests. In *International Conference on Machine Learning*, pages 9845–9880. PMLR, 2022.

Jiwoong J Jeong, Brianna L Vey, Ananth Bhimireddy, Thomas Kim, Thiago Santos, Ramon Correa, Raman Dutt, Marina Mosunjac, Gabriela Oprea-Ilie, Geoffrey Smith, et al. The EMory BrEast Imaging Dataset (EMBED): A Racially Diverse, Granular Dataset of 3.4 Million Screening and Diagnostic Mammographic Images. *Radiology: Artificial Intelligence*, 5(1):e220047, 2023.

Eun Young Kim, Yoosoo Chang, Jiin Ahn, Ji-Sup Yun, Yong Lai Park, Chan Heun Park, Hocheol Shin, and Seungho Ryu. Mammographic breast density, its changes, and breast cancer risk in premenopausal and postmenopausal women. *Cancer*, 126(21):4687–4696, November 2020.

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR, 06–11 Aug 2017.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the Accuracy of Influence Functions for Measuring Group Effects. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

Sean Kulinski and David I. Inouye. Towards Explaining Distribution Shifts. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17931–17952. PMLR, 23–29 Jul 2023.

Bogdan Kulynych, Hsiang Hsu, Carmela Troncoso, and Flavio P Calmon. Arbitrary Decisions Are a Hidden Cost of Differentially Private Training. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1609–1623, 2023.

- Jimmy Lin, Chudi Zhong, Diane Hu, Cynthia Rudin, and Margo Seltzer. Generalized and Scalable Optimal Sparse Decision Trees. In *Proceedings of the 37th International Conference on Machine Learning, ICML'20*. JMLR.org, 2020.
- I. E. Livieris, N. Alimpertis, G. Domalis, and D. Tsakalidis. *An Evaluation Framework for Synthetic Data Generation Models*, page 320–335. Springer Nature Switzerland, 2024. ISBN 9783031632198.
- S. R. Livingstone and F. A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A Dynamic, Multimodal Set of Facial and Vocal Expressions in North American English. *PLoS One*, 13(5):e0196391, 2018.
- Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- Chiyu Ma, Brandon Zhao, Chaofan Chen, and Cynthia Rudin. This Looks Like Those: Illuminating Prototypical Concepts Using Multiple Visualizations. *Advances in Neural Information Processing Systems*, 36, 2024.
- Dominique Makowski, Tam Pham, Zen J. Lau, Jan C. Brammer, François Lespinasse, Hung Pham, Christopher Schölzel, and S. H. Annabel Chen. NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing. *Behavior Research Methods*, 53(4):1689–1696, feb 2021.
- Charles Marx, Flavio Calmon, and Berk Ustun. Predictive Multiplicity in Classification. In *International Conference on Machine Learning*, pages 6765–6774. PMLR, 2020.
- Hayden McTavish, Chudi Zhong, Reto Achermann, Ilias Karimalis, Jacques Chen, Cynthia Rudin, and Margo Seltzer. Fast Sparse Decision Tree Optimization via Reference Ensembles. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 9604–9613, 2022.
- Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. This Looks Like That, Because... Explaining Prototypes for Interpretable Image Recognition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 441–456. Springer, 2021a.
- Meike Nauta, Ron Van Bree, and Christin Seifert. Neural Prototype Trees for Interpretable Fine-Grained Image Recognition. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 14933–14943, 2021b.
- Meike Nauta, Jörg Schlötterer, Maurice Van Keulen, and Christin Seifert. Pip-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2744–2753, 2023.
- Pedro C. Neto, Rafael M. Mamede, Carolina Albuquerque, Tiago Gonçalves, and Ana F. Sequeira. Massively Annotated Datasets for Assessment of Synthetic and Real Data in Face Recognition, 2024.
- Gabriel Peyré and Marco Cuturi. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. ISSN 1935-8237.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018.
- Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. ProtoPShare: Prototypical Parts Sharing for Similarity Discovery in Interpretable Image Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021.
- Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable Image Classification With Differentiable Prototypes Assignment. In *European Conference on Computer Vision*, pages 351–368. Springer, 2022.

- Dawid Rymarczyk, Joost van de Weijer, Bartosz Zieliński, and Bartłomiej Twardowski. ICICLE: Interpretable Class Incremental Continual Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1887–1898, 2023.
- Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.
- Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT ’22. ACM, June 2022.
- Shapley, Lloyd S. *A Value for N-Person Games*. Princeton University Press, Princeton, NJ, 1953.
- Zheyang Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards Out-of-Distribution Generalization: A Survey. *arXiv preprint*, 2021.
- Yong-Min Shin, Sun-Woo Kim, Eun-Bi Yoon, and Won-Yong Shin. Prototype-Based Explanations for Graph Neural Networks (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11): 13047–13048, June 2022.
- Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.
- Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017.
- Gavin Smith, Roberto Mansilla, and James Goulding. Model Class Reliance for Random Forests. *Advances in Neural Information Processing Systems*, 33:22305–22315, 2020.
- Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*, 8(5), 2007.
- Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-Distribution Detection with Rectified Activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021.
- Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-Time Training with Self-Supervision for Generalization Under Distribution Shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3319–3328. JMLR.org, 2017a.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, page 3319–3328. JMLR.org, 2017b.
- Jessica Torres-Soto and Euan A Ashley. Multi-Task Deep Learning for Cardiac Rhythm Detection in Wearable Devices. *NPJ Digit. Med.*, 3(1):116, September 2020.
- Allan Tucker, Zhenchen Wang, Ylenia Rotalinti, and Puja Myles. Generating High-Fidelity Synthetic Patient Data for Assessing Machine Learning Healthcare Software. *npj Digital Medicine*, 3(1):147, Nov 2020. ISSN 2398-6352.
- Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards Robust and Reliable Algorithmic Recourse. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16926–16937. Curran Associates, Inc., 2021.
- Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* ’19, page 10–19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361255.

- Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5018–5027, 2017.
- Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis McCarthy, Helen Frazer, and Gustavo Carneiro. Learning Support and Trivial Prototypes for Interpretable Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2062–2072, 2023.
- Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable Image Recognition by Constructing Transparent Embedding Space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 895–904, 2021a.
- Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, Trimap, and PaCMAP for Data Visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021b.
- Jamelle Watson Daniels, Solon Barocas, Jake M Hofman, and Alexandra Chouldechova. Multi-Target Multiplicity: Flexibility and Fairness in Target Specification Under Resource Constraints. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 297–311, 2023a.
- Jamelle Watson Daniels, David C Parkes, and Berk Ustun. Predictive Multiplicity in Probabilistic Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10306–10314, 2023b.
- Yuanyuan Wei, Roger Tam, and Xiaoying Tang. MProtoNet: A Case-Based Interpretable Model for Brain Tumor Classification With 3D Multi-Parametric Magnetic Resonance Imaging. In *Medical Imaging with Deep Learning*, pages 1798–1812. PMLR, 2024.
- Frank Willard, Luke Moffett, Emmanuel Mokel, Jon Donnelly, Stark Guo, Julia Yang, Giyoung Kim, Alina Jade Barnett, and Cynthia Rudin. This Looks Better Than That: Better Interpretable Models With ProtoPNext. *arXiv preprint arXiv:2406.14675*, 2024.
- Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the Whole Rashomon Set of Sparse Decision Trees. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 14071–14084. Curran Associates, Inc., 2022.
- Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized Out-of-Distribution Detection: A Survey. *arXiv preprint arXiv:2110.11334*, 2021a.
- Julia Yang, Alina Jade Barnett, Jon Donnelly, Satvik Kishore, Jerry Fang, Fides Regina Schwartz, Chaofan Chen, Joseph Y Lo, and Cynthia Rudin. FPN-Iaia-BI: A Multi-Scale Interpretable Deep Learning Model for Classification of Mass Margins in Digital Mammography. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5003–5009, 2024.
- Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. Superb: Speech Processing Universal PERFORMANCE Benchmark. *arXiv preprint arXiv:2105.01051*, 2021b.
- Wenzhuo Yang, Jia Li, Caiming Xiong, and Steven C. H. Hoi. Mace: An Efficient Model-Agnostic Framework for Counterfactual Explanation, 2022.
- Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change Is Hard: A Closer Look at Subpopulation Shift. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39584–39622. PMLR, 23–29 Jul 2023.
- Haoran Zhang, Harvineet Singh, Marzyeh Ghassemi, and Shalmali Joshi. "Why Did the Model Fail?" Attributing Model Performance Changes to Distribution Shifts. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.

- Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain Adaptation Under Target and Conditional Shift. In *International conference on machine learning*, pages 819–827. PMLR, 2013.
- Zhiying Zhu, Weixin Liang, and James Zou. GSCLIP: A Framework for Explaining Distribution Shifts in Natural Language. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022), DataPerf Workshop*, 2022.

Appendix A. Cardiac Signal Simulation Parameters

The synthetic PPG (Cardiac) signals were generated using the following parameters using neurokit2 Makowski et al. (2021). Parameter values were chosen either as a fixed value or randomly chosen from the listed value range.

Parameters	Value Range
sampling_rate	80
heart_rate	81 - 100
frequency_modulation	5 - 21
ibi_randomness	5 - 21
drift	0 - 1
powerline_amplitude	0 - 1
burst_amplitude	0 - 1
burst_number	0 - 9
noise_shape	laplace
artifacts_amplitude	1
artifacts_frequency	5 - 9
artifacts_number	15 - 31
linear_drift	True / False

Appendix B. Evaluation of Prototypical Explanations

We perform several evaluations of our explanations in order to characterise their quality under different scenarios. As is typical with function approximation, approximation faithfulness and completeness is sacrificed if we reduce the complexity of the explanations. We define explanation quality in terms of its *faithfulness* to the underlying difference between two datasets:

Definition 12 (Faithfulness) *The faithfulness of a dataset difference explanation is the extent to which the explanation captures the actual difference between two datasets. The exact measure of faithfulness depends on the type of dataset explanation being generated.*

B.1 Prototype-Based Explanations for NSPD and NSDD

The main assumption made for prototype-based explanations of this type is that the prototype is representative of the neighbourhood. This is generally true if the neighbourhood is small, but the quality of the explanation will degrade as the neighbourhood grows, because it will contain more varied data. On the other hand, the explanations are more general if they cover a larger neighbourhood. This is analogous to an argument made in selecting the number of clusters for k -means, except that prototypes are a generalisation of cluster-centres and can be chosen to prioritise certain neighbourhoods. Based on the above analysis, we have two conflicting desiderata for an ideal prototypical explanation:

- Each prototype must faithfully represent its neighbourhood, which means the neighbourhood should be sufficiently small.
- The explanations must be general, which means the neighbourhood should be sufficiently large. (This leads to a smaller overall number of prototypes.)

We illustrate the tradeoffs associated with these desiderata for explaining the HELOC and Adult datasets in the figures below. We assume a similar setup as in Sections 3.4.5 and 3.4.4.

