RPTU visual information analysis

Interactive Exploration of Model Predictions for Multivariate Data

---

Vom Fachbereich Informatik der
Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau
zur Verleihung des akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation
von

Jan-Tobias Sohns

Datum der wissenschaftlichen Aussprache
12. Februar 2026

Dekan
Prof. Dr. Christoph Garth

Vorsitz der Promotionskommission
Prof. Dr. Marius Kloft

Berichterstatter
Prof. Dr. Heike Leitte (RPTU)
Prof. Dr. Hans Hasse (RPTU)
Dr. Gunther H. Weber (Lawrence Berkeley National Laboratory, USA)

DE-386

Abstract

Decision-making processes increasingly rely on complex machine learning models, but their predictive power often comes at the expense of transparency, limiting our ability to understand and learn from predictions. Such models achieve their accuracy by exploiting multivariate feature relationships, yet these relationships remain difficult to interpret: high-dimensional spaces prevent direct inspection, dimensionality reduction can distort neighborhoods and decision boundaries, and existing tools rarely connect model behavior to domain knowledge. This dissertation addresses this gap by contributing three model-agnostic visual analytics techniques for multivariate tabular data, organized around complementary perspectives of model interpretation.

Input-based analysis Decision boundaries, i.e., regions where a model's prediction flips under multivariate input changes, are hard to explore in high dimensions. This dissertation presents an interactive system that systematically probes input perturbations using local linear maps, preserving input distances more accurately and revealing closer boundaries than prior approaches.

Relationship-based analysis Visual feature enrichment in non-linear dimensionality reductions struggles to balance multiple objectives, such as representing clusters, outliers and distortion. This dissertation introduces a topology-based augmentation method that effectively relates feature distributions to projection regions, simultaneously highlighting clusters, outliers, and ambiguities.

Knowledge-based analysis Matrix completion yields prediction matrices whose patterns are challenging to interpret beyond basic heatmap expansion and sorting. This dissertation proposes a hierarchical evaluation framework that aggregates and links these patterns to domain-knowledge not used during model building. Its application yielded insights that substantially advanced thermodynamic modeling.

All approaches are integrated in comprehensive visual analytics systems designed for interactive use. Case studies, stakeholder evaluations, and successful applications confirm their effectiveness in promoting the explainability of model predictions and domain insight. Together, these contributions establish structured methods for exploring multivariate feature spaces visually, driving measurable advances in both model and data interpretability.

Kurzfassung

Entscheidungsprozesse stützen sich zunehmend auf komplexe Modelle des maschinellen Lernens, doch deren Vorhersagekraft geht oft zu Lasten der Transparenz, was sowohl das Verständnis als auch den Erkenntnisgewinn einschränkt. Solche Modelle erreichen ihre Präzision, indem sie multivariate Zusammenhänge nutzen. Diese Zusammenhänge sind jedoch nach wie vor schwer zu interpretieren: Hochdimensionale Räume verhindern eine direkte Visualisierung, Dimensionsreduktion kann Nachbarschaftsverhältnisse und Entscheidungsgrenzen verzerren, und bestehende Lösungen verbinden das Modellverhalten selten mit Fachwissen. Diese Dissertation befasst sich mit dieser Lücke, indem sie drei modellunabhängige visuelle Analysetechniken für multivariate tabellarische Daten vorstellt, die komplementäre Perspektiven der Modellinterpretation bieten.

Eingabebasierte Analyse Entscheidungsgrenzen, also Bereiche, in denen sich die Vorhersage eines Modells bei multivariaten Eingabeänderungen umkehrt, sind in hohen Dimensionen schwer zu untersuchen. In dieser Dissertation wird ein interaktives System vorgestellt, das Änderungen der Eingabeparameter mithilfe lokaler linearer Abbildungen systematisch untersucht, und dabei Abstände zu Entscheidungsgrenzen genauer beibehält und nähere Grenzen aufzeigt als bisherige Ansätze.

