BACHELOR THESIS

Translating Explanations to Further Understanding of the Disagreement Problem in Explainable Machine Learning

written and submitted by

Ellen Kunigk
born June 7, 2001 in Pinneberg
Matriculation number: 3450419

1st Examiner:	Dr. Florian Seiffarth University of Bonn
2nd Examiner:	Prof. Dr. Christian Bauckhage University of Bonn
Supervisor(s):	Sebastian Müller University of Bonn

August 28, 2024

Abstract

The use of machine learning models often raises concerns as they operate as black-boxes and do not explain how they reach their predictions. Attribution methods potentially give insights into the decision making process of the models by providing post-hoc explanations. However, these explanations often vary strongly among methods, resulting in the Disagreement Problem. As the lack of agreement causes distrust in employing attribution methods, the need for further investigation of the nature of the Disagreement Problem is apparent.

This thesis quantitatively examines structural aspects of the Disagreement Problem with a focus on the underlying concepts of gradient- and perturbation-based methods. Initially, we explore numerical similarities and separability of explanations produced by these methods. We then investigate how the shared conceptual foundation of these methods influences the success of translating explanations from one method to another. Additionally, we assess whether masking less informative features enhances the ease of translation by reducing the amount of information processed.

Our findings show that the explanations are separable and explanations produced by gradient- or perturbation-based methods often show similar characteristics. We do find a method pair whose explanations are closely aligned in characteristics, that shares a concept, and that consistently performs best in all translations. Yet, simply sharing a broad gradient- or perturbation-based approach does not suffice to always assume successful translations. Masking parts of the translations does not regularly assist translations.

Statement of Authorship

I hereby confirm that the work presented in this bachelor thesis has been performed and interpreted solely by myself except where explicitly identified to the contrary. I declare that I have used no other sources and aids than those indicated. This work has not been submitted elsewhere in any other form for the fulfillment of any other degree or qualification.

Bonn, August 28, 2024

Contents

1 Introduction **1**

2 Related Work **3**
2.1 Attribution Methods 3
2.2 Disagreement Problem 3
2.3 Approaches to Improve Understanding 4

3 Definitions and Methods **5**
3.1 Attribution Methods 5
3.1.1 Gradient-based Methods 5
3.1.2 Perturbation-based Methods 7
3.2 Metrics 8
3.2.1 Feature Agreement 8
3.2.2 UMAP 8
3.3 Autoencoder 9

4 Experimental Setup **11**
4.1 Datasets and Models 11
4.2 Explanation Set 11
4.2.1 Explanations 11
4.2.2 Preprocessing 12
4.3 Preliminary Analysis 12
4.3.1 Numerical Distribution 12
4.3.2 Feature Agreement 13
4.3.3 Structural Analysis 13
4.4 Translation Analysis 14
4.4.1 Translations 14
4.4.2 Rankings 15
4.4.3 Translations over Subsets of Features 16

5 Preliminary Analysis **17**
5.1 Numerical Distribution 17
5.2 Feature Agreement 20
5.3 Structural Analysis 21
5.3.1 UMAP Results 21

5.3.2 Classification Results . . . 22
5.4 Discussion . . . 22
5.5 Summary . . . 24

6 Translation Analysis **27**
6.1 Translations . . . 27
6.2 Rankings . . . 30
6.3 Translations over Subsets of Features . . . 33
6.4 Discussion . . . 36
6.4.1 Translation Discussion . . . 36
6.4.2 Ranking Discussion . . . 37
6.4.3 Discussion of Translations over Subsets of Features . . . 37
6.5 Summary . . . 38

7 Conclusion **41**

Bibliography **43**

A Additional Results **49**

Chapter 1

Introduction

Machine learning models often operate as black-box models giving no explanations on how they reach their predictions. To develop trust in their use, the need for transparency has become apparent (Ali et al., 2023). Attribution methods are often seen as the greatest hope for resolving the issue of explainability (Watson, Hasan, and Moubayed, 2021). These methods intend to give post-hoc insights into complex models.

Current research sees a variety of methods of different approaches emerging. These methods are often separated into gradient- and perturbation-based methods. However, they frequently produce explanations for the local model predictions that are not alike or even contradictory to one another. Krishna et al. (2022) introduced the term *Disagreement Problem* describing exactly this phenomenon in which attribution methods outputs disagree with each other. They developed a multitude of metrics to evaluate disagreement among methods in an attempt to quantify how explanations from different methods relate to each other.

Krishna et al. (2022) additionally showed that it remains unclear how to handle the disagreement, and it is not fully understood to this moment. Trustworthy machine learning systems require consistent explanations for the decisions of the included models. There is a need to identify common ground among attribution methods and allow for deeper investigation of the Disagreement Problem in current research. Engaging in understanding of state-of-the-art methods is necessary to advance machine learning use and to assist novel method development (Jesus et al., 2021; Cuzzocrea et al., 2023; Bhardwaj and Parashar, 2024).

In this work, we intend to improve understanding of the Disagreement Problem. We assume that all methods have the ability to extract some information about the machine learning model's inner workings. Therefore, we aim to investigate the overlap and differences in this information among methods. We focus on separating them into gradient-based and perturbation-based methods to see how a shared approach affects the similarity of the explanations. It is possible that the methods are attentive to different aspects of the underlying model. We employ a novel approach, as we translate from explanations of one method into those of another method. By evaluating the success of the translation, we intend to learn about the structural relationships between the attribution methods and to add some system to the Disagreement Problem. Additionally, we set out to learn what happens when we mask parts of the explanations and leave the translator to respond to a reduced subset of features.

We state the following hypotheses.

- H1 The translation of one explanation into another is more successful when both methods align in their underlying concepts of being gradient-based or perturbation-based.
- H2 It is easier to focus the translation on the features with the highest attribution scores of each method than a translation over the complete range of features.

To ensure a foundation for our hypotheses, we first conduct a preliminary analysis of the sets of explanations from different methods. We inspect numerical characteristics, feature agreement, and perform structural tests. We proceed to conduct translations for all pairs of methods in both directions. We translate linearly or via an autoencoder. Additionally, we create rankings based on the translation results. We repeat the process, but mask either one or two thirds of features with the lowest attribution scores, so that higher scores remain.

In line with our thesis, Müller et al. (2023) conducted a thorough analysis on method disagreement using similarity metrics on the same methods we use. Koenen and Wright (2024) aimed to improve understanding of the Disagreement Problem by investigating the fundamentals and the distribution of explanations among groups of methods.

Our preliminary findings show separability of explanations from different methods and shared characteristics among methods from aligned concepts. When translating, we find that sharing a gradient- or perturbation-based approach is not sufficient to predict successful translations. However, task-dependent, close alignment of concepts can result in similar structures of explanations and ease of translation as two gradient-based methods consistently performed best in translations. Masking one or two thirds of the features does not regularly help translation success.

We begin, in Chapter 2, by elaborating on related work. In Chapter 3 we define methods and metrics that we employ later. Chapter 4 presents the experimental setup of this thesis, while the following two chapters present the results from the preliminary analysis and the translations and discuss them in relation to our hypotheses. Lastly, we present our conclusion and possible future work in Chapter 7.

The code related to the thesis can be found here: https://github.com/ekunigk/BA_Disagreement_Problem.

Chapter 2

Related Work

In this section, we present previous work related to the attribution methods we observe and the Disagreement Problem.

2.1 Attribution Methods

From the growing need to increase trust in machine learning models, a variety of attribution methods have been developed in recent years. We focus on local post-hoc methods and categorize them as gradient- and perturbation-based methods. Gradient-based methods evolved from simple application of the gradients of the neural network as importance scores, as in VanillaGrad (Simonyan, Vedaldi, and Zisserman, 2013), to more complex methodologies. Popular representatives are SmoothGrad (Smilkov et al., 2017) and Integrated Gradients (Sundararajan, Taly, and Yan, 2017). Smilkov et al. (2017) introduced adding noise to the instances being explained and averaging the gradients over multiple noisy versions to produce smoothed-out attribution scores. Sundararajan, Taly, and Yan (2017) utilized a comparison to a baseline and integrating over gradients along the path from baseline to instance to explain a machine learning model's prediction.

Among the most frequently employed perturbation-based methods are KernelSHAP (Chen et al., 2022) and LIME (Ribeiro, Singh, and Guestrin, 2016). While Chen et al. (2022) used an approach from game theory to estimate the features' contribution in KernelSHAP, Ribeiro, Singh, and Guestrin (2016) used interpretable models to simulate behavior in the local neighborhood.

2.2 Disagreement Problem

Krishna et al. (2022) first coined the term Disagreement Problem. They applied a variety of metrics, like feature or rank agreement, to the explanations of different methods to show that they disagree with each other. Through their analysis and quantitative framework, as well as a survey with many machine learning practitioners, they extended understanding of the missing alignment between attribution methods' output.

Even before, Neely et al. (2021) found that none of the methods they observed agreed on the ranks that they ascribed certain features. They argued that this disagreement, however, does not define the validity of the methods.

In addition, Koenen and Wright (2024) concluded that a decision on which method is best employed is highly task- and data-dependent. They highlighted difficulties in choosing appropriate evaluation metrics and employing them.

Several works even voiced concerns, that the disagreement problem could have severe consequences when parties can freely choose the explanation supporting their own position in conflictual situations (Bordt et al., 2022; Aïvodji et al., 2019). At a minimum, the disagreement among methods and their missing alignment is reason to be doubtful if employing the attribution methods holds any benefit (Zhou et al., 2021; Cuzzocrea et al., 2023).

2.3 Approaches to Improve Understanding

Previous work has explored attribution methods and the disagreement among them, sometimes even keeping their underlying concept in mind. On the one hand, some analyses attempted the unification of attribution methods (Han, Srinivas, and Lakkaraju, 2022; Ancona et al., 2017), while on the other hand, others showed division among the methods (Cuzzocrea et al., 2023). Ancona et al. (2017) found a strong relation between four methods based on model gradients. Cuzzocrea et al. (2023) and their work indicated, that LIME and SHAP are missing consistency and other desirable qualities.

Prior work has evaluated both categories of attribution methods against each other. Alvarez-Melis and Jaakkola (2018) found both groups to lack stability, but perturbation-based methods even more than those based on gradients. Koenen and Wright (2024) highlighted the influence of preprocessing, hyperparameters and evaluation metrics on the results of analyses with attribution methods, while contributing to the understanding of the disagreement problem.

Müller et al. (2023) assessed the same five attribution methods that we study. They found their rankings based on a variety of similarity metrics to not be consistent. Krishna et al. (2022) employed four gradient-based and two perturbation-based methods and found inconsistent agreement among them. The attribution methods based on gradients separated into some pairs agreeing and some disagreeing with one another. The authors stated that it would be interesting to study the reasons for the disagreement problem in a more systematic way (Krishna et al., 2022). Bordt et al. (2022) share this view, claiming that the disagreement problem is simply not sufficiently explored.

Chapter 3

Definitions and Methods

3.1 Attribution Methods

Attribution methods are at the center of this analysis. They are attributing a model's prediction back to the input features. A model, in our case denoted as $f : \mathbb{R}^n \rightarrow [0, 1]$, takes an input $x \in \mathbb{R}^n$ and produces a prediction $f(x) = y \in [0, 1]$. An attribution method utilizes input x, output y and model f to gain insight into the model's inner workings and to construct an attribution vector $A = (a_1, ..., a_n) \in \mathbb{R}^n$. a_i indicates the feature importance of feature x_i (Sundararajan, Taly, and Yan, 2017).

The methods often compare the input to a baseline vector x_0. The choice of baseline could be any vector, however, for the implementation of the following attribution methods the zero baseline $x_0 = (0, 0, ..., 0)$ with $|x_0| = n$ is used.

A variety of attribution methods have been developed. We will look at five different methods while categorizing them into gradient-based and perturbation-based methods (Krishna et al., 2022).

3.1.1 Gradient-based Methods

Attribution methods that are gradient-based are model-specific, as they require access to a given model f and its gradients. The gradients measure the change that all model weights undergo in combination with the change in the error of the prediction. Gradients can be somewhat compared to the model coefficients in a simpler model.

VanillaGrad, SmoothGrad and Integrated Gradients all make use of the model's gradients to compute attribution scores for a given input, but they take varying approaches.

VanillaGrad

The essence of the VanillaGrad attribution method lies in computing the gradients of the output of the model with regard to the input features. It does this by taking the derivative of the error with respect to each input feature. The computed gradients are indicators of how sensitive the model is to changes in the input for each feature. If the value of a gradient is high, it is a sign for a feature that is highly important for the predicted output.

$$A(x) = \frac{\partial f(x)}{\partial x} \tag{3.1}$$

VanillaGrad is a fairly simple method and convinces with ease of implementation. However, it can be noisy and does not account for potential sharp fluctuations in the derivatives on a small scale (Simonyan, Vedaldi, and Zisserman, 2013).

SmoothGrad

SmoothGrad works with the same foundation as VanillaGrad. It takes gradients of the model regarding the input x and output y but adds more steps to the computation of the attribution scores in an attempt to reduce the noise.