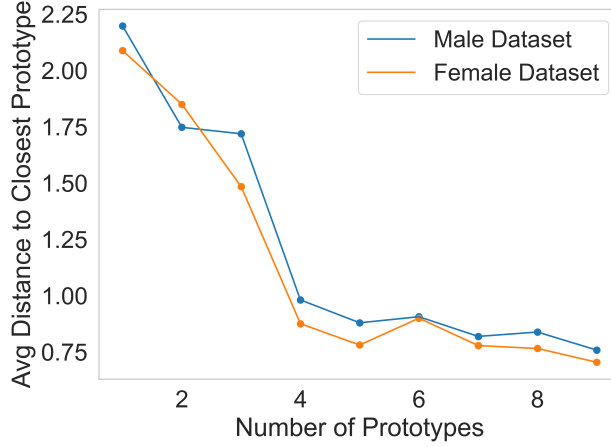


Figure 28: Illustrating the faithfulness-interpretability tradeoff for prototype-based explanations on the Adult male and female datasets. Here, the complexity – and therefore interpretability – of the explanation is determined by the number of prototypes. The representativeness of a prototype in its neighbourhood – as measured by the average distance of points to the prototype – determines faithfulness.

B.2 Choosing Partial Prototypes

For tabular datasets such as HELOC, we want to choose a subset of features for each prototype that best represent the neighbourhood. In Section 3.3.5, we selected features according to the following desiderata:

- The chosen features should not vary much in the neighbourhood of the prototype.
- The chosen features should be important for both the prototype and its neighbourhood. Here, importance is measured relative to the underlying task at hand.

Because the distance metric used in computing the NSPD and NSDD involved all of the features, when we then restrict ourselves to using a subset of the features, the observations within the neighbourhood may differ from each other on this subset.

To capture this degradation in the distance metric, we use two measures of faithfulness:

Definition 13 (Random Triplet Accuracy (from Wang et al. (2021b))) Choose any 3 points from a dataset over $N = 1000$ trials. The random triplet accuracy then measures the proportion of trials where the triplets maintain their relative order in both low and high-dimensional space feature spaces.

Definition 14 (Global Permutation Accuracy) The global permutation accuracy captures the distance between permutations. Given two arrays containing separate distance measurements, first argsort both arrays. The global permutation accuracy is the proportion of positions where the rankings of elements in both arrays match (e.g., if element 1 has the 36th largest distance in both arrays, then this is considered one match).

Let $X_p \in \mathcal{X}$ be a prototype of interest in dataset \mathcal{D} . Let V be all the points in its neighbourhood. Let $\{\alpha_{(1)}, \dots, \alpha_{(K)}\}$ represent the chosen subset of K features, with corresponding partial prototype $X_p[\alpha_{(1)}, \dots, \alpha_{(K)}]$. We now illustrate this degradation in explanation quality as a function of K .

1. We first order the points in V according to distance from the prototype X_p . Let $\sigma(V)$ represent this ordering.
2. We then select K random features from the feature set and compute the partial prototype $X_p[\alpha_{(1)}, \dots, \alpha_{(K)}]$.
3. We now order points in V according to distance to the partial prototype. Let $\sigma_K(V)$ represent this new ordering of points.
4. We then compute the Random Triplet and Global Permutation Accuracies of the chosen K features.

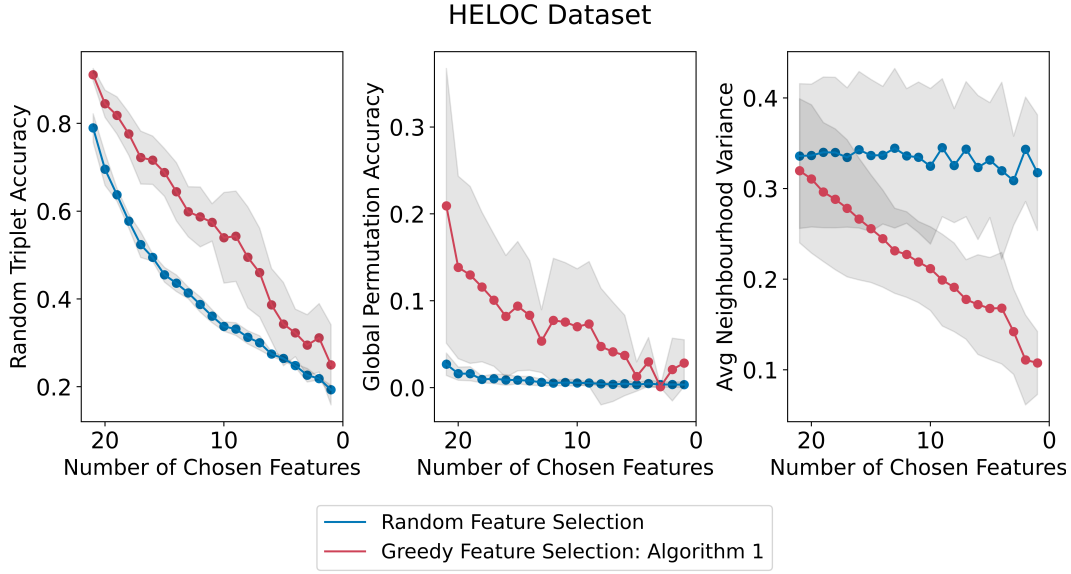


Figure 29: Illustrating the faithfulness-interpretability tradeoff (averaged across all prototypes) for partial prototype-based explanations on the HELOC risky and non-risky datasets in Section 3.4.5. The greedy feature selection procedure in Algorithm 2 is choosing features which are value-stable in the prototype neighbourhood (i.e., they vary the least) by setting c_1 and c_2 to 0. As the number of features chosen for a partial prototype reduces, there is an increasing degradation in local and global structure preservation. This is measured using two related interpretations of faithfulness – Random Triplet Accuracy (left) and Global Permutation Accuracy (middle). We also note that the variance around the neighbourhood (right figure) is lower than with random feature selection, implying that the partial prototype generated using our method is more faithful to the neighbourhood.

B.3 Partial Prototype Feature Selection: Sensitivity Analysis

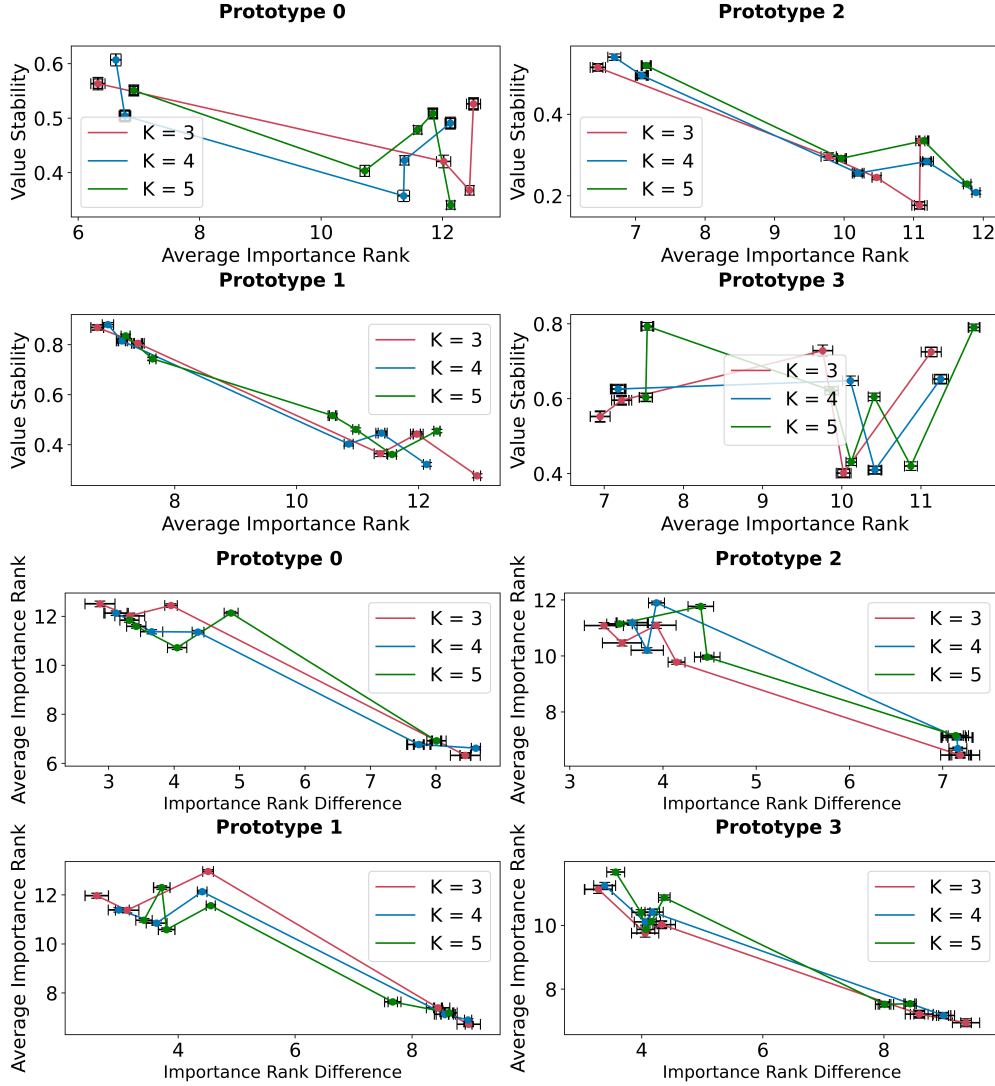


Figure 30: Relationship Between Rank Difference and Value Stability (top) and Rank Difference and Absolute Rank (bottom) of chosen partial prototype features (HELOC dataset). These curves were obtained by sampling random values of c_1 , c_2 , and c_3 in a logarithmically spaced interval $[10^{-2}, 10]$. For a given (c_1, c_2, c_3) tuple (which represents a single point on both curves), we then found the $K \in \{3, 4, 5\}$ most relevant features of the prototype and compute the rank difference, value stability, and average rank of these features in the δ neighbourhood. We chose δ as the 10th percentile distance of all points from the prototype.

From Figure 30, we can see the following:

- There exists a tradeoff between importance rank difference and average importance rank – this is analogous to a bias-variance tradeoff. In particular, one can choose a feature that is on average more important in a prototype neighbourhood, but this feature will have higher variation in importance rank. That is, the absolute rank difference between the feature’s importance for the prototype and the importance of its neighbourhood point will be higher.
- There is also a tradeoff between average importance rank and value stability. This means that one can choose a feature that is on average more important to the underlying task in a prototype neighbourhood, but this feature is likely to take on a larger spread of values.

Navigating these tradeoffs according to user requirements is an essential aspect of choosing the correct partial prototype features that are truly representative of the neighbourhood.

B.4 All learned summarization prototypes for the Office-Home Experiment

Here, we show all the learned prototypes for summarizing the differences between dataset \mathcal{D} and \mathcal{D}' in Section 3.5.5.