Beziehungsbasierte Analyse Die visuelle Auswertung von Attributen in nichtlinearen Dimensionsreduktionen muss mehrere Ziele wie die Darstellung von Clustern, Ausreißern und Verzerrungen in Einklang bringen. In dieser Dissertation wird eine topologiebasierte Auswertungsmethode vorgestellt, die Attributsverteilungen effektiv mit Projektionsbereichen in Beziehung setzt und gleichzeitig Cluster, Ausreißer und Mehrdeutigkeiten hervorhebt.

Wissensbasierte Analyse Matrixvervollständigungsmethoden liefern Vorhersagematrizen, deren Analyse kaum über die visuellen Muster in sortierten Heatmaps hinaus gehen. In dieser Dissertation wird ein hierarchisches Bewertungssystem vorgeschlagen, das diese Muster aggregiert und mit Fachwissen verknüpft, welches bei der Modellerstellung nicht verwendet wurde. Ein Anwendungfall liefert Erkenntnisse, die die thermodynamische Modellierung erheblich vorangebracht haben.

Alle Methoden sind in umfassende visuelle Analysesysteme integriert, die für die interaktive Nutzung konzipiert sind. Fallstudien, Bewertungen durch Interessengruppen und erfolgreiche Anwendungsfälle bestätigen ihre Effektivität bei der Verbesserung der Erklärbarkeit von Modellvorhersagen und Fachwissen. Zusammen bilden diese Beiträge strukturierte Methoden zur visuellen Erforschung multivariater Merkmalsräume und führen zu messbaren Fortschritten bei der Interpretierbarkeit von Modellen und Daten.

Acknowledgments

I would like to thank my supervisor Prof. Dr. Heike Leitte for her unconditional support and trust throughout my PhD. Her mentorship not only guided me academically but also created an environment in which this dissertation became a far less stressful journey. Through her practical focus, she opened avenues that made this work profoundly interdisciplinary, for which I am deeply grateful.

I am equally grateful to Prof. Dr. Christoph Garth, who first introduced me to the field of visualization during my bachelor's studies and has continuously supported me ever since. His advice and encouragement have been invaluable throughout my academic journey.

My sincere thanks go to Dr. Gunther Weber for giving me the opportunity to work in an international environment, an experience that has greatly shaped my personal growth.

I would also like to thank Prof. Dr. Hans Hasse, Prof. Dr. Fabian Jirasek, and Dominik Gond for our collaborations on interdisciplinary projects, which not only amplified the impact of this work but were also a great source of inspiration and enjoyment.

I further want to thank the German Research Foundation (DFG) for funding this research through the priority program *SPP 2363 – Molecular Machine Learning* and the *IRTG 2057 – Physical Modeling for Virtual Manufacturing Systems and Processes*.

To all my colleagues in the visualization group, thank you for the fruitful discussions and the wholehearted support between everyone. In particular, I am grateful to Dr. Kilian Werner for the countless discussions and brainstorming sessions that made my time at the university both more productive and more enjoyable.

I owe heartfelt thanks to my family for their endless support and encouragement throughout my life, giving me both the freedom and the confidence to pursue my dreams.

Finally, my deepest gratitude goes to Annemarie for always having my back and supporting me in every possible way.

Contents

Chapter 1

Introduction

Machine learning models have become powerful tools for analyzing complex data, rapidly accelerating their integration into everyday life and a broad range of scientific disciplines. With growing fields of application, our ability to trust, understand and learn from the predictions of these models is becoming increasingly important [9, 10]. This dissertation addresses the challenge of interpreting machine learning predictions, focusing on how to visualize and understand the relationships between multivariate input features and model predictions.

Algorithmically, machine learning models operate by mapping inputs to outputs through patterns extracted from data. These patterns may reflect genuine associations, causal relationships, or spurious correlations [11]. Regardless of their origin, they encode valuable dependencies of the underlying process [12, 13, 14]. When aligned with the true structure of the process, such patterns can both enable accurate generalization and serve as a source of insight [15]. In principle, machine learning models thus hold potential not only as predictive tools but also as scientific instruments for discovery.