In addition to the input x, multiple noisy versions of x are created by adding Gaussian $\mathcal{N}(0, \sigma^2)$ or similar noise. For each of these samples, SmoothGrad computes the gradients and subsequently calculates the average over all the versions.

$$A(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial f(x + \mathcal{N}(0, \sigma^2))}{\partial x} \tag{3.2}$$

This methodology strives for smoothed out attribution scores. The input is represented in a more stable way. Enhancing the strategy of VanillaGrad, SmoothGrad is more resistant to local fluctuations in the gradients which have no meaning on a larger scale. However, the denoising effect comes at a computational cost. It requires many more computations of predictions and gradients due to the several versions of the input (Smilkov et al., 2017).

Integrated Gradients

The third of the gradient-based method takes a slightly different approach. The computation of Integrated Gradients requires a baseline $x_0 \in \mathbb{R}^n$ and considers the straightline path from baseline to input. The method then computes the gradients at all points along the path and cumulate them. Finally the cumulated gradients are multiplied with the distance from baseline x_0 to input x.

$$a_i(x) = (x_i - x_{0,i}) \int_{\alpha=0}^{1} \frac{\partial f(x_0 + \alpha(x - x_0))}{\partial x_i} d\alpha \tag{3.3}$$

This integral of the gradients along the path can be approximated with a sum of several points at small intervals along the path.

The attribution method Integrated Gradients has a strong theoretical base. The authors work with an axiomatic approach to ensure that the method works as intended. Attributions produced by Integrated Gradients consider the entire path from baseline to input, which makes them more reliable but also more computationally expensive. The choice of baseline is not trivial either (Sundararajan, Taly, and Yan, 2017).

3.1.2 Perturbation-based Methods

Perturbation-based methods are model-agnostic, meaning they can be applied to any model, without knowing anything about its internals. KernelSHAP and LIME both rely on perturbing the input x and observing the change in model output to produce explanations.

LIME fits an interpretable model to the neighborhood of the instance, while KernelSHAP observes model output on different subsets of features.

LIME

The explanations produced by LIME are presented as the coefficients of a model $g \in G$, G being a class of potentially interpretable models. g aims to approximate the local behavior of f.

For a data point x, the algorithm draws an instance z around x multiple times. z is weighted by its proximity to x by the proximity measure π_x. Then, LIME retrieves the prediction $f(z)$ of the original model f for these instances. The aforementioned perturbed instances are built by replacing some non-zero elements of x with elements from sample z to create z'. In the next step, the interpretable model g is fit to the perturbed samples.

To produce the feature attributions, LIME minimizes a loss function $\mathcal{L}$ that weights the perturbed samples according to their distance to x and takes into account the difference between the original model output and the prediction of g as follows to approximate the behavior of the original model f in this neighborhood.

$$A(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{3.4}$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z)(f(z) - g(z'))^2 \tag{3.5}$$

The coefficients of the interpretable model g present the feature attributions for the model f.

LIME provides clear and interpretable explanations and is easy to implement. The quality of the explanations is sensitive to the perturbations that are done and the characteristics of the local neighborhood. As not all features might be important in the locality of the instance, LIME is rather known for sparse attributions and grasping a few important features well (Ribeiro, Singh, and Guestrin, 2016).

KernelSHAP

The Shapley Value is a concept from game theory to allocate credit to players in games with coalitions. For KernelSHAP the ideas from game theory have been transferred onto the machine learning model f which is to be explained. Therefore, KernelSHAP is of strong theoretical foundation, as Shapley Value theory is based on axioms.

For each subset of present features S of all features D, KernelSHAP evaluates the model's prediction. Absent features, not in S, are replaced with baseline values from x_0.

The weighting kernel, $W(S)$, weights the evaluations based on subset size. Additionally, KernelSHAP includes an additive game $u(S)$ that sums coefficients for all present features and the evaluation $v(S)$ of the model on the perturbed sample according to the current subset.

$$A(x) = \arg\min_{\beta} \sum_{S \subseteq D} W(S)(u(S) - v(S))^2 \tag{3.6}$$

$$u(S) = \beta_0 + \sum_{i \in S} \beta_i \tag{3.7}$$

$$v(S) = f(x, x_{0,\overline{S}}) \tag{3.8}$$

KernelSHAP calculates the marginal contribution of each feature by evaluation with and without its presence. It fairly distributes the prediction among the features, considering all feature combinations, which adds compuational complexity. The choice of zero baseline is however simple to implement (Chen et al., 2022).

3.2 Metrics

In the following section, we introduce metrics to analyze our explanations.

3.2.1 Feature Agreement

Krishna et al. (2022) investigated what constitutes disagreement between two explanations and how they compare to each other. Machine learning practitioners, that participated in their survey, established as an essential notion of explanation agreement that the k features with the highest attribution scores align. They measure disagreement with respect to these top-k features in the metric called Feature Agreement (FA).

This metric receives a set of the top-k features from the first explanation E_a and the second explanation E_b. It computes the fraction of common features in the top-k features by looking at the intersection of these sets. The metric returns 1 when the set of features is the same for both explanations, while lower values indicate higher disagreement between explanations (Krishna et al., 2022).

$$FA(E_a, E_b, k) = \frac{|top\_features(E_a, k) \cap top\_features(E_b, k)|}{k} \tag{3.9}$$

3.2.2 UMAP

The uniform manifold approximation and projection for dimension reduction (UMAP) is a method to visualize some structural elements of a high-dimensional data set that is otherwise hard to grasp. It generates an interpretable, low-dimensional embedding while aiming to keep the global and local structure of the data set (McInnes et al., 2018).

The algorithm is based on a theoretical framework that is grounded in Riemannian geometry and algebraic topology.

The underlying idea is to approximate a manifold, that presumedly contains the data and then construct a fuzzy simplicial representation of this data in the manifold. In practice, the algorithm mostly constructs weighted graphs and manipulates them.

In the first phase the algorithm constructs a weighted graph with k neighbors, where the data points are represented by nodes and where changes to the edges relate to local distances. In the second phase this graph is turned into a low-dimensional layout, while attempting to hold onto the characteristics of the k-neighbor graph through an objective function.

The most relevant parameters in practice in the UMAP implementation are $n\_neighbors$ and min_dist. $n\_neighbors$ determines the size of the local neighborhood and thereby decides to set the focus more on local or global structures. min_dist is essentially controlling the output and how closely the data points finally are placed together (McInnes et al., 2018).

3.3 Autoencoder

Generally speaking, an autoencoder (AE) is a neural network that achieves its functionality through two parts. The first part, the encoder, receives the input and compresses it into a relevant representation, from which the second part, the decoder, picks up. It decodes the representation with the aim of reconstructing the original input as closely as possible.

Various applications for utilizing the latent representation created by autoencoders exist, for example in dimensionality reduction or to generate new data (Bank, Koenigstein, and Giryes, 2023). In our case, we use the autoencoder for translation from one set of explanations into another.

The foundation to this is to train the autoencoder to map an input vector to a desired output vector and in the process learn a translation function between two different data scopes. The input explanation set A contains explanations from one method for a set of data and the output explanation set B carries explanations from another method on the same set of data. Therefore, one explanation from A always corresponds to another in B due to the common explained data point.

The encoder E maps the input $a \in A$ to its compressed representation z. From z the decoder D maps onto the target $b \in B$. Both encoder and decoder combine to form the autoencoder. It trains on instances from a training set to minimize the loss function and can subsequently be used to predict the output on test instances.

Encoder $E : \mathbb{R}^n \rightarrow \mathbb{R}^p$ and decoder $D : \mathbb{R}^p \rightarrow \mathbb{R}^n$ try to minimize the expectation of the reconstruction loss function Δ in the following way.

$$\arg\min_{E,D} \exp[\Delta(x, D \cdot E(x)] \tag{3.10}$$

The simplest solution to minimize the loss would be to learn the identity function. But to ensure that there is a compressed representation, the dimension of the hidden layer of the autoencoder can be adjusted, thereby enforcing a bottleneck. The autoencoder aims to create a meaningful representation in between encoder and decoder from which it is possible to achieve a qualitative reconstruction (Bank, Koenigstein, and Giryes, 2023).

Chapter 4

Experimental Setup

4.1 Datasets and Models

We focus on two real-world or real-world-based datasets of different constitution.

The open ML dataset breast-w is based on the dataset Breast Cancer Wisconsin. From a digitized image of a fine needle aspirate of breast mass several features are gained. They describe the cell nuclei in the images and their characteristics. The features serve a classification task on if a tumor is benign or malignant. A Bayesian network was trained on this dataset so it could generate further pseudo instances. Among the nine predictors are features such as clump thickness or how uniformly shaped the cells are. The data set has 39366 instances with no missing values among them (Wolberg et al., 1995).

The second dataset, spambase, is a UCI dataset that is based on a collection of spam and non-spam e-mails. It also is a binary classification task on whether an e-mail is spam or no spam. Most of the 57 attributes describe if a particular word or character is frequently present in the email, as well as some features on the length of capital character sequences or such. Spambase consists of 4601 instances with no missing values (Hopkins et al., 1999).

The experimental setup is based on explanations from three different models for each of the two datasets. They performed well on the datasets.

4.2 Explanation Set

The explanation sets, we work with, are constituted and preprocessed in the following way.

4.2.1 Explanations

For all five attribution methods, Integrated Gradients (IG), KernelSHAP (KS), LIME (LI), SmoothGrad (SG) and VanillaGrad (VG), we received their computed explanations as our data to work with. They explained 1000 test instances from breast-w and

921 test instances from spambase. As we investigate five attribution methods, this produces explanation sets of 5000 and 4605 explanations, respectively, for all three models. We will mostly employ the explanations of the first model, but later on consider all three models as well.

These explanations constitute the data to conduct our experiments on. We will refer to them as the explanation sets, while the term dataset refers to breast-w and spambase.

4.2.2 Preprocessing

We scale the explanations by their maximum absolute values as a simple and efficient scaling technique. This scaling is likely to preserve sparsity of the explanation data, which is relevant for sparse attributions like those from LIME. The scaling helps assimilate the explanations without misrepresenting the relative distances within an explanation and thereby its structure. Additionally, it keeps the values of the instances in the range of $[-1, 1]$.

For the breast-w dataset, the explanation set on the first model contains 394 instances, where the attribution scores for all nine features are zero, meaning that no feature has been rewarded any feature importance for this data points prediction. All of the 394 explanations stem from the LIME explanation method. As for spambase's first model, there are 155 instances, which have been explained by LIME with zero feature importance in all 57 features.

We create an explanation set, where all LIME explanations contain at least one feature with an attribution score distinct from zero. The 394 or 155 instances are removed from the explanation set, and so are the corresponding explanations for the data points created by the four other attribution methods. We refer to this explanation set as the reduced explanation set. Whenever LIME participates in a section of the experiments, for example as one part of a pair in a translation task, we turn to the reduced explanation set for both explanation methods. It ensures that any participating explanation contains at least some information.

4.3 Preliminary Analysis

We conduct a preliminary analysis on explanations based on the first of the three models to improve understanding of the explanations. We want to be aware of their constitution, as they are the foundation of later experiments. In particular, we want to test if the assumptions of the hypotheses have a basis and to potentially state them more precisely after conducting the preliminary analysis. Additionally, we intend to look out for characteristics to estimate how well a translation could work when a specific method is participating.

4.3.1 Numerical Distribution

Initially, we pay attention to some statistical characteristics of the explanations. For each attribution method we observe the distribution of attribution values over the range

$[-1, 1]$ in a histogram to visualize, what kind of values are typically contained in the explanations of a certain attribution method.

We look at the variance of the explanations in nine features to see how stable the importance of a feature within a method is.

Then we investigate the variance within the instances for each method as well. This helps to see how far the attribution values in one explanation from a method deviate from the mean of the explanation. We average the variance over all explanations from each method.

Lastly, we examine the amount of features that contribute to 80% of the sum of absolute attribution scores of an explanation. The average over all numbers of features contributing to this arbitrary value hopefully provides an intuition on how the attribution information is spread over the explanations of an attribution method.

4.3.2 Feature Agreement

To gain insight on the agreement between the explanations from the different attribution methods, we then employ the Feature Agreement metric from section 3.2.1. We calculate pairwise Feature Agreement between all methods on the explanation sets from both datasets. We average the FA score over all 1000 or 921 instances, respectively. As the value for k, we select one third and two thirds of the features, as these are subsets we will continue to work with in different steps of the analysis.

Instead of computing agreement of the top-k features by magnitude we employ the non-absolute feature values, because we are naturally more interested in the features which contribute towards the prediction of the model and not against it. The results are presented in a heatmap, as they are typically relied on to display correlations and help visualize the intensity of agreement.

We hope to see, which methods align well in which features they give high importance reflected in the heatmaps. The Feature Agreement is an important contributor to expectations for which pairs of methods we expect to struggle or succeed most in the translation.

4.3.3 Structural Analysis

In this section, we explain the structural tests we utilize to see how separable the explanations from different methods are from each other. It is an important prerequisite for the translations, as we make sure that the feature values from different attribution methods are discriminable. This should ensure that there is a structural component to the explanations.

UMAP

We take the scaled and preprocessed explanations and reduce the dimensionality of each explanation set to two using the UMAP algorithm and implementation introduced in Section 3.2.2. Because of the reduction, we can present the explanations in a two dimensional scatter plot. We set $n\_neighbors$ to 15 as it is the default value in the UMAP implementation and showed to produce the most clear clusters for explanation sets on both data sets. We increase min_dist from its default value 0.1 to 0.7 as it is an aesthetic parameter and produces the best visuals this way for our task. Clusters are shown in a less dense way that is easier to draw information from.