For dataset \mathcal{D} (which contains office objects):

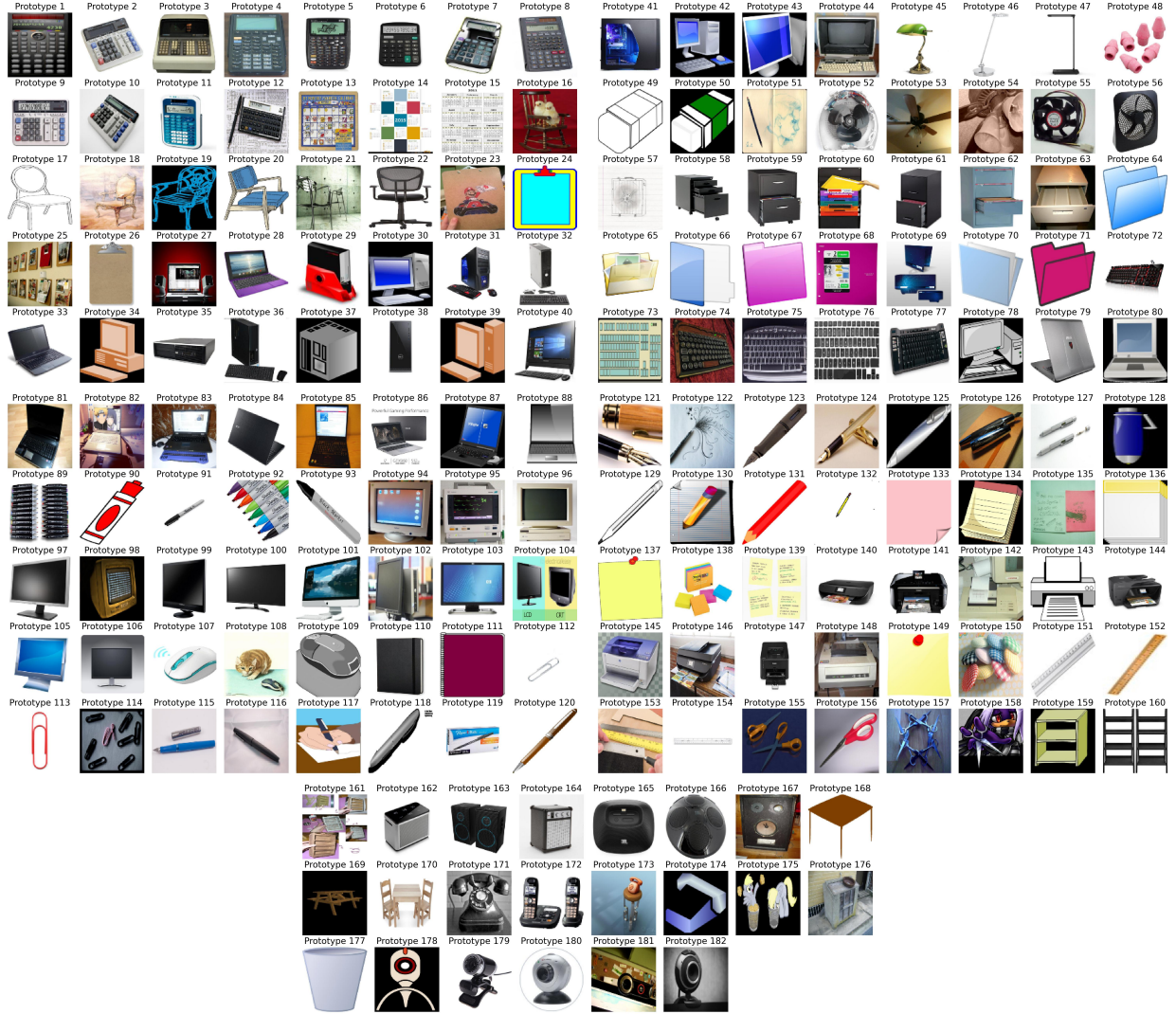


Figure 31: All learned prototypes for dataset \mathcal{D} with office objects.

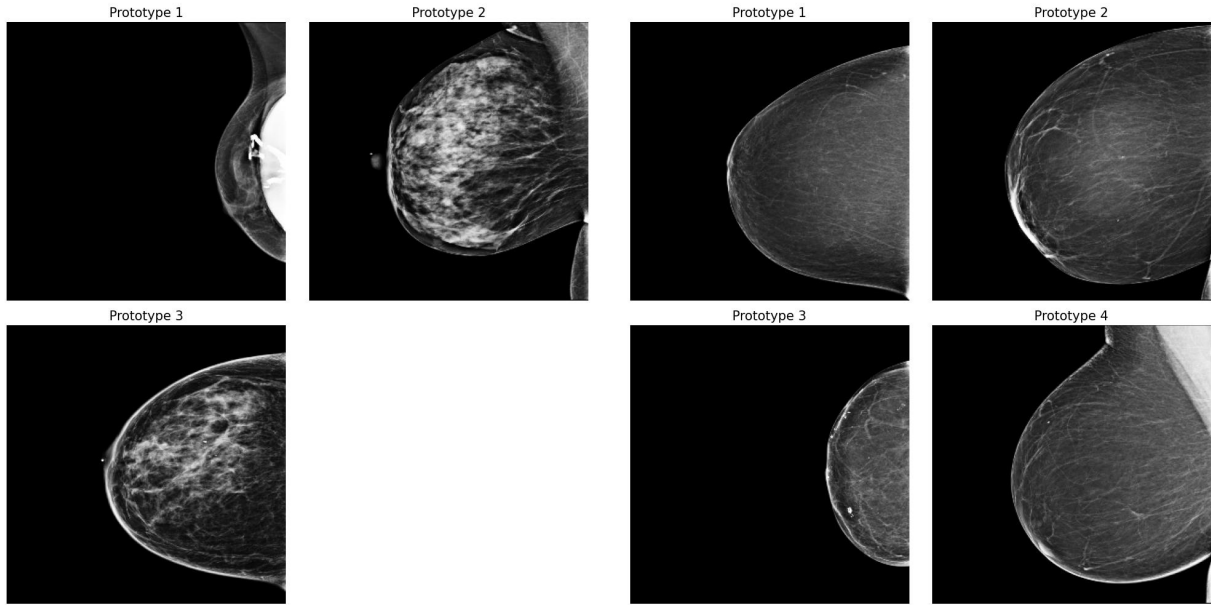
For dataset \mathcal{D}' (which contains home objects):



Figure 32: All learned prototypes for dataset \mathcal{D}' with home objects.

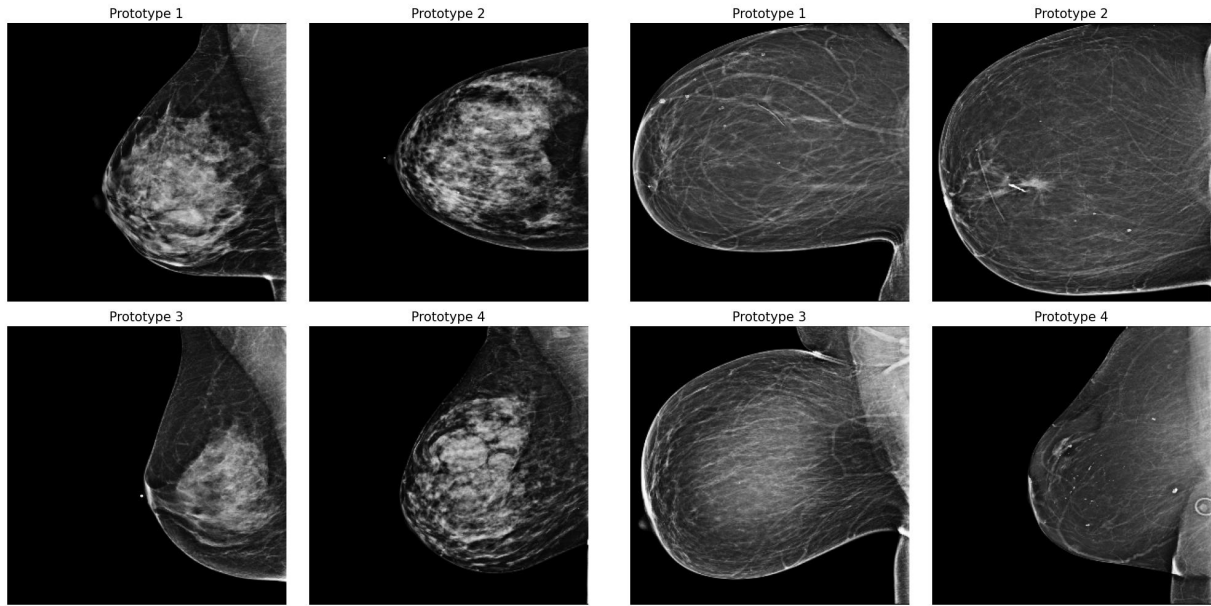
B.5 Robustness of Prototype-summarization-based explanations

In this section, we present the five sets of summarization prototypes learned from the bootstrap versions of the mammography dataset \mathcal{D} and \mathcal{D}' described in Section 3.5.4. We are able to reach the same conclusion from all different bootstraps, thus demonstrating the robustness of our approach.



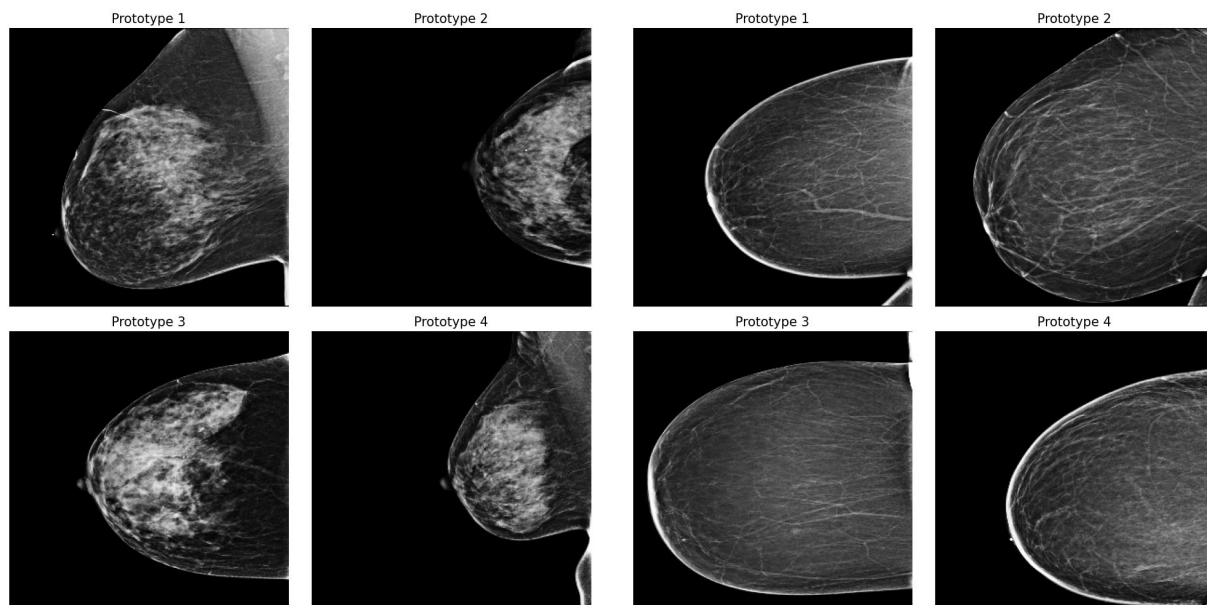
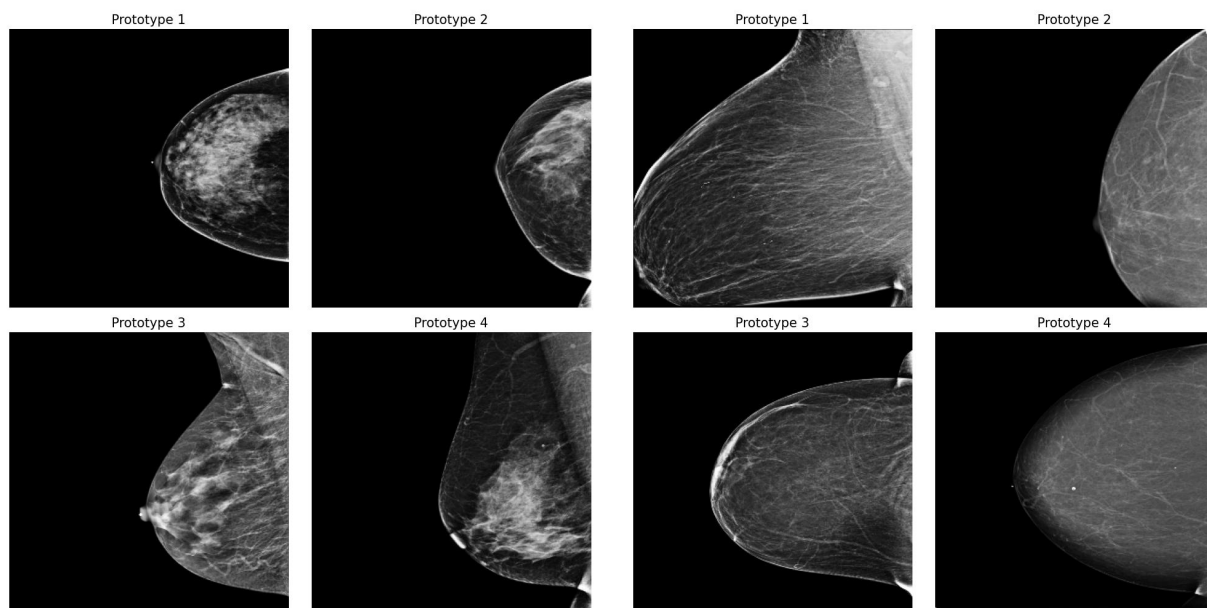
(a) \mathcal{D} prototypes from bootstrap 1.