Real-world examples illustrate this promise: graph neural networks have uncovered novel chemical substructures relevant for drug design [16, 17], AlphaFold has predicted protein folding patterns long elusive to biology [18, 19], and matrix completion models have provided new perspectives on thermodynamic phase behavior [20], leading to more accurate models for industrial applications [6, 21]. As machine learning continues to spread across domains, new opportunities arise to uncover structural relationships that were previously too complex to grasp.

To leverage these opportunities, it is crucial to develop methods that allow us to interpret and understand the learned relationships between machine learning decisions and human-interpretable features [22], an approach referred to as *explainable machine learning* [23]. Visual analytics offers such an approach, combining interactive visualization with computational techniques to reveal structure in high-dimensional data [24]. By enabling users to explore relationships between inputs and predictions, these methods provide the means to explain complex model decisions and to extract scientific insight from the learned dependencies.

1.1 Problem Statement

A core challenge for explainable machine learning is the complexity of input data [24]. Most models are applied to *multivariate data* – data where each sample is described by multiple features whose combined effects determine the outcome. Such data can be thought of as spanning a high-dimensional space, where each dimension represents a different feature. While models learn mappings from these high-dimensional spaces to outputs, the relationships embedded in these mappings are often inaccessible: They are difficult to extract, visualize, and interpret.

Despite the progress in explainable machine learning, current approaches often reduce interpretation to individual feature effects (e.g., LIME, SHAP, PDPs [25, 26, 27]) or raw data visualization (e.g., scatterplot projections, counterfactuals [28, 29]) [24]. These methods rarely capture multivariate relationships or domain knowledge, limiting our ability to understand how features interact to create predictions [30]. In particular, three challenges arise when interpreting model predictions for multivariate data:

Multivariate dependencies remain hidden within high-dimensional input spaces. Existing visualizations struggle to represent complex interactions among multiple features, making it unclear how combinations of features influence predictions [31, 32]. In classification models, for example, decision boundaries partition the input space into regions of different outcomes. Without effective visualization, the continuous, multivariate nature of these boundaries cannot be explored, leaving users unable to understand how simultaneous changes in input features affect predictions [33].

Dimensionality reduction distorts relationships. While methods such as PCA, t-SNE or UMAP [34, 35, 36] can reduce high-dimensional data to a visualizable representation, they often introduce distortions that obscure connections to original features [37]. Non-linear projections are particularly problematic. They can create ambiguous regions where visual patterns cannot be reliably related to the learned data structure, leading to misinterpretation of model behavior [38].

Aligning model behavior with domain knowledge remains challenging. Existing tools rarely connect visual patterns back to meaningful domain structures, making it difficult to validate insights, integrate expertise, or uncover scientifically relevant relationships [39, 40]. For instance, models predicting pairwise relationships (e.g., chemical interactions) yield large prediction matrices, but current practice relies mainly on manual inspection of simple heatmaps [41]. Without methods that integrate domain knowledge into visual analytics, opportunities for discovery are lost.

In summary, current visualizations fail to capture multivariate dependencies, preserve feature context in projections, and integrate domain knowledge. These gaps limit our ability to learn from machine learning models beyond their predictive accuracy. This dissertation addresses these challenges by developing interactive visualization techniques that reveal how multivariate feature spaces relate to model predictions in interpretable and actionable ways.

1.2 Contribution

Towards this goal, this dissertation makes three major contributions structured around the three levels of analysis, each offering a complementary perspective.

Input-based analysis explores what needs to be changed in input space to obtain a different output. Decision boundaries mark interesting regions in input space where the prediction flips, but current analysis on individual features or distorted projections is limiting. This dissertation introduces local linear maps that capture *decision boundaries in multivariate input space*, making their structure accessible. These maps are integrated into an interactive visual analytics tool, enabling users to explore how simultaneous changes in input variables influences classification outcomes. Case studies reveal multivariate dependencies that were previously hidden and show both more accurate and closer decision boundaries than prior methods.