By observing how the instances scatter on the plot after the UMAP dimension reduction, we hope to see clusters of explanations that stem from the same attribution method. This would be indication that the algorithm correctly identifies some structure inherent to the explanations from the same attribution method.

Classification

Another aspect of the structural analysis is a classification task, in which we attempt to train a logistic regression model to distinguish explanations produced by different methods. We conduct multiple pairwise classifications. For these, we split the explanations of two methods at a time into ten sets of 90% training data and 10% test data as known from 10-fold cross validation. We train the classifier on the training data and test it on the remaining tenth each of the ten times.

The independent variables are the explanations with their nine or 57 feature scores and the dependent variable is a number indicating which of the two methods they originate from.

We use the default linear solver from the sklearn Logistic Regression implementation (Scikit-learn, 2024), and measure the success of the classification task through the model's accuracy. When the accuracy is very high, we interpret this as a sign that the method pairs have some easily identifiable structural components. They are distinguishable. When the accuracy is low, we infer that the classifier is struggling to find any regularity in the structure of the methods to tell them apart consistently. We hope to see an accuracy above 90% for some pairs as indication, that it is possible to find some structural indicators. However, the interpretation is limited by the possibility of other distinctive characteristic that might clearly identify an explanations' method.

We present the ten results in boxplots to show the complete range of classification accuracies.

4.4 Translation Analysis

After the preliminary analysis, we reach the most relevant part of the experiments to answer our hypotheses.

H1 The translation of one explanation into another is more successful when both methods align in their underlying concepts of being perturbation-based or gradient-based.

H2 It is easier to focus the translation on the features with the highest attribution scores of each method than a translation over the complete range of features.

4.4.1 Translations

Considering the explanations from each method as separate explanation sets, we aim to translate from one explanation set into another to answer H1. We conduct translations in both directions for each pair of methods, as these translations are not symmetrical. Predicting the independent variables from the dependent variables causes different minimization functions and errors when the variables are reversed.

As the translators we use multivariate linear regression or an autoencoder as introduced in 3.3. We test out different autoencoder architectures with a bottleneck of varying size between two and six. Aditionally, we observe architectures with different hidden layers. The autoencoder either has no hidden layer, one in front of or behind the bottleneck or one on each side of it with eight or 16 neurons.

We measure translation performance as the success to produce an output closely replicating the aimed for explanation. The remaining distance is quantified by the mean squared error (MSE) between the explanation from the second explanation set and the translator output.

The best performance among many pairs of methods is achieved by the most elaborate autoencoder with one hidden layer of 16 neurons before and after the bottleneck, which itself has six neurons. However, this architecture with the maximum amount of neurons performs similarly to others. Therefore, we choose an additional autoencoder with only the bottleneck of five neurons aside from the input and output layer.

For each pair in each direction we again split the explanation set into ten fragments of 10% of the explanation set, each being the test set for the respective 90% training data. This is done for all three available models on either of the breast-w and the spambase dataset. We show the MSE results for all three models in a scatter plot and produce one plot for each combination of translation architecture and underlying dataset.

Mean Baseline

In addition to the scattered scores, we show the performance of a mean MSE baseline. With the help of this baseline the autoencoder and the linear regression results have a point of reference for how well they perform.

The mean baseline shows the mean squared error of the pairwise translation if the translator were to always predict the mean for each feature of all explanations of the method that is the second part of the translation.

4.4.2 Rankings

With regard to H1, we want to create rankings for the methods based on their translation performance. Ranking the method pairs against each other enables observations on how the pairs of methods perform in contrast to all other pairs.

Before ranking the method pairs, the MSE values achieved in the translations are divided by their corresponding mean baseline value. We choose to do this in an attempt to compensate for the possibility that it is inherently easier in this task to translate into one method versus another.

Afterwards, the scaled mean squared errors are ranked within their iteration of the ten evaluations, meaning among one of the ten explanation set splits into training and test data. All 20 values of the first split are ranked, the lowest MSE receiving rank one and the highest MSE after division by the baseline value receiving rank 20. We present the ten rankings based on the ten splits in a boxplot for each method and each translation, as well as all of the three models.

4.4.3 Translations over Subsets of Features

We perform the same pairwise translations as in section 4.4.1, but when one third or two thirds of explanation feature values are masked with zero. Masked are the features with the lowest attribution score within each explanation, so that the features with the highest contribution according to the explanation methods remain. The mean squared error is calculated accordingly to ensure that only non-masked features weigh into the error calculation.

We also show scatter plots to compare translation performance when there is none, one third or two thirds of the features with the lowest attribution score masked to tackle H2.

Chapter 5

Preliminary Analysis

5.1 Numerical Distribution

We present the results of the numerical explorations of the explanation sets. When we refer to pairs of methods, we denote them as Method1-Method2.

Breast-w Dataset

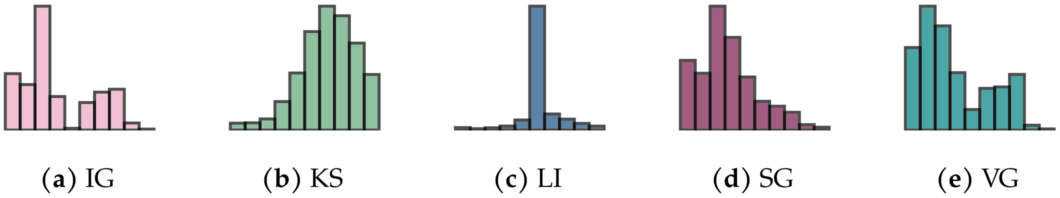


Figure 5.1: Feature value distribution per method. Each bar represents the frequency of values in one 0.2 interval over the range $[-1, 1]$. (breast-w)

Considering the value distributions of all methods on the breast-w dataset in figure 5.1, a few things stand out. The IG and VG distribution curves have a very similar look with relatively few values distributed around 0 and almost no values above 0.6. VG apparently produces more strongly negative values. The explanations from all gradient-based methods contain more negative values, in contrast to the more positive distribution from KS and LI. However, the value distribution curve from LIME stands out with a striking amount of values in the $[0, 0.2)$ range. Even after removing all all-zero explanations, a closer look shows, that many instances from the LIME method remain with mostly zeros. In total there are 2732 zero values in the 5454 possible feature values of the reduced explanation set for LIME.

The dimensional variance shows similar tendencies among all features in how the values are distributed. Feature one, three and six show the highest variance across all methods. The values shown in Table 5.1 seem to align best for IG and VG, even though IG has a much higher variance in feature one, that VG compensates for in feature six and nine. They show high values across features in comparison to the other methods.

	1	2	3	4	5	6	7	8	9
IG	0.88	0.17	0.47	0.06	0.14	0.37	0.16	0.12	0.22
KS	0.36	0.11	0.18	0.09	0.11	0.20	0.14	0.13	0.13
LI	0.36	0.03	0.23	0.01	0.04	0.23	0.10	0.05	0.08
SG	0.53	0.09	0.20	0.05	0.07	0.24	0.14	0.08	0.14
VG	0.62	0.13	0.45	0.06	0.11	0.46	0.16	0.14	0.30

Table 5.1: Dimensional variance of explanations. (breast-w)

SG shares similar relative dimensional variances, but the variances are less extreme. The variances shown for KS are more evenly distributed, while LI only shows higher variance in three of the nine features. The other features display a variance below or equal to 0.1.

IG	KS	LI	SG	VG
0.05	0.17	0.16	0.10	0.07

Table 5.2: Mean variance within explanations. (breast-w)

The mean of variance within the explanations shows highest values for KS and LI and a low mean variance in explanations from IG and VG in Table 5.2.

IG	KS	LI	SG	VG
6.56	5.49	2.69	5.84	6.31

Table 5.3: Average feature contribution to 80% of the summed feature score per method. (breast-w)

Table 5.3 shows that more than six features usually contribute to make up 80% of the absolute feature values in IG and VG explanations. About one feature less is needed for KS and SG. On average 2.69 features contribute to 80% in LI, which is by far the lowest value.

Spambase Dataset

For the spambase dataset, in Figure 5.2, the value distributions appear to be less diverse. The curves, that the histograms indicate, are roughly bell-shaped and symmetrical around the center. IG and VG appear to be very similar in the details of the distribution, again. VG shows the most high negative or high positive values in comparison to the other attribution methods. The KS distribution is much narrower than the gradient-based distributions, indicating that most feature scores lie in a more constrained range of attribution values. LI, again, shows the most narrow distribution, with the most values by far in the range $[0, 0.2)$. The explanations in the reduced explanation set amount to 37049 zero values over the 43662 possible feature values.

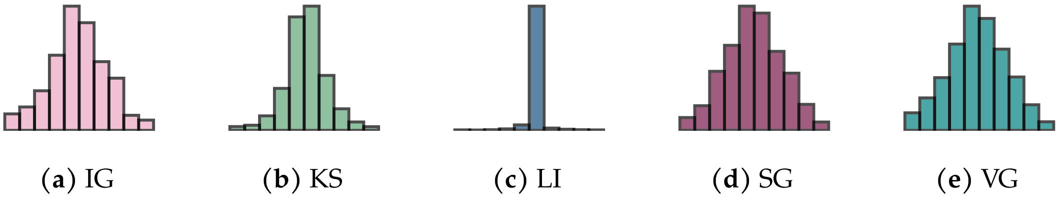


Figure 5.2: Feature value distribution per method. Each bar represents the frequency of values in one 0.2 interval over the range $[-1, 1]$. (spambase)

	20	25	27	46	52	53	38	47	50
IG	0.53	0.56	0.96	0.64	0.40	0.76	0.01	0.00	0.00
KS	0.10	0.36	0.30	0.22	0.21	0.26	0.04	0.04	0.04
LI	0.03	0.14	0.13	0.07	0.15	0.27	0.00	0.01	0.00
SG	0.22	0.33	0.70	0.35	0.27	0.48	0.04	0.05	0.05
VG	0.31	0.48	0.92	0.50	0.32	0.65	0.02	0.01	0.02

Table 5.4: Dimensional variance of a few selected features. (spambase)

Selected by the sum of all variance values for a feature, we present the six features with the highest variance and the three with the lowest variance from the 57 features of the explanations based on the spambase dataset in Table 5.4. IG and VG mostly agree on the dimensional variance again and display the highest variance values in comparison to the other features. The variance along the dimensions of KS explanations is more spread out, even showing variance of 0.04 in the features of lower variance. So does SG, but its explanations show higher dimensional variance in the other features than KS. We can see the lowest variances in LI explanations.

IG	KS	LI	SG	VG
0.16	0.09	0.03	0.16	0.17

Table 5.5: Mean variance within explanations. (spambase)

The mean variance within the explanations shows opposite behavior to the prior dataset, as LI has the lowest mean variance by far. KS follows and lastly, we see all gradient-based methods with a much higher mean variance of around 0.16 in Table 5.5.

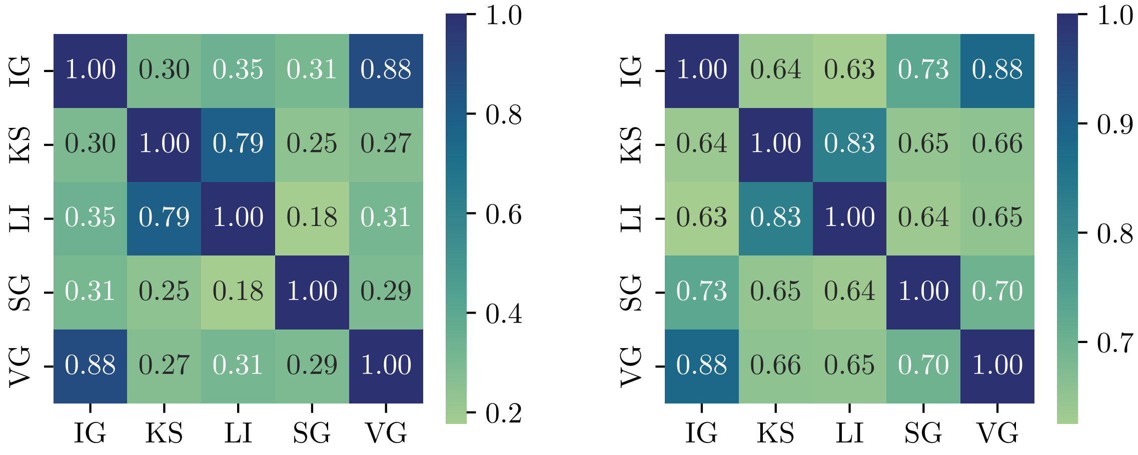
IG	KS	LI	SG	VG
28.14	27.47	4.10	28.85	29.14

Table 5.6: Average feature contribution to 80% of the summed feature score per method. (spambase)

Of the 57 total features around 28 features on average contribute to 80% of the total absolute attribution value within an explanation for all gradient-based methods and KS. LI only needs around four features to make up the 80%.