(b) \mathcal{D}' prototypes from bootstrap 1.



(c) \mathcal{D} prototypes from bootstrap 2.

(d) \mathcal{D}' prototypes from bootstrap 2.

(e) \mathcal{D} prototypes from bootstrap 3.(f) \mathcal{D}' prototypes from bootstrap 3.(g) \mathcal{D} prototypes from bootstrap 4.(h) \mathcal{D}' prototypes from bootstrap 4.

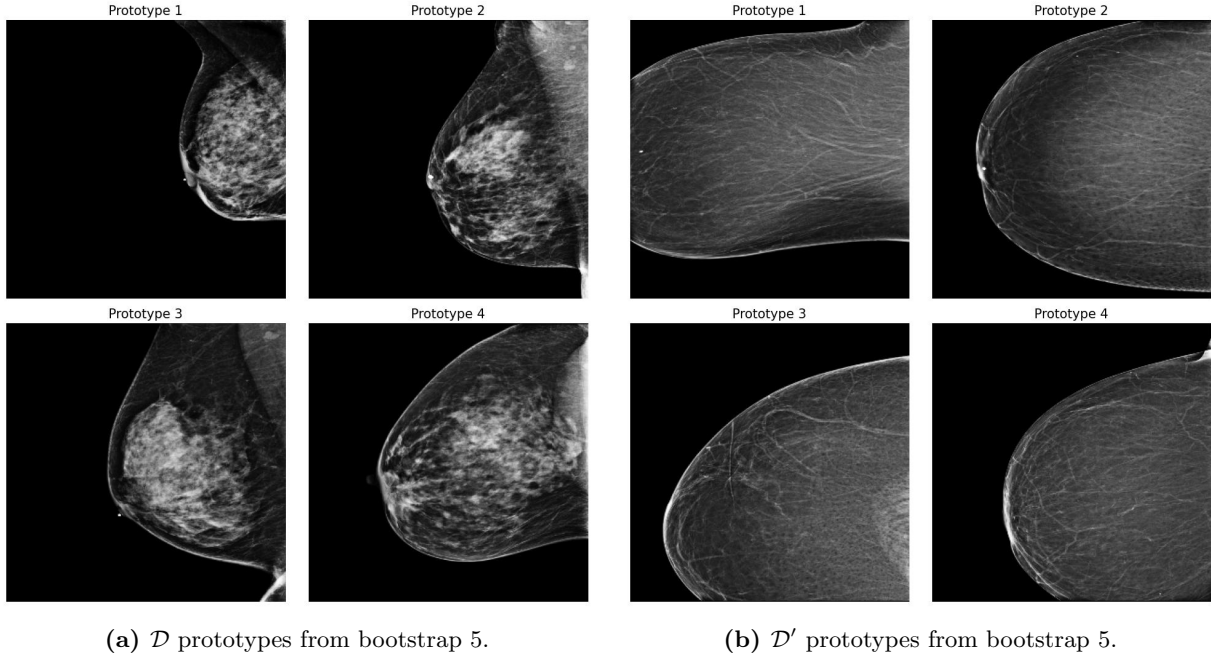


Figure 34: All learned prototypes for dataset \mathcal{D}' with home objects. The model only learned 3 unique prototypes for Bootstrap 1 of D .

B.6 Are Prototypical Neighborhoods Faithful for Vision and Signal Data?

We examine the robustness and faithfulness of the learned prototype neighbourhoods from our summarization approach in each experiment. To do so, we visualize a selection of prototypes along with 10 nearest samples for each prototype. We can see that all the models have learned a high quality latent space, as neighbouring samples appear to be very similar to the prototype. Note that while Bootstrap 1 in Figure 34a only learned 3 unique prototypes for D , this does not detract from our observations.

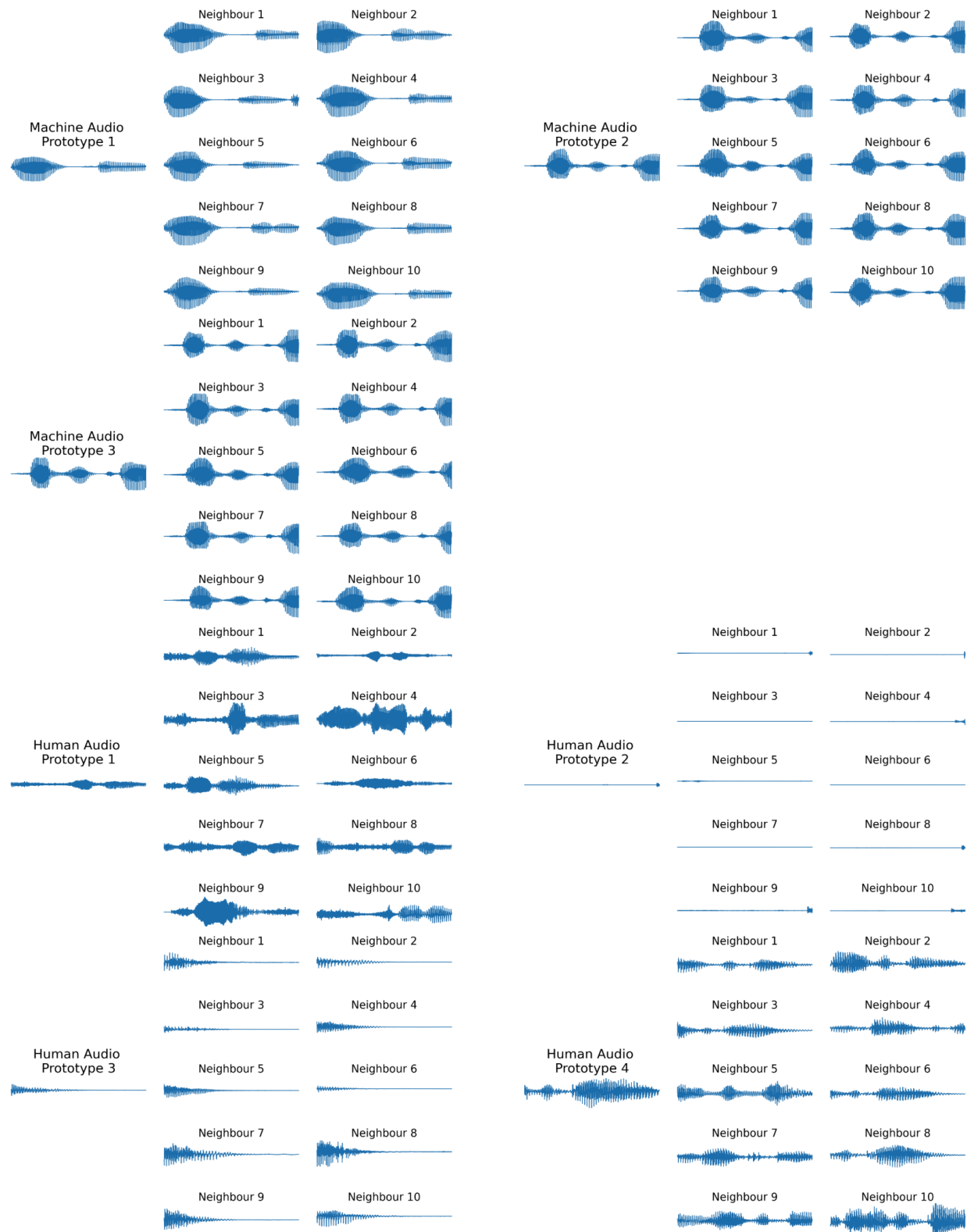


Figure 35: Neighbouring sample visualization for audio prototypes learned in Section 3.5.3.