Relationship-based analysis visualizes the relationship of original features to non-linear projections. Visual feature enrichments in non-linear dimensionality reductions struggle to balance multiple objectives, such as representing clusters, outliers and distortion. This dissertation introduces a topology-based augmentation method that *effectively relates feature distributions to projection regions* and implement it within a small multiples framework with integrated quality control. Case studies and runtime evaluations demonstrate practical effectiveness in simultaneously handling projection ambiguity, clusters, and outliers.

Knowledge-based analysis connects structure in model predictions with domain knowledge. Matrix completion models produce valuable interaction matrices that lack systematic methods to relate patterns to domain knowledge beyond sorting and heatmap expansion. This dissertation proposes a hierarchical evaluation framework that aggregates and *relates these patterns to domain-knowledge* not used during model building. Applying this framework on the pairwise interaction of chemical substances yielded insights that substantially improved thermodynamic modeling.

Together, these contributions enhance the explainability of machine learning models by providing structured approaches for exploring complex relationships in multivariate data, advancing the field through a blend of theoretical innovations and practical implementations.

1.3 Structure of the Dissertation

This dissertation is organized to guide the reader from foundational concepts to the specific contributions. It begins with a background chapter that introduces key terminology and provides an overview of approaches in multivariate data visualization and explainable machine learning. This chapter establishes the context for the contributions presented in the subsequent sections.

The core of the dissertation is divided into three main chapters, each focusing on one of the contributions outlined above. Each of these chapters begins with a more specialized introduction to the relevant problem domain and the motivation behind the proposed approach, followed by a detailed description of the methodology and implementation. To demonstrate the practical value of the work, each chapter concludes with case studies illustrating the effectiveness of the proposed methods in real-world scenarios.

Thereafter, a short section on complementary contributions explores extensions and applications of the core methods. This includes applications in thermodynamic modeling as well as related methods in multivariate data visualization, highlighting the scientific impact of the work.

The dissertation concludes with a reflection on the contributions and a discussion of open challenges and potential avenues for future research in multivariate data visualization for machine learning predictions.

Parts of this dissertation present work that has been previously published and is based on collaborations with other researchers. The respective contributors are listed as co-authors at the end of each chapter. While I am the primary author and researcher of the studies presented in this dissertation, I gratefully acknowledge the support and contributions of my collaborators. In appreciation of their efforts, I will use the academic *we* when describing the research work.

Chapter 2

Background

The content of this dissertation lies at the intersection of explainable machine learning and multivariate data visualization. As the later chapters explore specific aspects of these areas in depth, this chapter provides the relevant background to contextualize the contributions that follow. To that end, we begin with an introduction to multivariate data visualization, as interactive exploration, and thus visual analytics, is the primary methodological focus of this work. We then turn to explainable machine learning, which defines the application scenario in which we employ these visualization techniques. We conclude by proposing visual analytics as a solution to the challenges occurring at this intersection of fields.

2.1 Multivariate Data Visualization

As introduced in the previous chapter, multivariate data is one of the most prevalent forms of data in both research and real-world applications. It arises in a wide range of domains, from patient records in healthcare and sensor streams in robotics to experimental measurements in engineering. Effectively analyzing and visualizing such data is essential for uncovering relationships, identifying patterns, and detecting anomalies. These insights are often key to informed decision-making and model development, thus we begin by defining multivariate data and then review the main strategies for visualizing such data, highlighting their strengths and limitations.

2.1.1 Multivariate Data: Definition and Challenges

Formally, multivariate data refers to datasets in which each observation is described by two or more variables. As such, this structure captures multiple characteristics of each subject simultaneously, allowing for the analysis of complex relationships. The variables can be independent, correlated, or causally related, depending on the nature of the data.