5.2 Feature Agreement

Breast-w Dataset



(**a**) FA for k=3

(**b**) FA for k=6

Figure 5.3: FA for one third and two thirds of the top-k features. (breast-w)

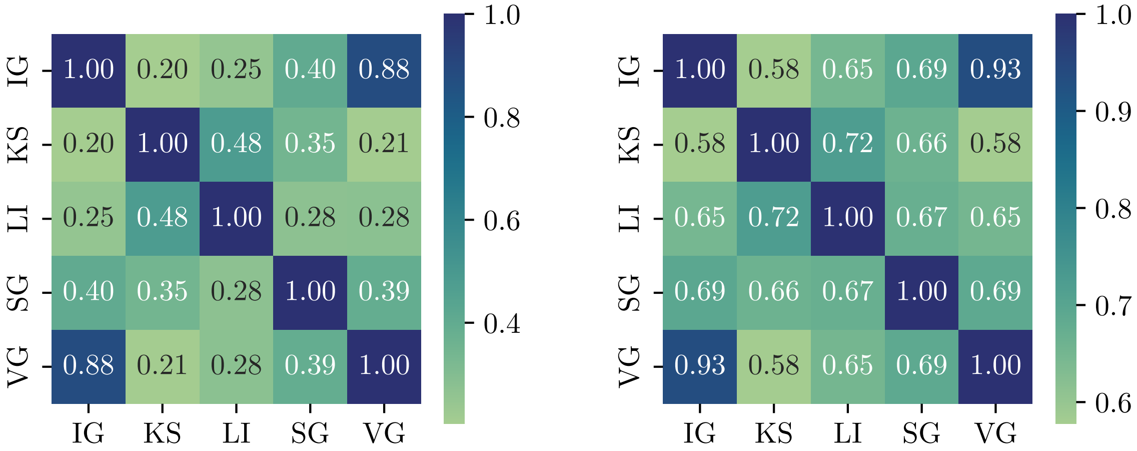
The FA on explanations based on the breast-w dataset when k is set to 3 is between 0.18 and 0.35 for almost all pairs of methods. The two exceptions are IG-VG with 0.88 agreement and KS-LI with 0.79 agreement (5.3 (a)).

Regarding two thirds of the features, when k is set to 6, similar behavior is visible in Figure 5.3 (b). However, the less agreeing methods from before have reached FA of around 0.65. IG-VG remains the most agreeing method pair, and KS-LI slightly increase their agreement. Both other gradient-based method pairs, IG-SG and SG-VG, set themselves slightly apart from the remaining method pairs.

Spambase Dataset

Observing agreement on one third of the features for the explanations on the spambase dataset in Figure 5.4 (a), means setting k to 19. The agreement of the top features is low for most pairs, as many record feature agreement values below or equal to 0.4. KS-LI show 0.48 feature agreement and IG-VG agree most with a feature agreement of 0.88.

When k is set to 38 in Figure 5.4 (b), most FA values range from 0.6 to 0.7. The feature agreement value for KS-LI is 0.72, and for IG-VG it is 0.93. The least agreeing pairs are either IG or VG in combination with KS.



(**a**) FA for k=19

(**b**) FA for k=38

Figure 5.4: FA for one third and two thirds of the top-k features. (spambase)

5.3 Structural Analysis

We proceed to present the results of the experiments on the structure of the explanations from either explanation set.

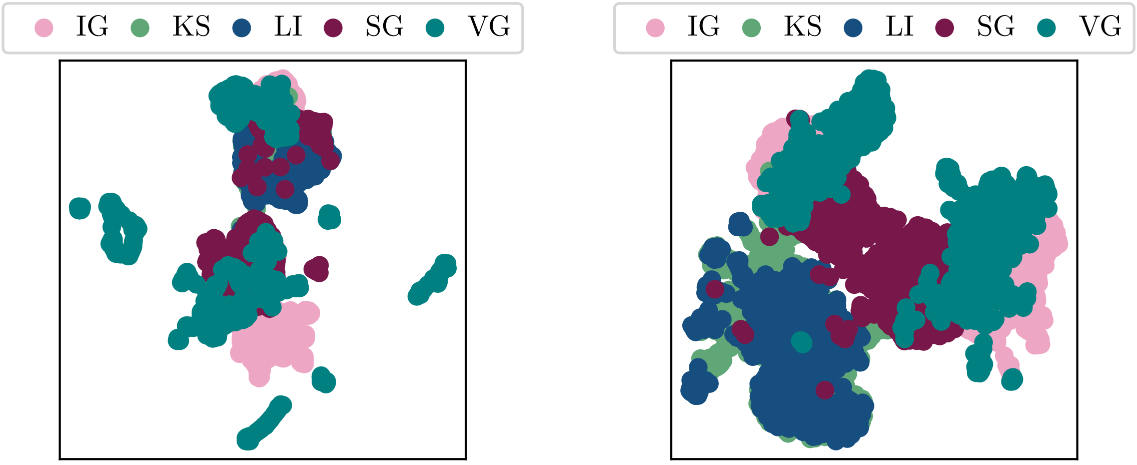
5.3.1 UMAP Results

Breast-w Dataset

The UMAP mapping of explanations based on breast-w shows an upper cluster with almost all KS and LI explanations (Figure 5.5 (a)). They are overlapping significantly. This cluster is added onto by some instances of each of the gradient-based methods. Additionally, we see a lower cluster with most explanations from IG, SG and VG. Lastly, some points, mostly from VG, are scattered in much smaller clusters, separated from any bigger cluster. The KS and LI clusters seem more compact in contrast to the lower density of the cluster of explanations from VG. The separation between the five explanation methods and from the clusters in general is not complete, but some groupings are clearly discernible.

Spambase Dataset

The explanations of data points from the spambase dataset again display a big cluster on the bottom left with notable overlap of all KS and LI explanations in Figure 5.5 (b). This time, only a few SG and VG instances are mixed within. We see two smaller IG and VG clusters which are interconnected by a spatially larger SG cluster. The SG cluster seems



(**a**) breast-w (**b**) spambase

Figure 5.5: UMAP dimension reduction of the explanation sets.

to almost separate itself into two clusters as well. There is a significant amount of overlap for IG and VG explanations and some overlap from both of the methods to SG. The explanations from all attribution methods are more clearly clustered in the spambase scatter plot than in the breast-w scatter plot.

5.3.2 Classification Results

Breast-w Dataset

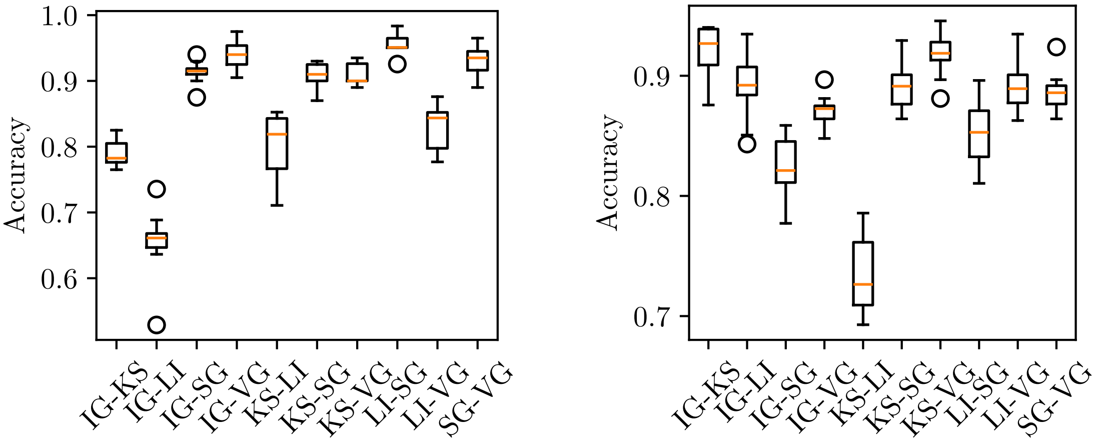
In Figure 5.6 (a) , we can see many classification results with an accuracy above 90% on the breast-w data set. All method pairs, where SG participates, have a median above 0.9. Further considering the median, the lowest accuracy by more than 10% is recorded by IG-LI, followed by KS-LI. The pair of LI and VG is also fairly low in accuracy, showing lower accuracy in the classification task for all pairings with LI, except for LI-SG.

Spambase Dataset

For spambase, Figure 5.6 (b) shows only two method pairs with a median above 0.9. Most pairs' accuracy is in the range from 0.85 to 0.9. The median for IG-SG is easily below 0.85. The pair of KS and LI show the lowest accuracy by far with a median around 0.77. The classifier achieves above 90% accuracy for many of the evaluations.

5.4 Discussion

The numerical analysis shows similar distributions among the gradient-based methods, especially for IG and VG. This aligns well with their similar dimensional variance. SG



(**a**) breast-w

(**b**) spambase

Figure 5.6: Accuracy of the logistic regression classification on pairs of explanation methods.

shows less extreme variance values than the other two gradient-based methods, but distributes the variance values more evenly among all features. SG often seems to position itself somewhere in the middle between characteristics shown by KS and LI versus characteristics shown by IG and VG.

The dimensional variance is even more evenly distributed for explanations from KS and their value distribution shows less inclination to extreme attribution scores, meaning scores close to -1 or $+1$. On explanations based on either dataset, LI displays a striking amount of zero attribution scores. This property possibly causes the explanations' higher variance on the lower dimensional explanation set and the lower variance on the higher dimensional explanation set. It is likely that the 57 features lead to a big amount of features that contribute little to the prediction for LI, especially as four features are supposedly already covering 80% of the absolute feature sum. For all other attribution methods we find it interesting, that it takes this many features to make up the 80%. The methods, except for LI, apparently agree, that a big variety of features influence the model's prediction fairly strongly in either direction.

The FA repeats some observations from the numerical analysis as there is strong agreement for the pair IG-VG, followed by KS-LI. However, the rest of the methods show low FA, which we find surprising, especially for the other gradient-based pairs. Only when increasing k to two thirds of the features, they set themselves apart from the method pairs with mixed underlying concepts.

Structurally, it seems that there are some components for each explanation set, that the classifier and UMAP algorithm can successfully detect. The methods clearly produce UMAP clusters for each method, which are usually more compact for the perturbation-based methods than the others. KS and LI show impressive overlap in their clusters, while the gradient-based methods tend to spread more. The scatters of IG and VG explanations are often quite similar, which is expected after prior investigation.

The pairwise logistic regression classification for explanations based on the spambase dataset shows the lowest accuracy for KS-LI indicating that the explanations from these methods could be hard to tell apart. The method pair is more accurately classified for the breast-w dataset. We observe that IG-VG explanations seem to be surprisingly easy to accurately classify, but this could also simply be an indicator for some inherent telling attribute. Method pairs with SG usually display a high accuracy as well, indicating that SG could be distinct from the other methods in some way. Overall, the classification results are not easily aligned with the other results. Therefore, we consider the classification task as more of a check, if a classifier is in general able to find some giveaways for different pairs of methods. We do find classifications with high accuracy and the UMAP plots show distinct clusters, which we consider indication for the idea of an identifiable structure of the explanations from different methods.

Over the complete preliminary evaluation, IG and VG consistently show similar characteristics and agree on variance in both features and explanations. Their UMAP clusters seem similar. KS and LI are often grouped together throughout the analysis as well, but not as consistently as IG and VG. However, the UMAP algorithm produces a reduction where they cluster with the most evident overlap out of all methods. Additionally, the many zeros in LI are very apparent. SG seems to be a bit different to either method pair.

5.5 Summary

The preliminary analysis improves understanding of the structure of the explanations produced by different attribution methods. We investigate the assumption, that the underlying concept of an attribution method determines some part of the structure of this method's feature attributions.

Generally, we find a basis for the formulation of our hypotheses H1 in the preliminary analysis. We see characteristics of the explanations that show similarities between methods that share an underlying concept, more than when underlying concepts do not align. The feature attributions from different methods seem to have some structural component and thereby appear to be separable from one another.

In accordance with the results of the preliminary evaluation, we expect the gradient-based pair IG-VG to translate especially well, as well as the perturbation-based pair KS-LI. However, the constitution of LI with its many zeros might induce difficulties to the translations. Additionally, we expect the translations to struggle when SG is part of the translations. Still, it is not inconceivable that the distinct characteristics of SG lead to ease of translation.

It remains unclear if similarities between methods are an indicator of successful translation or if clearly distinguishable structures help the translation process.

In regards to H2, the feature contribution results, as well as the fairly high variance shown in the numerical experiments, make us doubt, whether a focus on only the features of highest importance will make the translations more successful. Perhaps, a reduction by two thirds will remove too much of the content of the attribution scores.

We proceed with our prior hypotheses with special attention to the pairs IG-VG and

KS-LI for H1 and the expectation that masking two thirds of the features might not make the translations easier for H2.

Chapter 6

Translation Analysis

6.1 Translations

The plots in Figure 6.1 and 6.2 of translation MSEs show the ten different explanation set splits for each of the three models and each of the possible attribution method pairs in both directions. There are two plots, one for the more elaborate autoencoder architecture and one for the linear regression. The results from the simpler autoencoder architecture mentioned in Section 4.4 are very similar. They can be found in Appendix A.

We denote a translation from one method into another method as Method1_Method2.