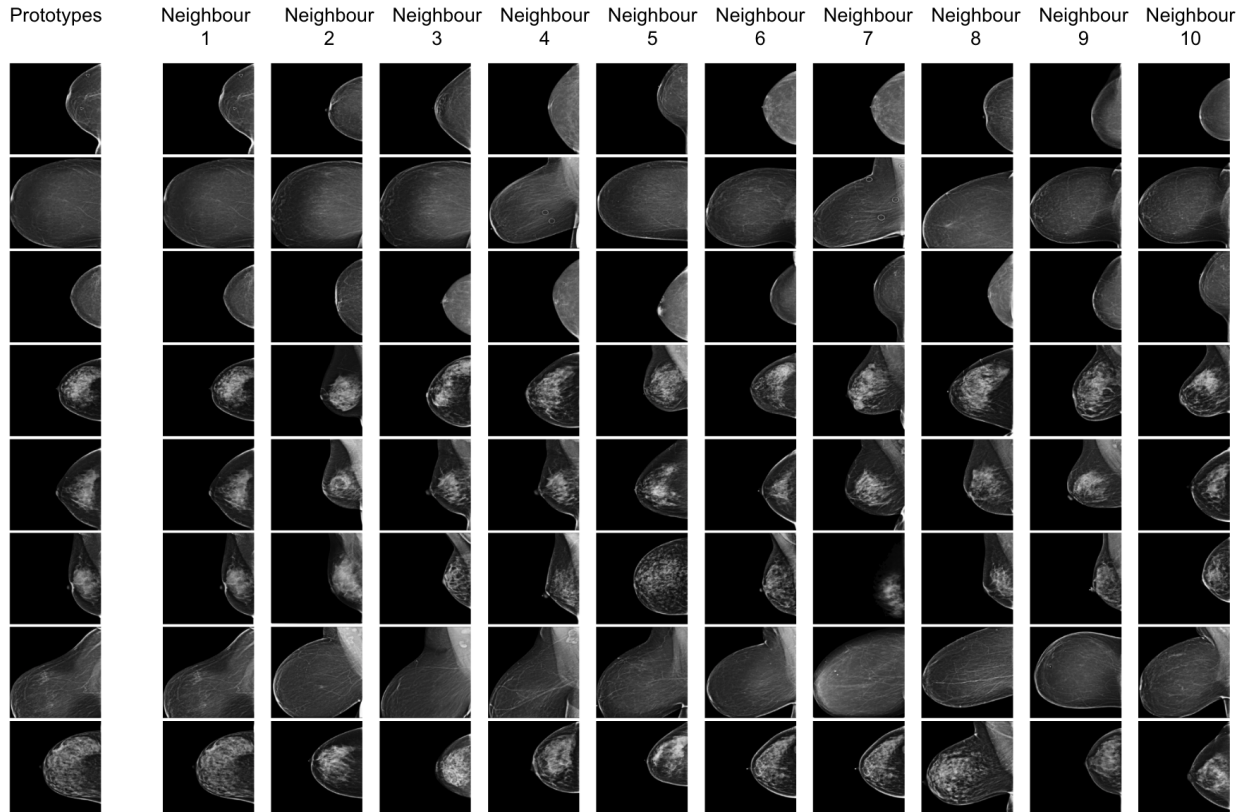


Figure 36: Neighbouring sample visualization for mammography prototypes learned in Section 3.5.4.

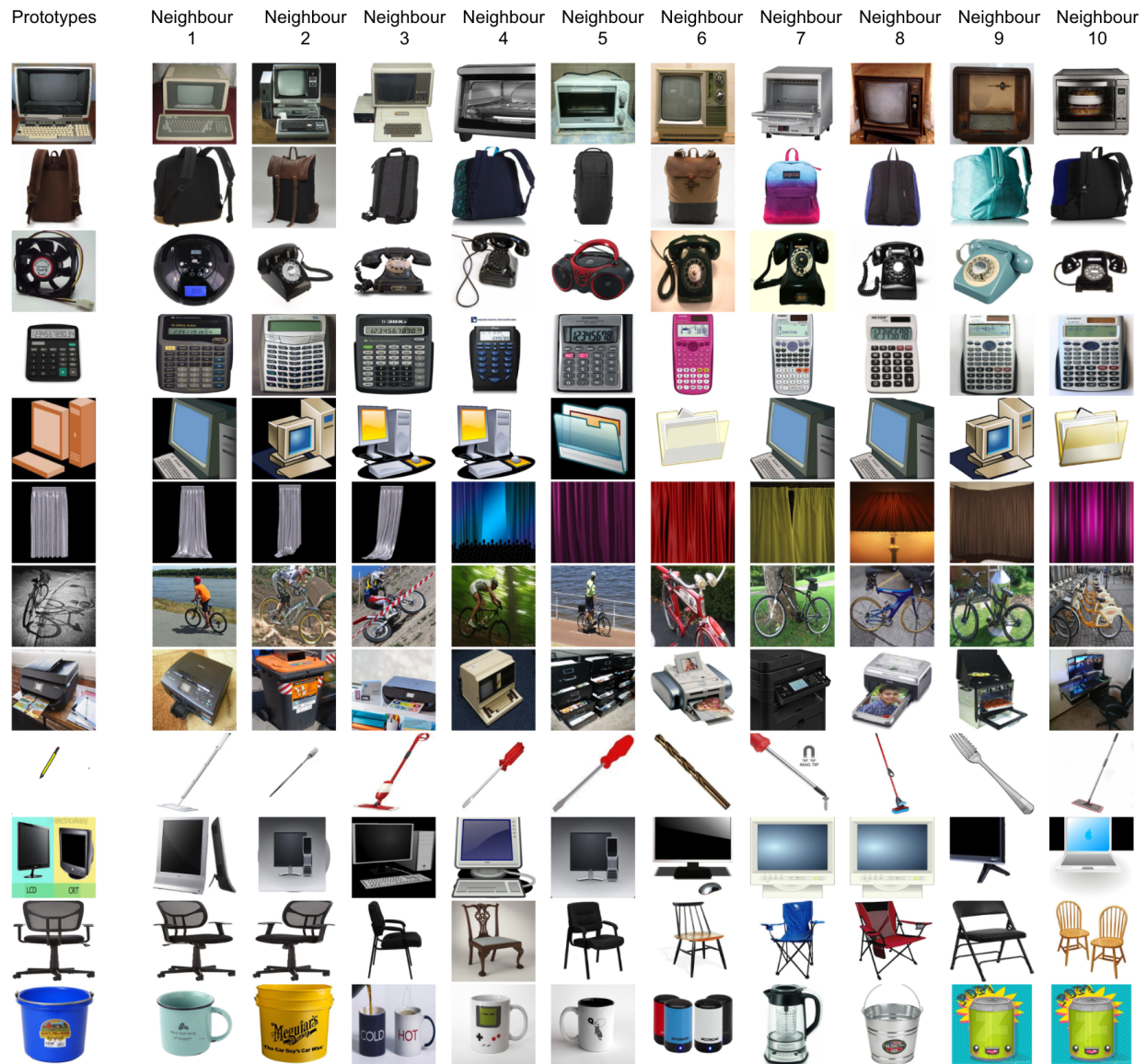
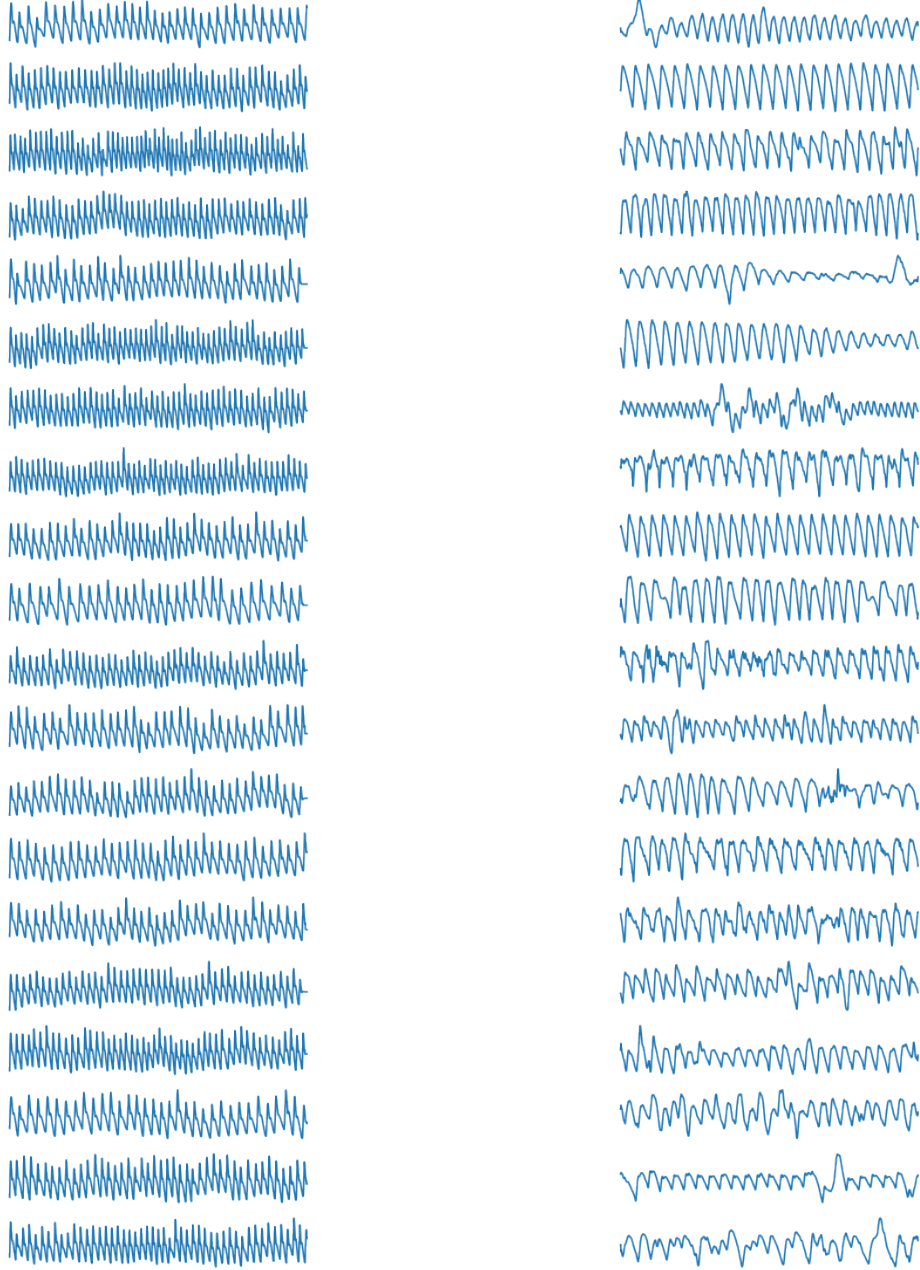


Figure 37: Neighbouring sample visualization for office home object prototypes learned in Section 3.5.5.



(a) Examples of synthetic PPG signals.

(b) Examples of real PPG signals.

Figure 38: Here we show 20 samples each of both real and synthetic PPG signals.

Appendix C. Evaluation of Influential Example Explanations

C.1 Influential Example Explanations: Alignment

We now evaluate influential example explanations by asking the question: *Given datasets \mathcal{D} and \mathcal{D}' and their respective models, are we choosing the right examples from either dataset to remove to remediate feature importance differences between the datasets?*

The first step in the explanation pipeline is training a discriminator to classify whether the local intrinsic feature importance (LiFIM) for an example originates from \mathcal{D} or \mathcal{D}' (see Algorithm 1). Then, by computing influences for LiFIM, we determine the appropriate examples that have the highest (positive) influence on the discriminator loss (i.e., removing them increases the loss). We first determine the validity of our method by computing the theoretical influences of each example in \mathcal{D} and \mathcal{D}' using Equation 4 and then empirically calculating the loss of the discriminator after the example is removed from training. This is shown in Figure 39 below. Here, the empirical estimates match the theoretically computed values for the influence. We remark here that these are results for logistic regression, whereas the correlation may not be as high for nonlinear functions, though other works have found high correlations (Koh and Liang, 2017).