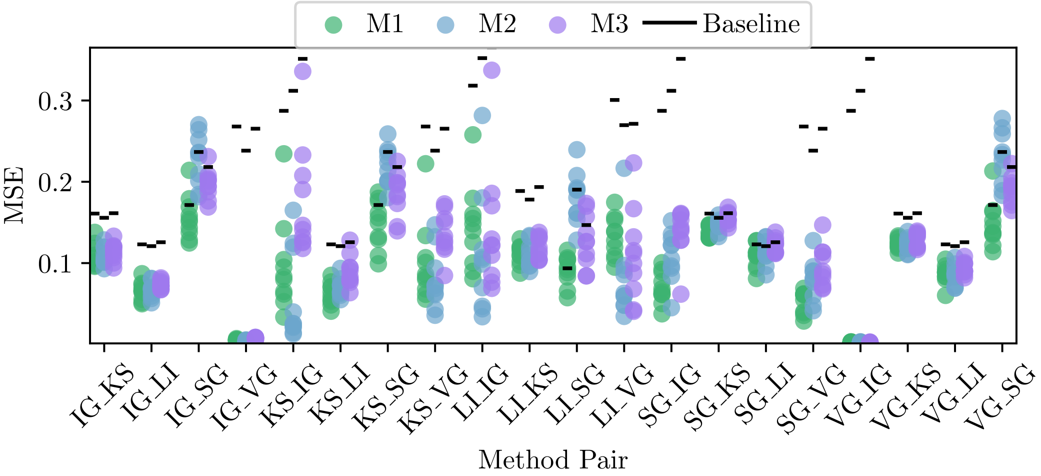
Breast-w Dataset

As seen in Figure 6.1 (a), the MSE values of the ten evaluations of the autoencoder are often spread out over a wide range of values, differing by up to 0.15 in their MSE. The errors per model and method tend to be of higher range when the MSEs are higher themselves. Comparatively well performing method pairs often show that the ten MSE values are more densely stacked. There are a few badly performing outliers.

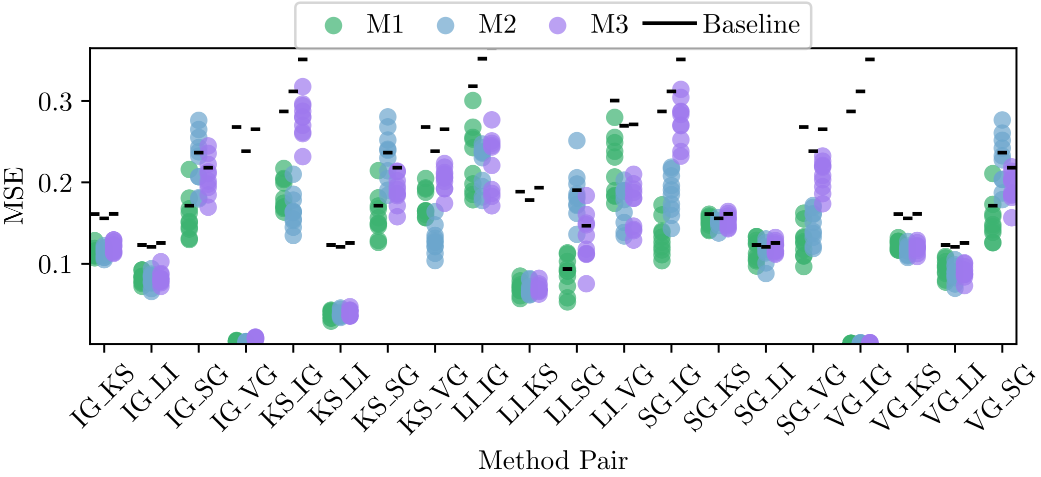
The biggest MSE on average can be seen for IG_SG, KS_SG and VG_SG. For some method pairings like IG_SG or LI_VG the plot shows big differences in MSEs among the three different models. The most outstanding translation results are the MSEs of IG and VG in both directions. Their mean squared errors of all evaluations across models is very close to zero. The next best translation results are recorded by KS_LI, IG_LI and VG_LI, showing that a translation into LI works well.

The mean MSE baseline values, indicated by a black line in the plot, differ by a lot. The baseline values are especially high for VG and IG, followed by SG. LI presents the lowest baseline by far. There is just one baseline of a similarly low value for SG as part of the translation from LI, because the reduced explanation set is in use. The KS baseline is also much lower than the remaining gradient-based baselines.

Most evaluations overall beat the mean MSE baseline, but there are some occasions where the MSE of the translation is higher than the corresponding mean baseline value. We especially see this happening for translations into SG and for SG_LI as well, as the LI baseline is very low.

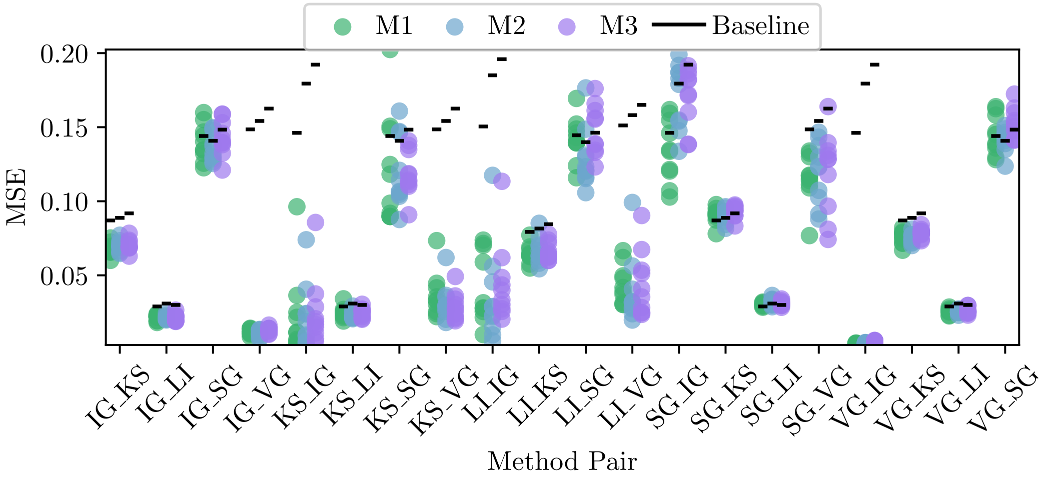


(**a**) Translation via Autoencoder

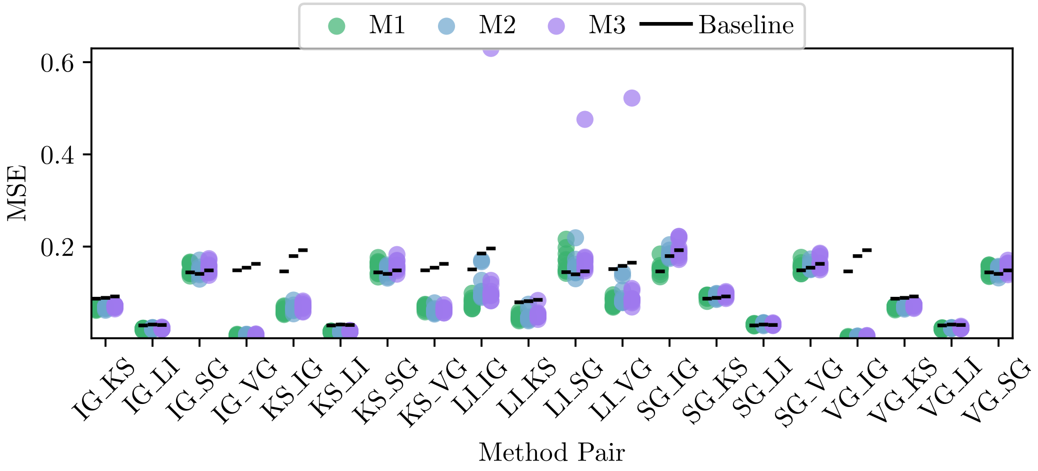


(**b**) Translation via Linear Regression

Figure 6.1: MSE of translations of explanation method pairs for models 1, 2 and 3. (breast-w)



(**a**) Translation via Autoencoder



(**b**) Translation via Linear Regression

Figure 6.2: MSE of translations of explanation method pairs for models 1, 2 and 3. (spambase)

The linear regression translator manages to translate the prior well performing methods with a similarly low mean squared error (Figure 6.1 (b)). However, the average MSE for the worse performing method pairings is higher than when utilizing an autoencoder. Oftentimes, model two or three seem to lead to worse translations than model one. Additionally, KS_LI and LI_KS are emerging as well performing method pairs with a significantly lower MSE in comparison to when the autoencoder is employed.

Spambase Dataset

Figure 6.2 (a) with spambase as the underlying dataset and the autoencoder as translator shows a much clearer picture in general. We detect a lower tier of method pairs with smaller MSEs at the bottom and an upper tier of higher MSEs towards the top of the plot. The mean squared errors are in general lower than for breast-w explanations.

Multiple times the mean MSE baseline value sits just on top or within the five higher MSEs of the ten evaluations. The baseline is clearly beaten by IG_VG, KS_IG, KS_VG, LI_IG, LI_VG and VG_IG, so any time we translate into IG or VG and SG is not part of the pairing. All pairs in the upper tier have SG participation, only missing SG_LI and SG_KS to complete the collection. The translations into LI and KS are often beating their baselines as well, but as the baseline values are so low, there are a few exceptions.

When employing linear regression as the mean of translation, the same distribution of method pairings and performance can be seen in Figure 6.2 (b). Some translation directions of the upper tier have a higher mean squared error and therefore compare worse to their baseline value. We detect a few outstandingly high MSEs when model three is the basis for the explanations.

6.2 Rankings

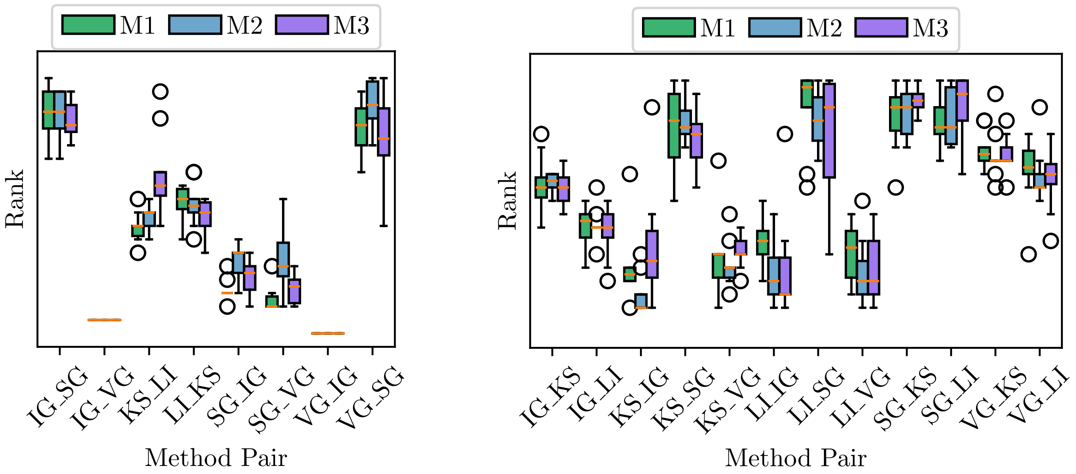
In the ranking plots all tens of translation MSE values are divided by their mean MSE baseline value and ranked in comparison to the other translation method pairs.

Breast-w Dataset

In Figure 6.3 (a) we see that VG_IG and IG_VG consistently rank first and second. Translations from SG rank roughly around fifth, while translations into SG from other gradient-based methods gather in the area around rank 16. Regarding perturbation-based method pairs, KS_LI rank around the eigth and LI_KS around the ninth rank.

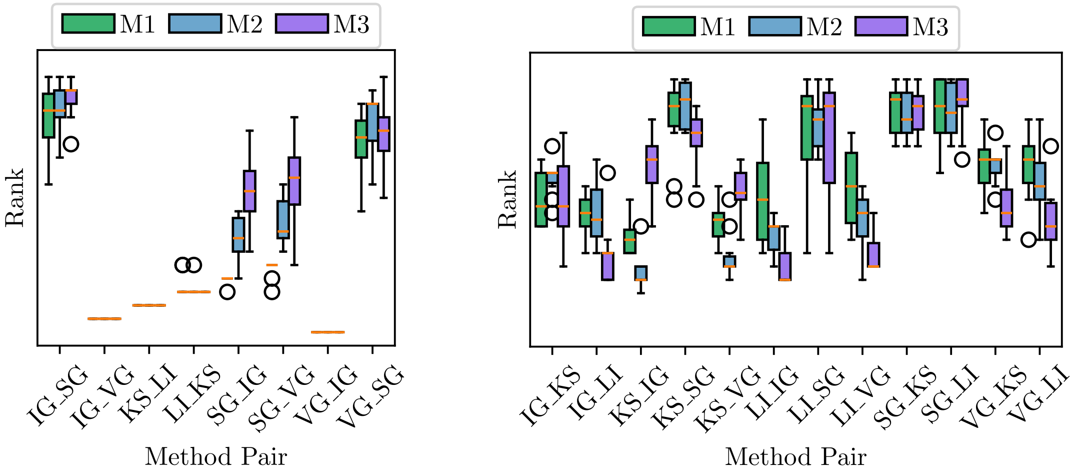
The ranking of method pairs with different underlying concepts (Figure 6.3 (b)) occupy all different kinds of ranks. On especially high ranks, often up to rank 20, we see translations from and into SG. Translations from VG rank rather high as well. In this plot, most ranks are not consistent. The translation direction of KS or LI into VG or IG translates better than the other way around.

In the plot for linear regression translation (Figure 6.3 (c)), it is immediately visible that KS_LI and LI_KS consistently rank third or fourth. For model one, SG_IG and SG_VG are almost constantly occupying rank five and six, however translating into SG still remains on high ranks.



(**a**) Shared concepts with AE

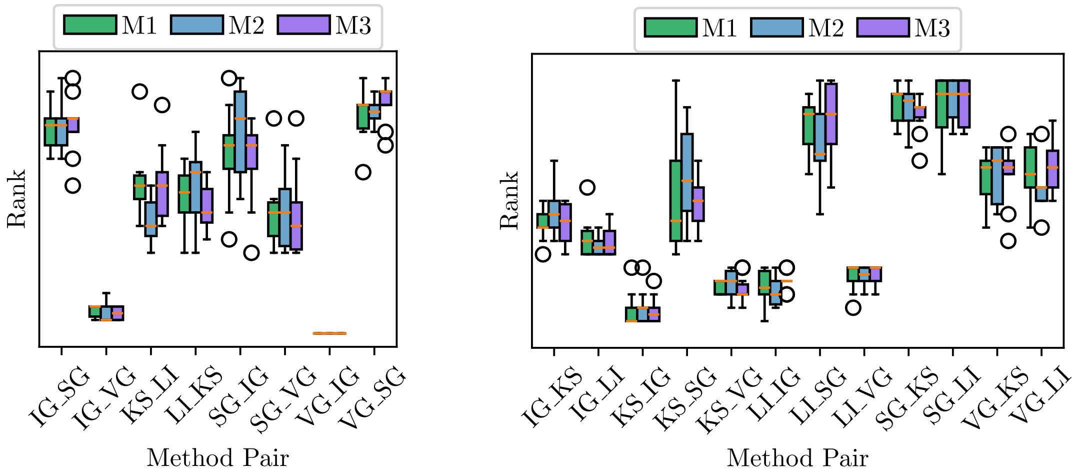
(**b**) Mixed concepts with AE



(**c**) Shared concepts with LR

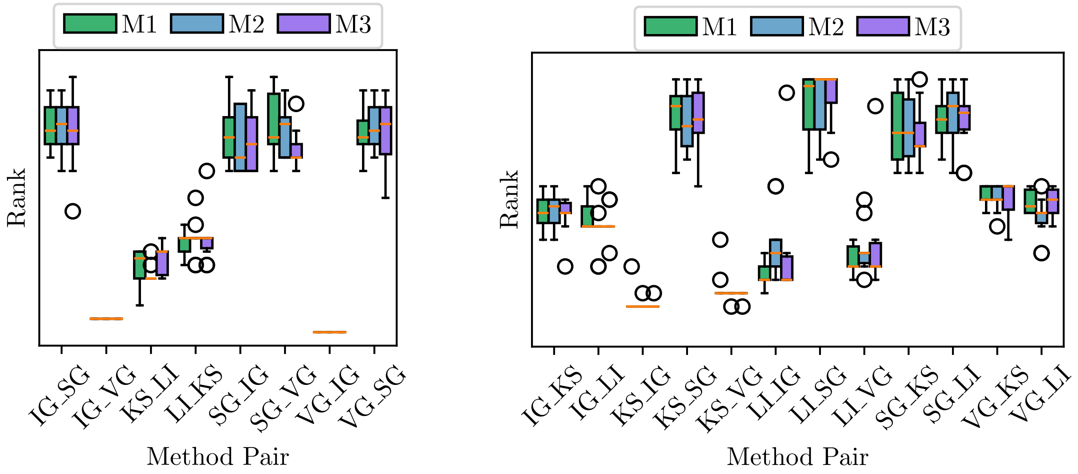
(**d**) Mixed concepts with LR

Figure 6.3: Ranking of MSEs of translations of explanation method pairs for models 1, 2 and 3. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts (breast-w).



(**a**) Shared concepts with AE

(**b**) Mixed concepts with AE



(**c**) Shared concepts with LR

(**d**) Mixed concepts with LR

Figure 6.4: Ranking of MSEs of translations of explanation methods for models 1, 2 and 3. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts (spambase).

The ranking improvements when underlying concepts are shared, lead to lower ranking for the mixed concepts in Figure 6.3 (d), but an otherwise similar ranking to the autoencoder.

Spambase Dataset

For the explanations based on the spambase dataset both directions of the IG_VG pairing are still invariably first and second. The rankings for translations of gradient-based pairs from SG show much worse results than on breast-w data. SG_IG reaches similar ranks as translations into SG, while SG_VG still ranks slightly better (Figure 6.4 (a)).

Ranks three through six, seldom even rank two, are filled with KS_IG, KS_VG, LI_IG and LI_VG most of the time. They are followed by the same pairings in the other translation direction.

In the linear regression translations the perturbation-based pairs KS_LI and LI_KS again place around sixth and seventh rank. Pairings translating from SG occupy the area around rank 16 and KS_IG and KS_VG most often take up rank three or four. Overall, we see a similar distribution of rankings, but they appear to be slightly more consistent.

6.3 Translations over Subsets of Features

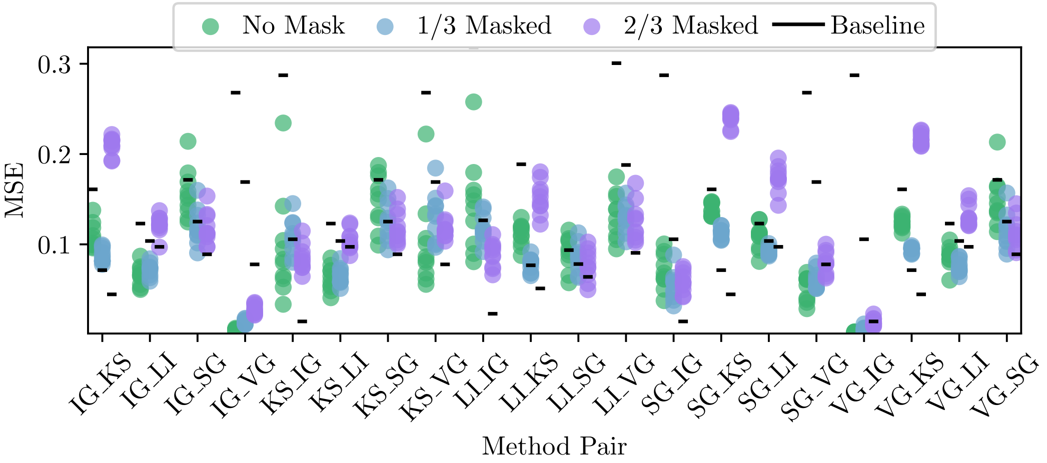
The scatter plots, again, show the ten evaluations per translation method pairing. The green points indicate that none of the features have been masked. The blue and purple scatters indicate that one and two thirds of the smallest features scores per explanation have been masked with zero. The black line indicates the corresponding mean MSE baseline value for all features, and the subset of the remaining two or one thirds of highest features, respectively.

Breast-w Dataset

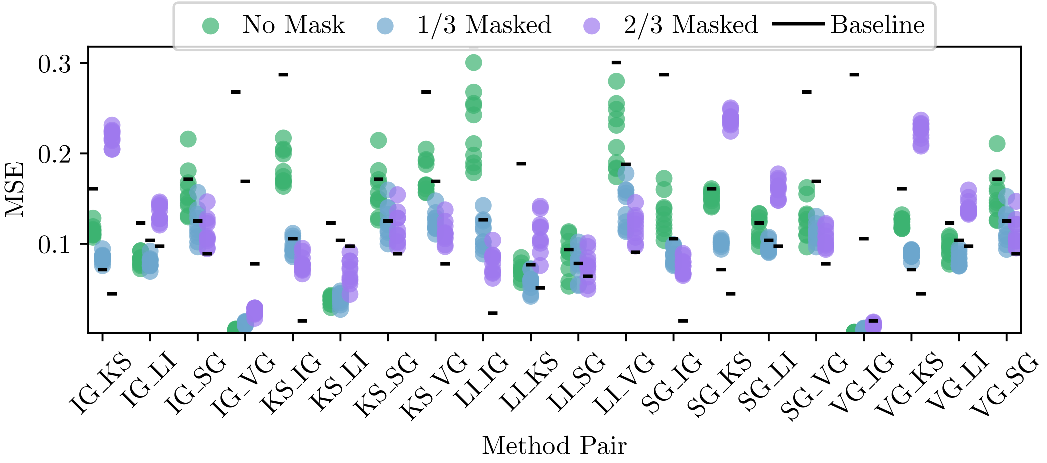
For the autoencoder architecture, 8 of the 20 method pairs and directions show an improvement in their translation MSE when one third of the less contributing features are masked in Figure 6.5 (a). The blue MSEs scatter lower than the green ones, but usually not by much. Translating into KS seems to especially benefit from the masking. There is a comparable performance on seven other method pairs and the MSE is clearly worse for the pairings with the lowest MSE, IG_VG and VG_IG. Three other MSEs also deteriorate.

When six features are masked with zero, we only detect two improvements in comparison to no mask. They only improve by a little, like SG_IG. 12 method pairs and directions perform worse in translation, and often their MSEs worsen by a lot. The remaining pairs show similar MSEs. Translating into SG often benefits from the mask.

Without masking any features, MSEs of the evaluations by linear regression are slightly worse than they are for the autoencoder (Figure 6.6 (b)). Exceptions are KS_LI and LI_KS, where the linear regression performs particularly well on. We see that many MSEs are improving, only both directions of IG and VG deteriorate and four of the 20 pairs and directions stay approximately the same. The plot shows clear improvements and much more consistent results overall.

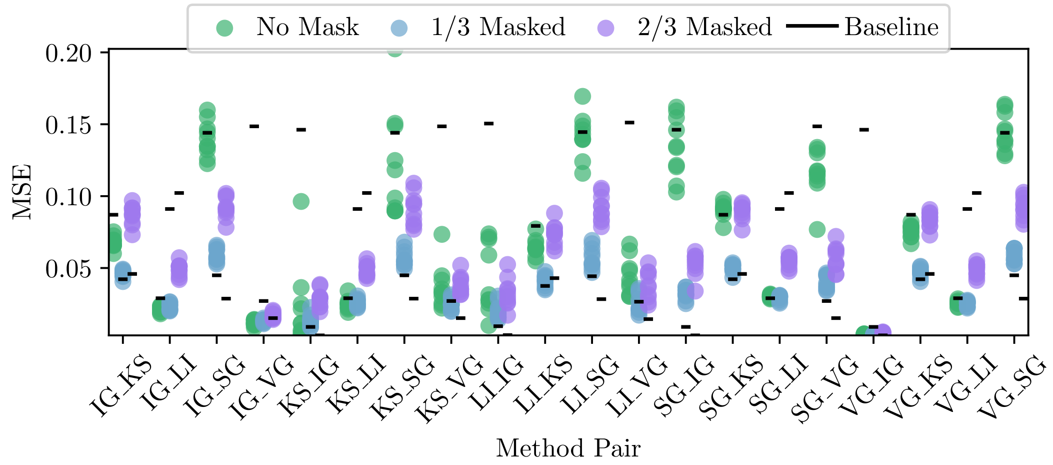


(**a**) Translation via Autoencoder

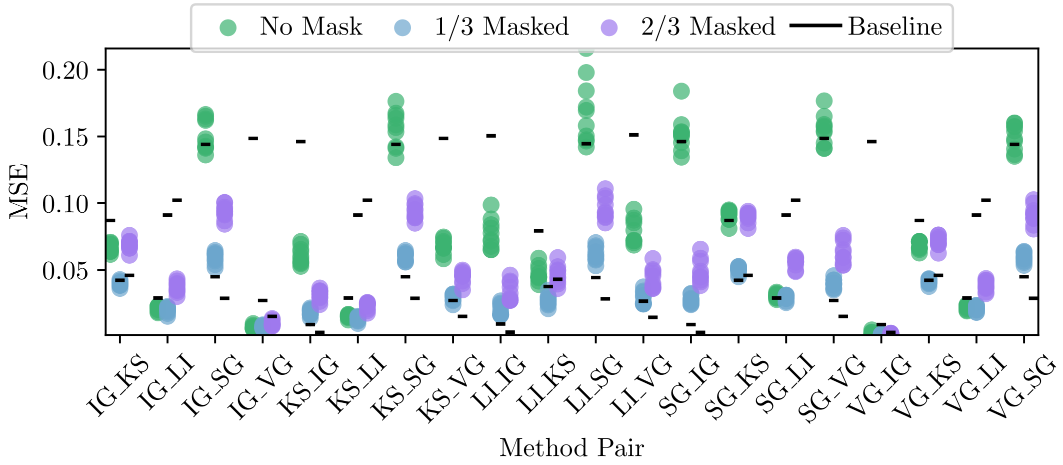


(**b**) Translation via Linear Regression

Figure 6.5: MSEs of translations where none, one third or two thirds of the less important features have been masked. (breast-w)



(a) Translation via Autoencoder



(b) Translation via Linear Regression

Figure 6.6: MSEs of translations where none, one third or two thirds of the less important features have been masked. (spambase)

The linear regression translation benefits more from masking another third than her predecessor as well. Nine improvements remain, ten method pairs show worse MSEs and one stays the same. Some improvements are very clear, and so are some drops in mean squared errors.

The mean MSE baseline value often drops by a lot for each third being masked. When two thirds are masked, the translation MSEs rarely beat their baseline value. When one third is masked, the baseline is sometimes beaten, sometimes lies within the ten evaluations and sometimes beats the translation MSEs.

Spambase Dataset

The explanation set based on spambase data points with 57 instead of 9 features shows a different reaction to the masking (Figure 6.6).