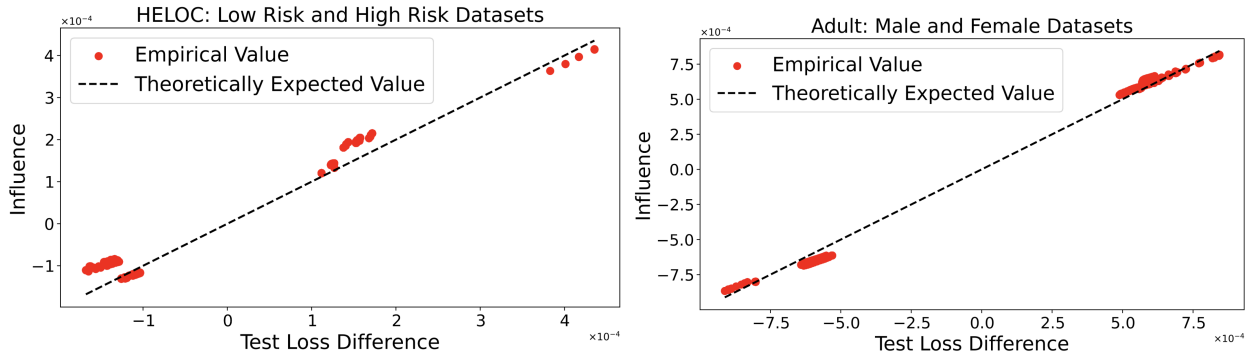


Figure 39: Theoretical Influence vs Empirical Change in discriminator loss for HELOC (left) and Adult (right) datasets. Each point corresponds to an example (represented by its feature importance vector) in the dataset $\mathcal{D} \cup \mathcal{D}'$. We compute the theoretical influence of each example and compare it with the empirical test loss obtained after removing the example from the training set.

In the next step, we want to see if the removed examples actually change the global feature importance measure when the task model is retrained.

- Without loss of generality, we choose the dataset we are removing examples from as \mathcal{D}' . For HELOC, this corresponds to the High-Risk dataset with `ExternalRiskEstimate` ≤ 70 . For Adult, this is the Female dataset.
- We remove a varying % of the most influential examples from \mathcal{D}' decided by Algorithm 1. Let S be the set of examples removed and $f_{\mathcal{D}'}$ and $f_{\mathcal{D}' \setminus S}$ the task specific models trained on \mathcal{D}' and $\mathcal{D}' \setminus S$ respectively.
- We then compute the average global feature importance alignment between the dataset \mathcal{D} and the datasets \mathcal{D}' and $f_{\mathcal{D}' \setminus S}$ respectively. Given GiFIMs $\phi_g(\mathcal{D})$, $\phi_g(\mathcal{D}')$, and $\phi_g(\mathcal{D}' \setminus S)$ for \mathcal{D} , \mathcal{D}' , and $\mathcal{D}' \setminus S$ respectively, alignment is defined as:

$$\text{Alignment} = \frac{\|\phi_g(\mathcal{D}) - \phi_g(\mathcal{D}')\| - \|\phi_g(\mathcal{D}) - \phi_g(\mathcal{D}' \setminus S)\|}{\|\phi_g(\mathcal{D}) - \phi_g(\mathcal{D}')\|} \quad (15)$$

or the % reduction in error between GiFIMs of \mathcal{D} and \mathcal{D}' once the influential examples are removed from consideration.

- For the task models, we experiment with decision trees of different depths to see if there is any impact on alignment. Figure 40 shows that removing a small number of examples can improve the alignment in feature importances of \mathcal{D} and \mathcal{D}' - however, there is a limit to this.

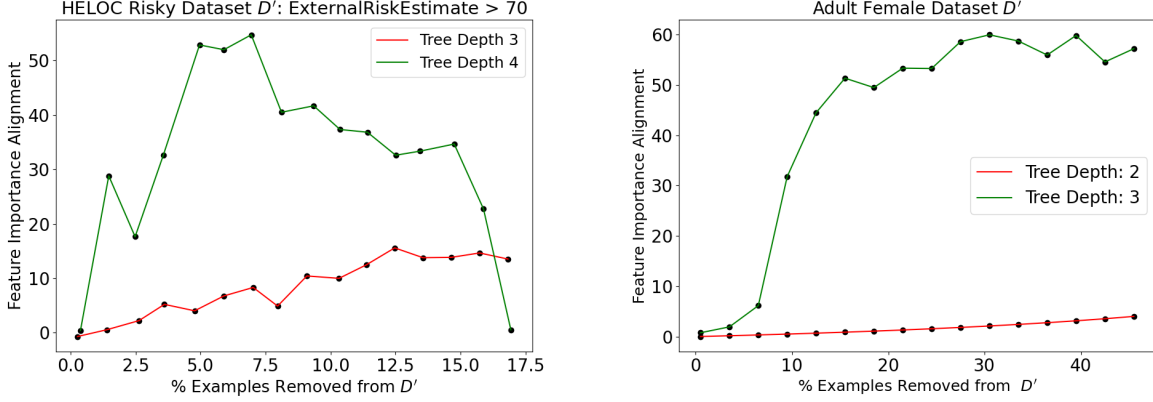


Figure 40: Feature Importance Alignment vs % Examples Removed from \mathcal{D}' for HELOC and Adult datasets. There are two distinct regimes here. These regimes correspond to removing all the positively influential points (which make the distributions different), and then when we run out of those points, we start to remove some additional points that are actually keeping the distributions similar. In the first regime, the number of examples removed is small relative to the dataset size. In this regime, increasing the number of examples removed causes an increasing alignment between the GiFIMs of \mathcal{D} and $\mathcal{D}' \setminus S$, thereby reducing the error. In the second regime, the number of examples removed is no longer insignificant relative to the dataset size. Here, in most cases, increasing the number of examples causes a plateau or reversal in alignment (i.e., the error increases). We hypothesize that this occurs because there are only a certain number of examples in a dataset with positive influence on the alignment. Once these examples have been removed, the remaining examples will have an increasingly negative influence on alignment.

C.2 Influential Example Explanations: Robustness

In this section, we evaluate whether our influential example explanations are sensitive to perturbations of the datasets. At a high level, we have two desiderata:

- The most influential examples exhibit similar influence scores across bootstrapped datasets. We show this in Figure 41.
- The ranking of the most influential examples is similar across bootstrapped datasets (i.e. the same originally influential examples remain influential). We show this in Figure 42.

To do this, we perform the following procedure:

- We first fix one of the datasets - call this \mathcal{D} .
- Next, we generate $N = 5$ bootstrapped datasets from \mathcal{D}' by sampling with replacement.
- For each bootstrapped dataset \mathcal{D}'_b , we compute the influential example explanation between \mathcal{D} and \mathcal{D}'_b . This involves storing the 50 most influential examples from \mathcal{D} using Algorithm 1 and recording their influence scores.

Let the set of the k most influential examples in \mathcal{D} explaining the difference in feature importances between \mathcal{D} and \mathcal{D}' be denoted as $E_{\mathcal{D}}(\mathcal{D}') = \{e_1, e_2, \dots, e_k\}$. For each bootstrapped dataset \mathcal{D}'_b , we now compute the Kendall-Tau ranking similarity between $E_{\mathcal{D}}(\mathcal{D}')$ and $E_{\mathcal{D}}(\mathcal{D}'_b)$, defined as:

$$K(E_1, E_2) = \frac{C - D}{\binom{n}{2}} \quad (16)$$

where C is the number of pairs in E_1, E_2 that have the same relative order in both rankings and D is the converse. A value of $K(E_1, E_2)$ close to 1 means that the ranking order is preserved, implying that the identified influential examples in \mathcal{D} are consistently recognized across different realizations of the dataset \mathcal{D}' . Figure 42 shows that the Kendall-Tau similarity values remain high (i.e. ~ 0.9975) across multiple bootstrap iterations for both Adult and HELOC datasets, indicating a robust identification of influential examples.

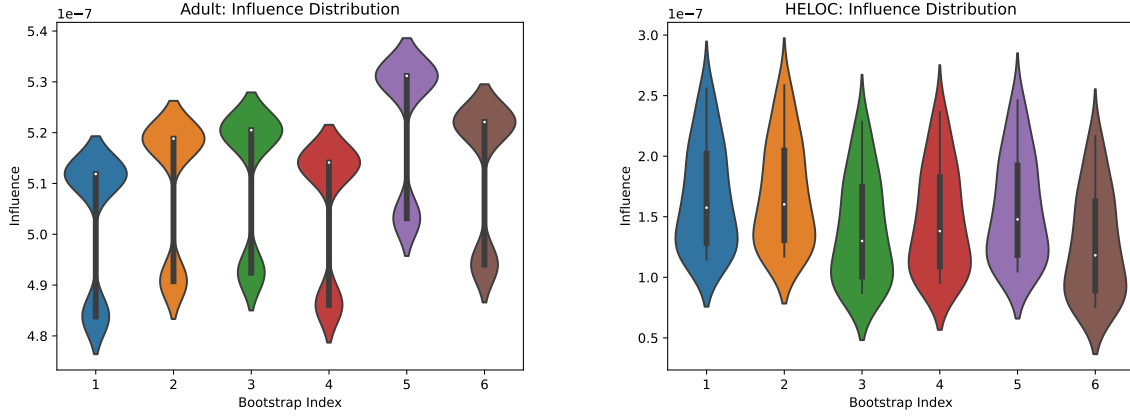


Figure 41: Influence distribution of 50 examples in the Adult and HELOC datasets with the highest influence scores. We see that this remains consistently stable across bootstraps, implying that our explanations will also remain stable.

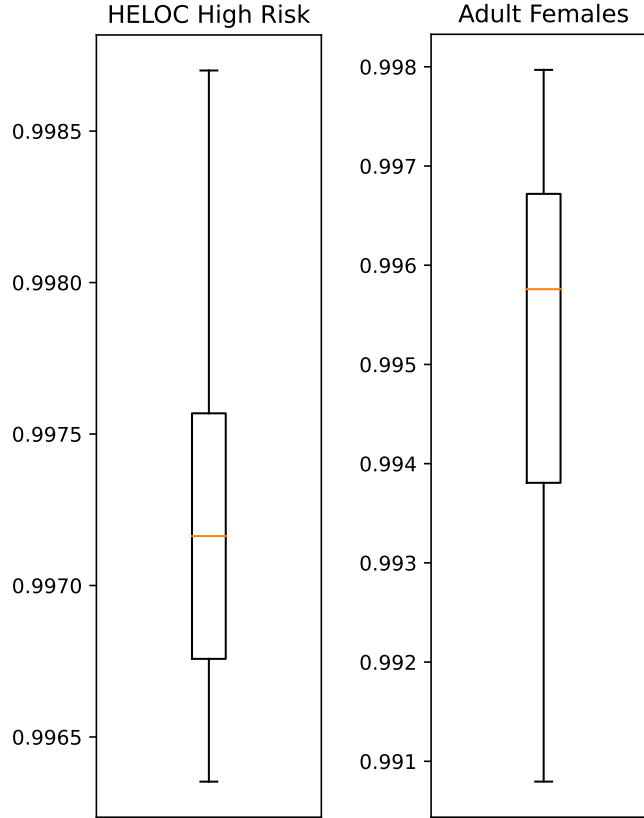


Figure 42: Kendall Tau Ranking Similarity of influential examples in \mathcal{D} across different bootstraps of \mathcal{D}' . For HELOC, we show how influential examples in the high-risk HELOC dataset (as defined as in Section 3.3.5) are affected as we bootstrap the low risk HELOC dataset. For Adult, we show how influential examples in the Female dataset are affected as we bootstrap the Male dataset. Ultimately, we see that the similarity across bootstraps is high, suggesting that the identified influential examples in \mathcal{D} are consistently recognized across different realizations of the dataset \mathcal{D}' . This shows that our explanations are robust to dataset perturbations.

C.3 Influential Example Explanations: Effect of Rashomon Set

We evaluate how our influential example explanations change as we change the size of the Rashomon set, i.e. the set of all near-optimal models. We can change the size of the Rashomon set using the ϵ parameter, which sets the maximum allowed sub-optimality gap for a model to enter the set. For a given epsilon value, we use the Rashomon Importance Distribution method from Donnelly et al. (2023) to generate intrinsic feature importances for \mathcal{D} and \mathcal{D}' . We then see how the resulting set of influential examples in either dataset changes as we vary the epsilon parameter. We also examine how long it takes to generate the explanation as we vary the size of the Rashomon set.

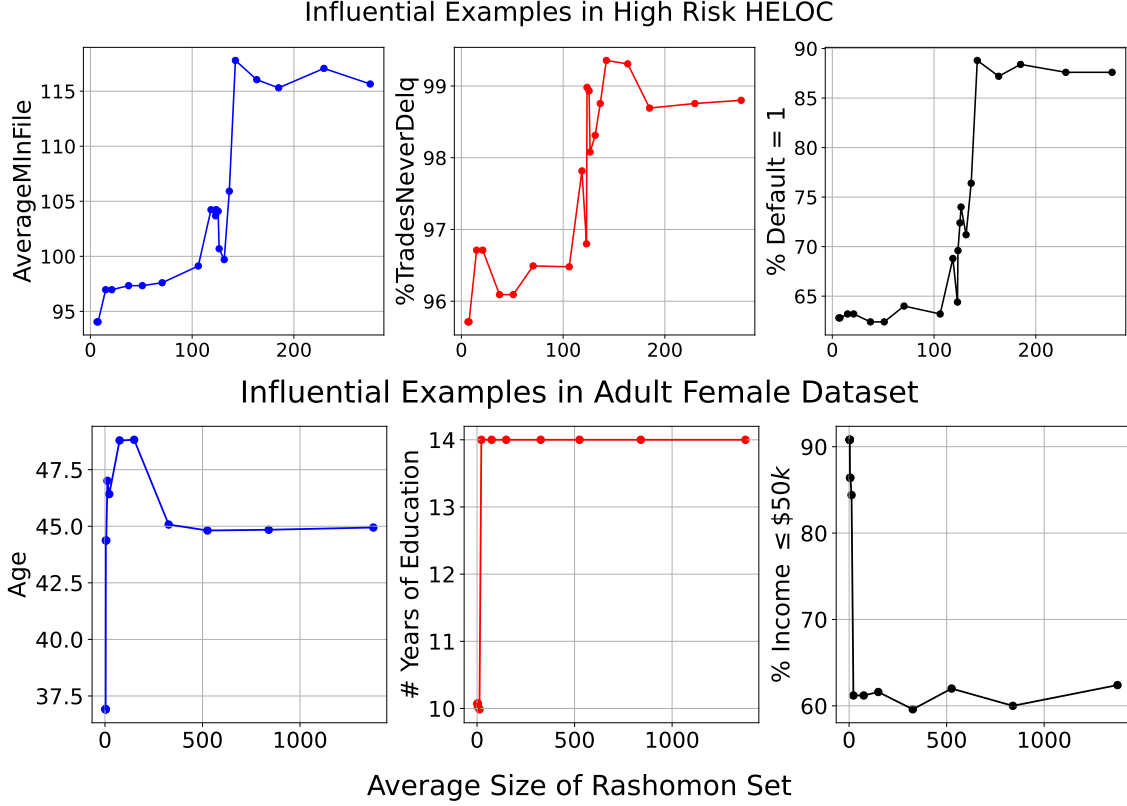


Figure 43: Properties of the 50 most influential examples in \mathcal{D} that explain differences between \mathcal{D} and \mathcal{D}' . As the Rashomon set size increases, the feature importances output by RID will stabilize, resulting in a corresponding stabilization in the influential examples output by our explanation.

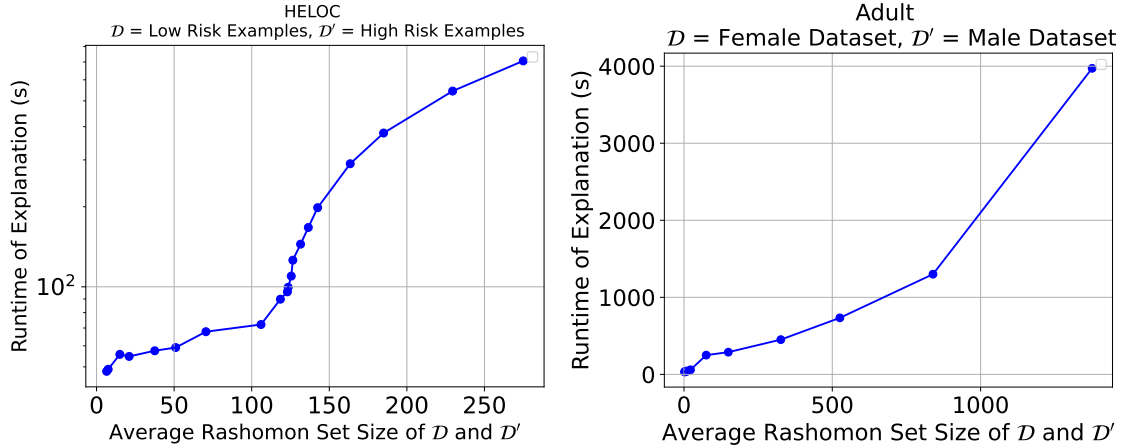


Figure 44: Size of the Rashomon set used to generate feature importances vs explanation runtime.

Appendix D. Illustrating a Failure Mode of Kulinski and Inouye (2023)

D.1 Methodology

We now illustrate an example where the distribution shift explanation of Kulinski and Inouye (2023) is incoherent, but our prototype-based explanations are able to accurately capture dataset differences. In particular, we simulate the mean shift of a mixture of Gaussians – Case 1 below is the same setup as Kulinski and Inouye (2023). Because the cluster centres are shifted by the same amount, we call the cluster centres of X and Y *paired*.

Case 1:

- We first sample $k = 6$ points uniformly from the circumference of a circle of given radius $r_x = 10$. These points are the cluster centres of a mixture of Gaussians with isotropic covariances and equal cluster proportions. We sample 60 points around each cluster centre. Call this resulting dataset of 360 points X .
- We then repeat this procedure, but with a circle radius of $r_y = 20$. Call this resulting dataset Y .

Case 2:

- Dataset X is generated in the same manner with the same parameters as above.
- We then sample cluster centres with a circle radius of $r_y = 20$. The resulting mixture of Gaussians still has isotropic covariances, but we now change the cluster proportions. To generate cluster proportions, we sample a 6 dimensional probability vector from a Dirichlet distribution with parameters $\alpha_1 = \dots = \alpha_6 = 1$. This is a distribution with the following pdf:

$$f(x_1, \dots, x_6; \boldsymbol{\alpha}) = \frac{1}{\beta(\boldsymbol{\alpha})} \prod_{i=1}^6 x_i^{\alpha_i - 1} \quad (17)$$

with $\beta(\boldsymbol{\alpha})$ is the Beta function serving as the normalizing constant. This setup ensures that the generated vector satisfies $\sum_{i=1}^6 x_i = 1$. We then sample from the mixture of Gaussians according to these cluster proportions, generating 360 points. Call this dataset Y .

Explanation computed by Kulinski and Inouye (2023): The explanation finds k clusters in X and illustrates how they shift from X to Y using an optimal transport formulation. The final output is the shifted cluster center and the map from X to Y - this is illustrated in the Figures below.

D.2 Case 1: Our explanation and Kulinski and Inouye (2023) is coherent

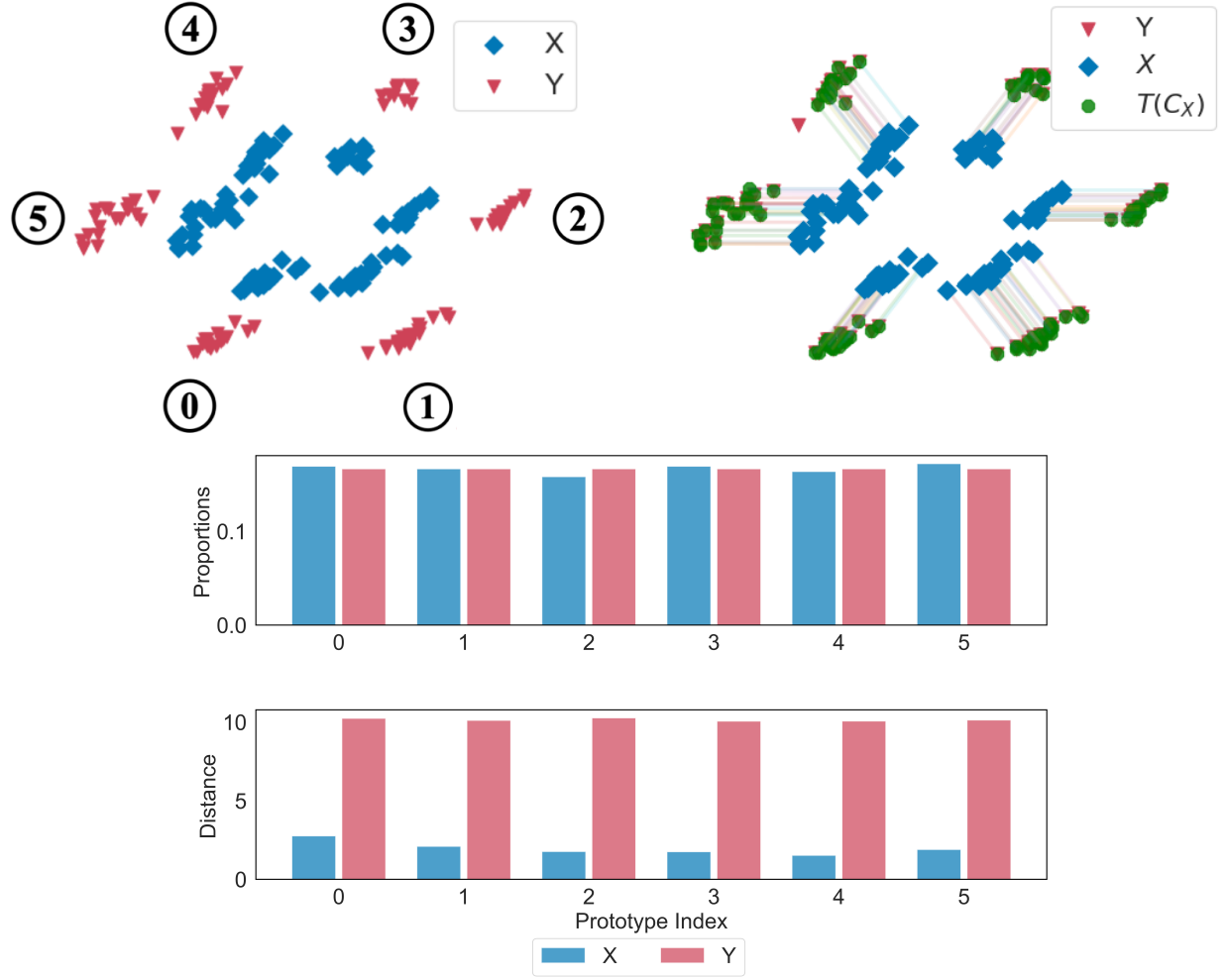


Figure 45: Top left - The two datasets X and Y have clusters with equal proportions. The paired cluster centres are labelled appropriately.