Investigating the autoencoder architecture, we initially see that the uppermost tier of scatters only contains green scatters indicating the complete set of features. It is made up of all translations into SG and two from SG. All of the MSEs when subsets have been masked steer clear of this value region of MSEs, staying below an MSE of around 0.12.

We detect 13 improved MSEs when one third is masked, many of them being significant improvements. The other seven MSEs are comparable to the no-mask MSEs. When masking another third of features, six improvements persist, all of them translations into or from SG. Eight translation pairings stay roughly the same and six show worse performance. The declines are detected for translations from IG or into LI.

Once again, the linear regression based translations start out with slightly worse MSEs than when translating with an autoencoder (Figure 6.6 (b)). However, with one third masked we see 14 often very clear improvements, while six MSEs remain about the same. Masking the additional third, ten improvements prevail, six translation pairs perform similarly and the MSEs of four pairs worsen.

The mean MSE baseline for spambase shows even harsher drops, when the mask is applied. Translations into LI present exceptions, as the baseline value actually rises.

6.4 Discussion

Following up on the results, we discuss patterns in the translation MSEs and rankings. We relate them back to the characteristics of the explanations and their methods.

6.4.1 Translation Discussion

The ten translation evaluations per method pair and model show errors that are often so spread out, that it is difficult to view them as a coherent result. We consider a good translation performance to depend on consistent evaluations as well. As a reminder, whenever LI participates in the translation, this is indication for different fundamentals. The reduced explanation set is in use, which, for example, leads to a completely different MSE baseline. Additionally, the performance of a method sometimes varies a lot depending on which model the explanations are based on. Roughly, we can state that, whenever the MSE is low, it is more likely to be consistent across the ten evaluations and the three models.

As for the performance in translations of specific methods or pairs, we can detect a tendency that translations from LI or into SG result in high MSEs. The pair of IG and VG in both directions are unmistakably the best performing translation pair across all architectures and the two underlying datasets breast-w and spambase. Translations into IG or VG work well if the underlying concept of methods is mixed, as well. Perhaps they contain clearly reproducable structural components.

The perturbation-based model pairs KS and LI show varying translation success. When the translation architecture is linear regression, they, for example, show much lower MSEs than for the autoencoder. Linear regression, however, is the only mean of translation where very badly performing outliers occur.

Translations on explanations based on the spambase dataset produce a much clearer picture with the tier-like distribution of MSEs as seen in Figure 6.2. A potential reason for this could be the much higher dimensionality of the explanations causing tendencies visible on the breast-w dataset to be more enhanced. The 57 dimensions could likely be the reason for the lower MSEs in general, as the model output is to be attributed over a much bigger range of features. Therefore, it would be natural for many features to not contribute to such a big extent and cause lower MSEs.

6.4.2 Ranking Discussion

The divison through the mean MSE baseline value naturally improves the standings of translations into IG or VG as their baseline value is usually especially high. It has a reverse effect on translations into LI and KS, because their corresponding MSE baseline is significantly lower.

Some observations remain the same when ranking the results, as we see IG_VG and VG_IG still unbeaten in first place, and they are even more clearly ahead of other method pairs after the division through the baseline value. Linear regression ranks KS and LI consistently around rank three and four. We see inconsistent rankings when the underlying concepts are mixed. However, lower ranks are often filled with KS or LI into IG or VG.

The autoencoder produces similar rankings, showing a small advantage on some translations, often including SG.

6.4.3 Discussion of Translations over Subsets of Features

The MSEs when none, one third or two thirds of the features are masked show that for the linear regression translation it is seemingly beneficial to remove some of the dimensionality of the explanations. Over all architectures, there is generally more improvement when a mask is applied to explanation sets from spambase than from breast-w. Perhaps the big difference in the amount of features is the reason for this.

Masking one third of the features much more often results in improved MSEs than a mask of two thirds of the features.

Considering specific methods, translating into SG often benefits from a mask. The prior experiments suggested some difficulty translating into SG, as well as some complicated elements of the structure of SG explanations in comparison to other methods. Potentially, a reduction of information is helpful in this case for the translations. In contrast, translating into LI seldom benefits from masking. We speculate that often times, some of the many zeros are simply masked with zero and the translation task roughly remains the same.

The low mean MSE baseline increases skepticism if the MSE of translation of the detected improvements can actually be evaluated as easier translation. If we were to divide the masked MSEs by their baseline value as before, we would be seeing a bad performance for many of the method pairs compared to no mask.

6.5 Summary

Among the gradient-based methods, IG and SG are supposedly superior to VG, as they are both enhanced gradient-based methods and more complicated than VG. So to this extent, the level of consistency and ease in translations from IG to VG, and the other way around, is surprising. It, however, suits the results of the preliminary analysis well. Although IG strives to detect non-linear relationships in contrast to VG's linear view, perhaps we can still detect an overlap in the methodology. IG's path integral can be seen as a collection of many linear approximations, fitting VG's single linear approximation. They both assume some smoothness to the function and perhaps, it is task-dependent how well this assumption holds. Conceivably, this aligns the methods well in this case.

SG and VG can differ significantly in practice due to the denoising iterations of SG. SG tends to identify consistent and robust features and the approach of denoising through adding noise is not really comparable with what the other gradient-based methods are doing in feature attribution. The path integral of IG and the perturbations of the input of SG both average out some evaluations as to reduce noise in the explanations. However, SG introduces perturbed instances and randomness, while IG uses the path from the baseline.

The perturbation-based methods KS and LI are different from one another, as LI tends to produce sparse attributions due to its local approximation and KS often distributes attributions over a greater range of features. Their similar characteristics in the numerical distribution as well as their overlapping UMAP scatterings are hopeful indicators for the translations. In the linear regression translation KS_LI and LI_KS performed convincingly. We speculate, that the reason for this is, that LI approximates with a linear model and KS uses additive feature attributions, suiting the linear form of translation well.

The numerous zero features scores of LI explanations could limit the translations of the perturbation-based method pairs. This constitution of explanations produced by LI is possibly the reason for the poor performance whenever a translation starts with LI. The many zeros are a less informative foundation for the autoencoders to encode and then decode from than other diverse explanations. However, translating into LI might lighten the task for the translator as there are fewer features to reproduce.

Of the methods with the same underlying concept, IG and VG outperform all other method pairs. KS and LI show varying success. They sometimes occupy strong ranks, other times placing on ranks in the middle. Gradient-based pairs from SG translate with varying success as well, while translation into SG is of poor quality almost regardlessly of the origin.

Non-aligning concepts rank in all kinds of ways. Often, translations into IG or VG perform well, and translations into SG perform badly once more.

Based on our experiments, drawing conclusions for all method pairs, that share an underlying concept as in H1, is not possible. It seems insufficient for methods to only share the basic approach to attributing feature importance. Our results point to the need for a task-dependent, close alignment of approaches.

Nonetheless, the best method pair in all translations does share an underlying concept, and so do KS and LI who often follow on IG and VG in the rankings. It seems that for successful translations a detectable structure is most important, as it might be the case for IG and VG, as well as the LI explanations with their many zeros. We can not really state, that the same concept is reason enough to assume a successful translation in comparison to non-aligning concepts.

Masking one third of features often brings improvement, especially, if we base the explanations on the spambase dataset or if linear regression is employed to translate. On the other hand, masking two thirds of the features even shows a drastic setback in some MSEs. Thinking back to the intuition the feature contribution results in Table 5.3 and 5.6 give us, this could be where much of the information, that the explanations carry, is removed. Additionally, masking one or two thirds might be too primitive of a cutoff. Perhaps, it would be better to employ a threshold based on how much percentage of the attribution scores the features carry. The low mean MSE baseline when a mask is employed, makes us skeptical, if the detected improvement in the MSEs can even be interpreted as ease of translation. It is possible, that the mask simply achieves to remove strongly negative values, as we for example often see in the value distributions of IG, SG and VG in Figure 5.1, thereby making a lower MSE much more easily achievable. Therefore, we can not state that a focus on a subset of features simplifies the translations in general, like we hypothesized in H2.

Chapter 7

Conclusion

In this thesis we explored structural aspects of the Disagreement Problem in explainable machine learning. Through translations of explanations from different attribution methods into each other, we investigated their overlap and differences of information.

We conducted the analysis with regard to the following hypotheses.

H1 The translation of one explanation into another is more successful when both methods align in their underlying concepts of being gradient-based or perturbation-based.

H2 It is easier to focus the translation on the features with the highest attribution scores of each method than a translation over the complete range of features.

We conducted a preliminary analysis to understand the constitution of the explanation sets stemming from the different methods and to examine if the explanations are separable from each other. We proceeded with pairwise translations of the explanation sets from different methods into one another via an autoencoder or with linear regression. Furthermore, we ranked the translation success of method pairs based on the mean squared error. Then, we masked one or two thirds of the less important features and repeated the translations.

Our preliminary results showed that explanation sets appear to be separable and that methods of the same underlying concept more often share characteristics than when they differ in their concept. However, the translations revealed that sharing a gradient- or permutation-based approach does not suffice to assume successful translations. As VanillaGrad and Integrated Gradients showed most aligning characteristics and their pair performed best throughout the translation, we suspect that task-dependent, close alignment of methods can result in structural similarities and ease of translation.

Masking one or two thirds of the features did not result in consistent improvements of the translation error.

In the future, we would find interest in a more exhaustive investigation of translations of explanations which include LIME explanations that are more easily compared to the other methods' explanations. We propose masking features with a less primitive threshold for future work.

Bibliography

Aïvodji, U. et al. (2019). “Fairwashing: the risk of rationalization.” In: *International Conference on Machine Learning*. URL: https://api.semanticscholar.org/CorpusID:59316669.

Ali, Sajid et al. (2023). “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence.” In: *Information Fusion* 99, p. 101805.

Alvarez-Melis, David and T. Jaakkola (2018). “On the Robustness of Interpretability Methods.” In: *ArXiv* abs/1806.08049. URL: https://api.semanticscholar.org/CorpusID:49352880.

Ancona, Marco et al. (2017). “Towards better understanding of gradient-based attribution methods for Deep Neural Networks.” In: *International Conference on Learning Representations*. URL: https://api.semanticscholar.org/CorpusID:3728967.

Bank, Dor, Noam Koenigstein, and Raja Giryes (2023). “Autoencoders.” In: *Machine Learning for Data Science Handbook*. Springer International Publishing.

Bhardwaj, Nitanshi and Gaurav Parashar (July 2024). “The Disagreement Dilemma in Explainable AI: Can Bias Reduction Bridge the Gap.” In: *PREPRINT (Version 1) available at Research Square*. DOI: 10.21203/rs.3.rs-4193128/v1.

Bordt, Sebastian et al. (2022). “Post-Hoc Explanations Fail to Achieve their Purpose in Adversarial Contexts.” In: *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. FAccT ’22. New York, NY, USA: Association for Computing Machinery, pp. 891–905. ISBN: 9781450393522.

Chen, Hugh et al. (2022). *Algorithms to estimate Shapley value feature attributions*. arXiv: 2207.07605 `[cs.LG]`.

Cuzzocrea, Alfredo et al. (2023). “Attribution Methods Assessment for Interpretable Machine Learning.” In: *Sistemi Evoluti per Basi di Dati*. URL: https://api.semanticscholar.org/CorpusID:261732578.

Han, Tessa, Suraj Srinivas, and Himabindu Lakkaraju (2022). *Which Explanation Should I Choose? A Function Approximation Perspective to Characterizing Post Hoc Explanations*. arXiv: 2206.01254 `[cs.LG]`. URL: https://arxiv.org/abs/2206.01254.

Hopkins, Mark et al. (1999). *Spambase*. UCI Machine Learning Repository. DOI: 10.24432/C53G6X.

Jesus, Sérgio et al. (2021). “How can I choose an explainer? An Application-grounded Evaluation of Post-hoc Explanations.” In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT ’21. Virtual Event, Canada: Association for Computing Machinery, pp. 805–815. DOI: 10.1145/3442188.3445941.

Koenen, Niklas and Marvin N. Wright (2024). "Toward Understanding the Disagreement Problem in Neural Network Feature Attribution." In: *Explainable Artificial Intelligence, Second World Conference, xAI 2024, Valletta, Malta, July 17–19, 2024, Proceedings, Part III*. Springer Nature Switzerland, pp. 247–269.

Krishna, Satyapriya et al. (2022). *The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective*.

McInnes, Leland et al. (2018). "UMAP: Uniform Manifold Approximation and Projection." In: *J. Open Source Softw.* 3, p. 861.

Müller, Sebastian et al. (2023). "An Empirical Evaluation of the Rashomon Effect in Explainable Machine Learning." In: *ArXiv* abs/2306.15786. URL: https://api.semanticscholar.org/CorpusID:259275144.

Neely, Michael et al. (2021). *Order in the Court: Explainable AI Methods Prone to Disagreement*. arXiv: 2105.03287 `[cs.LG]`. URL: https://arxiv.org/abs/2105.03287.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""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*. KDD '16. New York, NY, USA: Association for Computing Machinery, pp. 1135–1144.

Scikit-learn (2024). *sklearn.linear_model.LogisticRegression*. Accessed: 2024-08-18. URL: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps." In: *CoRR* abs/1312.6034.

Smilkov, Daniel et al. (2017). *SmoothGrad: removing noise by adding noise*. arXiv: 1706.03825 `[cs.LG]`.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks." In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17, pp. 3319–3328.