Top right - The explanation map showing the shift of each cluster in X to the corresponding cluster in Y computed by Kulinski and Inouye (2023). Their method almost perfectly maps each point from X to Y and the explanation is coherent.

Top bar plot - (Our Explanation) Proportion of points in the neighborhood of each prototype (cluster centre) in X for both datasets. The proportions are the same for all clusters.

Bottom bar plot - (Our Explanation) Average distance of points to the closest prototype (cluster centre) in X for both datasets. This is constantly high for dataset Y , suggesting a constant shift in clusters has occurred.

D.3 Case 2: Our explanation is coherent but Kulinski and Inouye (2023) is not

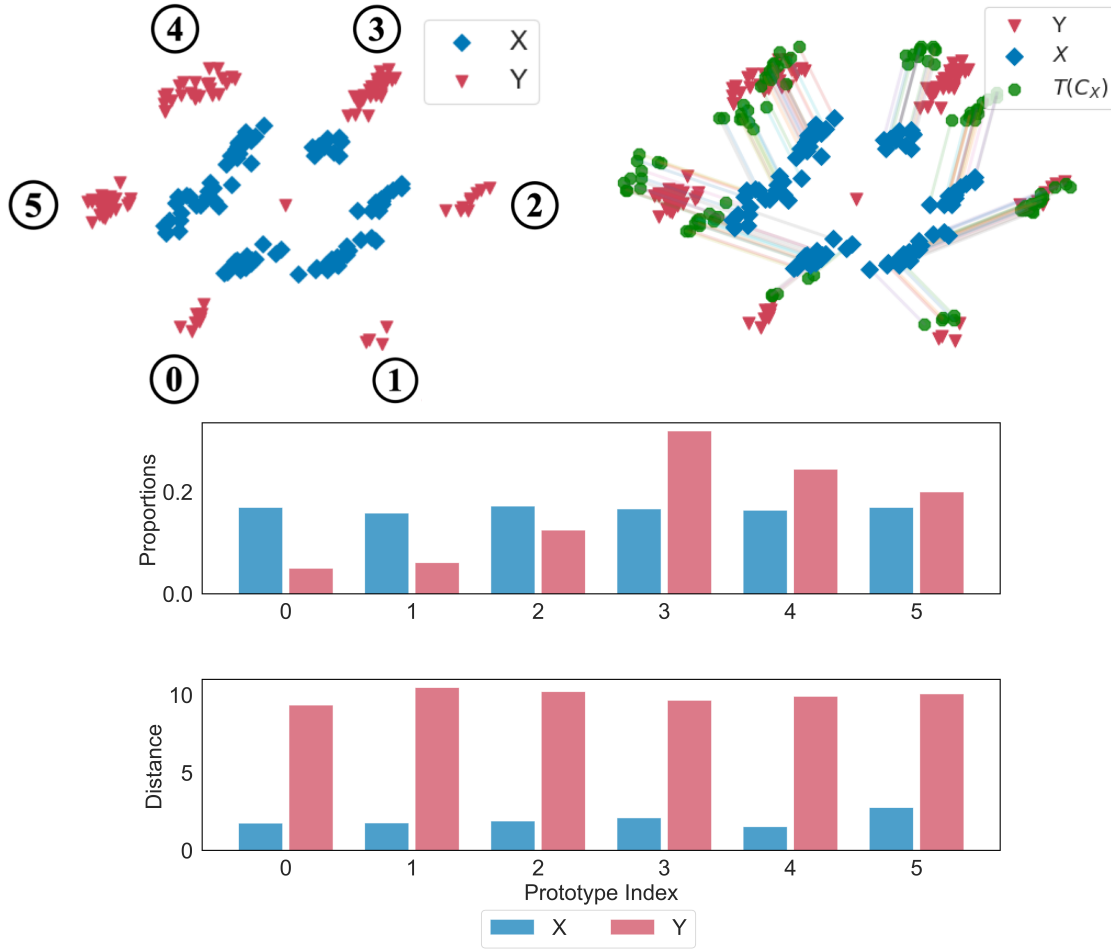


Figure 46: Top left - The two datasets X and Y have the same cluster centres as in Figure 45, but Y has unequal cluster proportions sampled using the methodology above. The paired cluster centres are labeled appropriately.

Top right - The explanation from Kulinski and Inouye (2023) is now unable to map the shift in clusters from X to Y as there is no longer a one-to-one mapping between points in the clusters. This renders the explanation uninterpretable.

Top bar plot - (Our Explanation) Proportion of points in the neighborhood of each prototype (cluster centre) in X for both datasets. The proportions are now different across all clusters and can be visually validated from the diagram, which is what we want to see from our explanation.

Bottom bar plot - (Our Explanation) Average distance of points closest prototype (cluster centre) in X for both datasets. This is constantly high for dataset Y , suggesting only a constant shift in cluster centres has occurred.

Our prototype-based explanations can, therefore, quantify exactly how cluster proportions and distances have changed.

Appendix E. Text Data Difference Analysis with “*Exact Counts*”

In this section, we briefly consider the idea of explaining two text datasets. We consider this to be a significantly more challenging (and general) problem compared to other modalities, owing to various pathologies exhibited by text data. In particular, prototype based methods fail here because it is hard to define an underlying latent space where distances between textual examples make sense (e.g., what does the ‘distance’ between a sentence prototype and another example mean?). Thus, pending future work, we provide a simple method to summarize two text datasets by extracting interpretable attributes from the datasets. Meanwhile,

we consider that the approach in Elazar et al. (2023) to be a useful starting point for understanding a given text dataset (and indeed drawing points of comparison with other similar datasets).

Dataset \mathcal{D} and \mathcal{D}' We first define datasets \mathcal{D} and \mathcal{D}' . In this example, we used the HC-3-English dataset (Guo et al., 2023) as a sample dataset for text dataset difference explanations. HC-3-English contains answers to questions from both humans (\mathcal{D}) and ChatGPT (2022 version) (\mathcal{D}') collected from several domains and tasks, including open-domain question-answering (QA), financial, medical, legal, and psychological areas. Human and ChatGPT answers are not compared pairwise (i.e., we did not assume a one-to-one mapping between the answers). Instead, we analyzed the differences between all human answers against all ChatGPT answers. We hypothesize that the datasets can be meaningfully compared using the following attributes:

1. Have consistent writing structure
2. Use formal language
3. Have a neutral tone
4. Show subjective opinion
5. Use of technical references

These attributes can fairly reliably be determined by querying language models.

For each input X in \mathcal{D} and \mathcal{D}' , we query GPT-3.5 Turbo using the following prompt:

Prompt:

Analyze the following text by answering the following questions including:

$\{A_1, A_2, \dots, A_n\}$

For each question provide "YES OR NO" answer only.

X_i

Where each A_i is an attribute.





Forming the explanation Using the prompt template defined in above, we show two query-answer examples in Figure 47. After answers were collected from ChatGPT for all samples in the dataset, we formed explanations based on the results shown in Table 5 – on these question-answering tasks, among the attributes analyzed, humans and ChatGPT answers mainly differ in *use of formal language usage*, *use of subjective opinion* and *writing structure consistency*. That is, we found that humans tend to show more subjectivity in their answers and tend to use more informal language than ChatGPT. Also, humans are much less likely to use consistent writing structure throughout each writing sample.

A simple logistic regression model was able to reach 85% accuracy in predicting whether the text is produced by humans or ChatGPT with the five attributes we collected for each text sample. This indicates that the selected attributes are often able to correctly identify differentiating attributes between the two datasets.

	Have consistent writing structure	Use formal language	Have a neutral tone	Show subjective opinion	Use of technical references
Human	59.4%	22.6%	77.5%	26.9%	35.0%
ChatGPT	99.2%	93.2%	99.6%	0.9%	35.2%
ChatGPT (Informal)	69.9%	10.1%	68.9%	17.5%	18.3%

Table 5: Human and ChatGPT answer analysis results.

Improving \mathcal{D}' Following the conclusions that we have reached, to improve the similarity between \mathcal{D} and \mathcal{D}' , we aimed to make each sample in \mathcal{D}' appear more informal and colloquial. Here we again leveraged GPT-3.5 Turbo to “humanize” the ChatGPT text samples. For each text X_i in \mathcal{D}' , we used the following prompt:

 <p>Analyze the following text by answering the following questions including: "Does it have consistent writing structure?" "Does it use formal language?" "Does it have a neutral tone?" "Does it show subjective opinion?" "Does it use of technical references?". For each question provide "YES OR NO" answer only.</p> <p>water is a polar molecule , when pure is odorless . When it is humid there is a higher density of water vapor in the air . During this time , other polar molecules , which may have a smell , bond to the water vapor which is inhaled . Hence the wet smell .</p>  <p>"Does it have consistent writing structure?" - YES "Does it use formal language?" - NO "Does it have a neutral tone?" - YES "Does it show subjective opinion?" - NO "Does it use technical references?" - YES</p>	 <p>Analyze the following text by answering the following questions including: "Does it have consistent writing structure?" "Does it use formal language?" "Does it have a neutral tone?" "Does it show subjective opinion?" "Does it use of technical references?". For each question provide "YES OR NO" answer only.</p> <p>"What a Wonderful World" is a song written by Bob Thiele and George David Weiss. It was first recorded and made popular by Louis Armstrong in 1967. The song has since been covered by many artists, including Eva Cassidy, Israel Kamakawiwo'ole, and Sam Cooke.</p>  <p>"Does it have consistent writing structure?" - YES "Does it use formal language?" - YES "Does it have a neutral tone?" - YES "Does it show subjective opinion?" - NO "Does it use technical references?" - NO</p>
--	--

(a) Human answer analysis.

(b) ChatGPT answer analysis.

Figure 47: Human and ChatGPT answer analysis example**Prompt:**

Make the following context sound less formal, paraphrase using some colloquial language.

The attribute calculated for the improved \mathcal{D}' is shown in the last row of Table 5 (ChatGPT (informal)). Using the attributes, on the same test split, the same logistic regression classifier was only able to achieve 45% accuracy. The decrease in the model's predictive performance indicates an increase in similarity between the text samples in \mathcal{D} and \mathcal{D}' .

Results By utilizing these human-interpretable attributes and a straightforward counting method, we effectively identified key differences between human- and machine-generated text. Building on this understanding, we quantitatively improved the similarity between the two by addressing these differences. Without this dataset-level explanation, achieving this result would have been both time-consuming and subject to considerable uncertainty.