Watson, Matthew, Bashar Awwad Shiekh Hasan, and Noura Al Moubayed (2021). *Agree to Disagree: When Deep Learning Models With Identical Architectures Produce Distinct Explanations*. arXiv: 2105.06791 `[cs.LG]`.

Wolberg, William et al. (1995). *Breast Cancer Wisconsin (Diagnostic)*. UCI Machine Learning Repository. DOI: 10.24432/C5DW2B.

Zhou, Yilun et al. (2021). "Do Feature Attribution Methods Correctly Attribute Features?" In: *ArXiv* abs/2104.14403. URL: https://api.semanticscholar.org/CorpusID:233443847.

List of Figures

5.1 Feature value distribution per method. Each bar represents the frequency of values in one 0.2 interval over the range $[-1, 1]$. (breast-w) 17
5.2 Feature value distribution per method. Each bar represents the frequency of values in one 0.2 interval over the range $[-1, 1]$. (spambase) 19
5.3 FA for one third and two thirds of the top-k features. (breast-w) 20
5.4 FA for one third and two thirds of the top-k features. (spambase) 21
5.5 UMAP dimension reduction of the explanation sets. 22
5.6 Accuracy of the logistic regression classification on pairs of explanation methods. 23

6.1 MSE of translations of explanation method pairs for models 1, 2 and 3. (breast-w) 28
6.2 MSE of translations of explanation method pairs for models 1, 2 and 3. (spambase) 29
6.3 Ranking of MSEs of translations of explanation method pairs for models 1, 2 and 3. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts (breast-w). 31
6.4 Ranking of MSEs of translations of explanation methods for models 1, 2 and 3. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts (spambase). 32
6.5 MSEs of translations where none, one third or two thirds of the less important features have been masked. (breast-w) 34
6.6 MSEs of translations where none, one third or two thirds of the less important features have been masked. (spambase) 35

A.1 MSEs of translations via the simpler autoencoder architecture on breast-w and spambase based explanation sets 50
A.2 Ranking of MSEs of translations of explanation methods via the simpler autoencoder architecture. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts. 51
A.3 MSEs of translations where none, one third or two thirds of the less important features have been masked when the simpler autoencoder architecture is in use 52

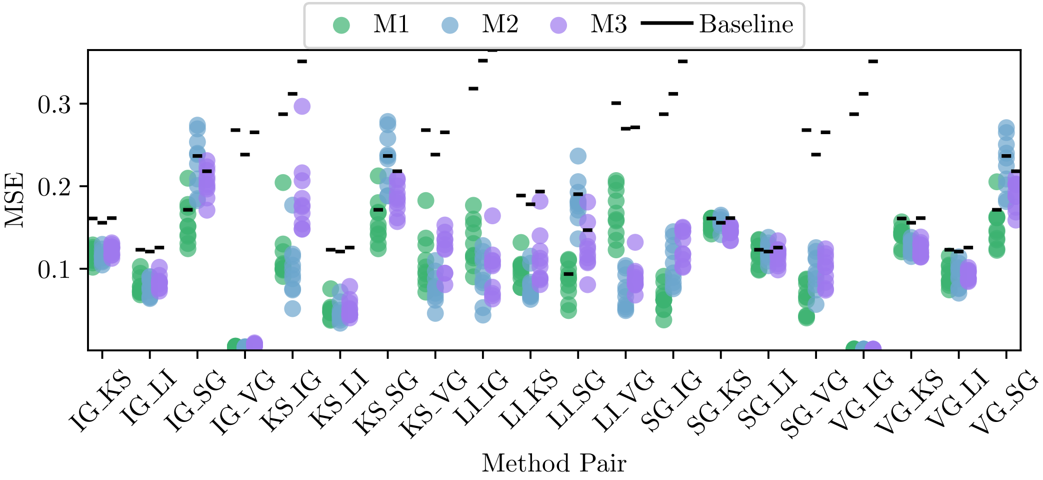
List of Tables

5.1 Dimensional variance of explanations. (breast-w) 18
5.2 Mean variance within explanations. (breast-w) 18
5.3 Average feature contribution to 80% of the summed feature score per method. (breast-w) 18
5.4 Dimensional variance of a few selected features. (spambase) 19
5.5 Mean variance within explanations. (spambase) 19
5.6 Average feature contribution to 80% of the summed feature score per method. (spambase) 19

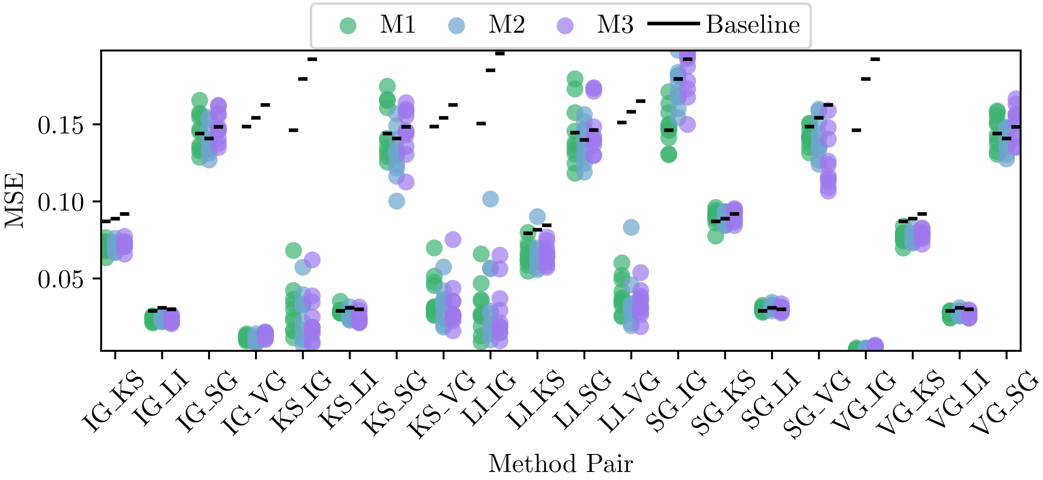
Appendix A

Additional Results

We include the additional results from chapter 6. The autoencoder with only a bottleneck of five neurons performed similarly to the other autoencoder across the translations, the rankings of the translations and when subsets of the features were masked.

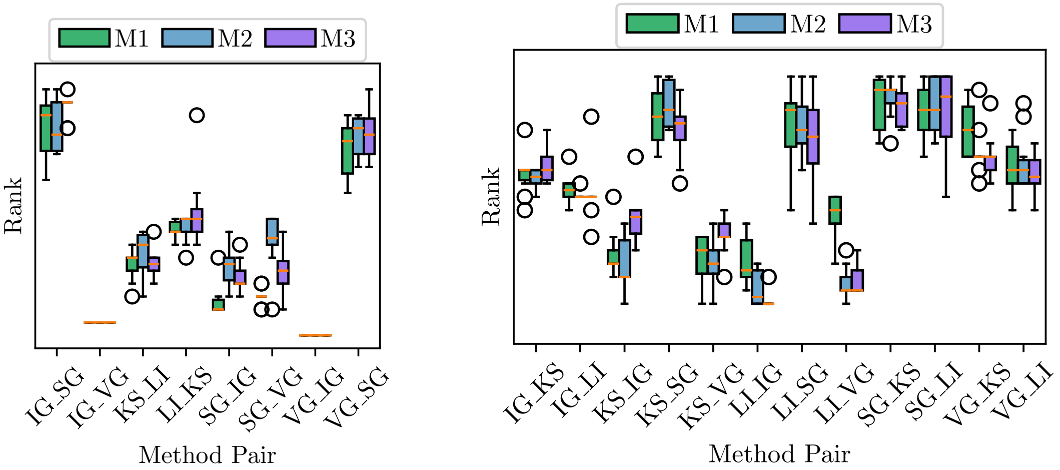


(**a**) breast-w



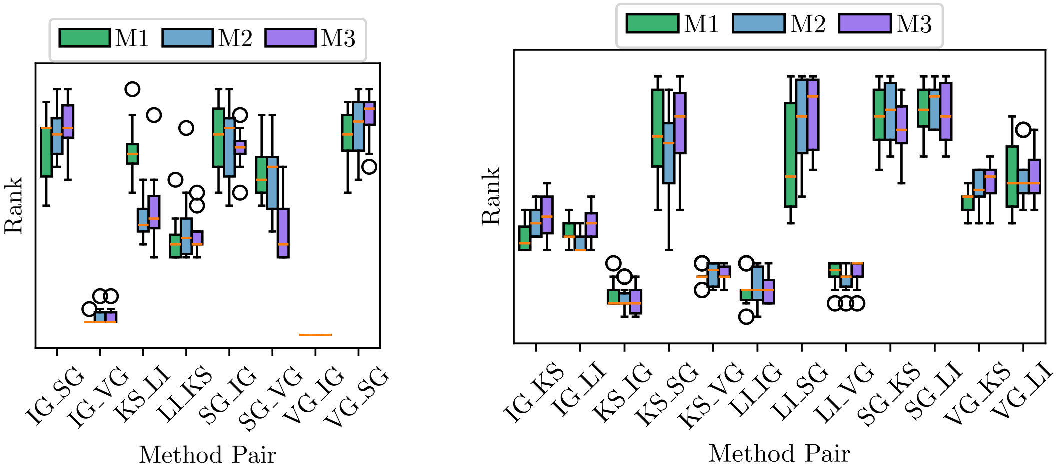
(**b**) spambase

Figure A.1: MSEs of translations via the simpler autoencoder architecture on breast-w and spambase based explanation sets



(**a**) Shared concepts (breast-w)

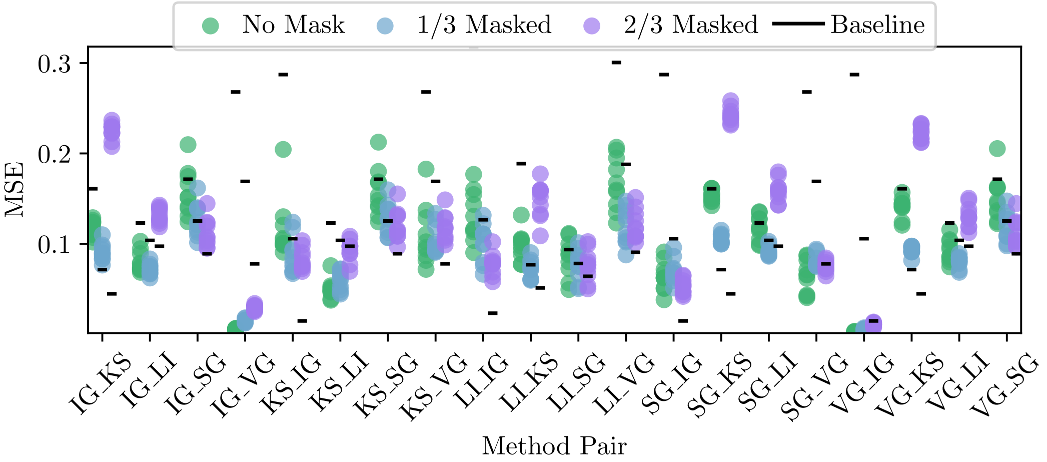
(**b**) Mixed concepts (breast-w)



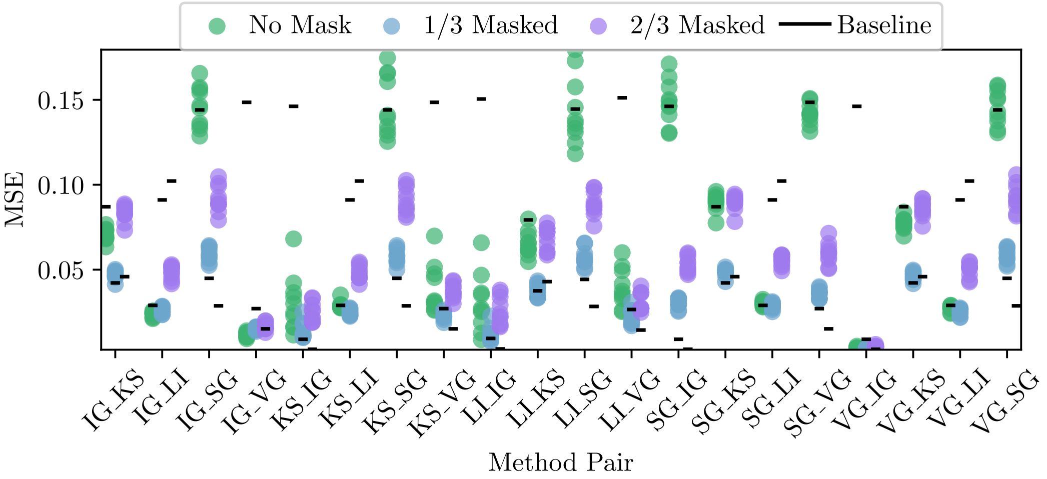
(**c**) Shared concepts (spambase)

(**d**) Mixed concepts (spambase)

Figure A.2: Ranking of MSEs of translations of explanation methods via the simpler autoencoder architecture. All three left plots show method pairs, that share an underlying concept, while the right plots show mixed underlying concepts.



(**a**) breast-w



(**b**) spambase

Figure A.3: MSEs of translations where none, one third or two thirds of the less important features have been masked when the simpler autoencoder architecture is in use