Mathematically, a multivariate dataset X consisting of n observations can be defined as

$$X = \{x_1, x_2, \ldots, x_n\}$$

where each observation $x_i \in \mathbb{R}^d$ is a vector with d variables:

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \quad \text{for } i = 1, 2, \ldots, n.$$

Thus, the dataset can be viewed as a collection of n vectors of dimension d, or equivalently as an $n \times d$ matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1d} \\ x_{21} & x_{22} & \ldots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nd} \end{bmatrix}$$

In data analysis, the variables associated with each observation are commonly referred to as *attributes* and are typically arranged as columns in a matrix, with each row representing an individual observation or sample. In the context of machine learning, these variables are usually called *features*, where each sample is represented as a vector of features used as the input to, or output from, a model. Because these concepts are algorithmically equivalent, we use the terms *variables*, *attributes* and *features* interchangeably throughout this dissertation. Likewise, instead of observations, we refer to *samples* or *instances*, which are more common terms in machine learning.

The most common form of representing multivariate data is within a table in the matrix format defined above. However, it is important to note that multivariate data can also take other forms, such as time series, graphs, text, or images. In these cases, the data may not be represented as a simple matrix but rather in more complex topological structures. For example, text data may include word embeddings, sentence structures, and semantic relationships; while image features need to account for pixel values, color channels, and spatial pixel patterns. Each of these forms induce individual requirements for visualization, which are typically not implied when talking about multivariate data. Consequently, this dissertation focuses on multivariate data in the sense commonly adopted in visualization research, that is, data where each sample is represented as a fixed-length vector of numerical attributes.

Specifically, the focus is on datasets where the number of variables d is significantly greater than two, commonly referred to as *high-dimensional* data. Although this term lacks a strict definition, it typically describes a scenario where the number of variables per sample makes manual interpretation difficult. In real-world scenarios, especially in ML-contexts, high-dimensional data is the norm, with datasets often containing dozens or even hundreds of variables.

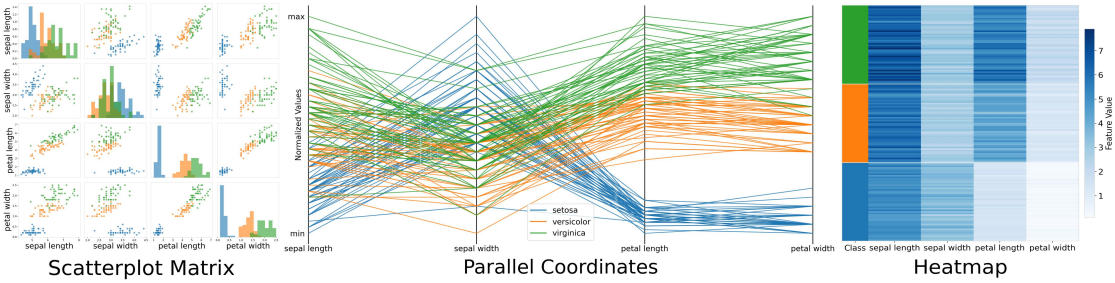
As mentioned in the introduction, each of these variables can be interpreted as an axis in a d-dimensional space. Consequently, each sample corresponds to a point in that space. The distances between points define their similarity or dissimilarity, which is quantified by a metric such as the Euclidean distance, Manhattan distance, or cosine similarity. The choice of metric significantly influences how structures are visualized and interpreted, which we will later demonstrate in Section 5.4.2.

However, as the dimensionality increases, the behavior of distance metrics changes significantly. The volume of the available high-dimensional space increases so quickly that the distribution of points inevitably becomes sparse. Consequently, the distances between points tend to converge, reducing their discriminative power for numerical analysis, a phenomenon called the "curse of dimensionality". The goal of visualization techniques is to convey the structure and, hence, the similarity of points, *visually* to mitigate this curse and rely on the superior human pattern recognition to interpret the data.

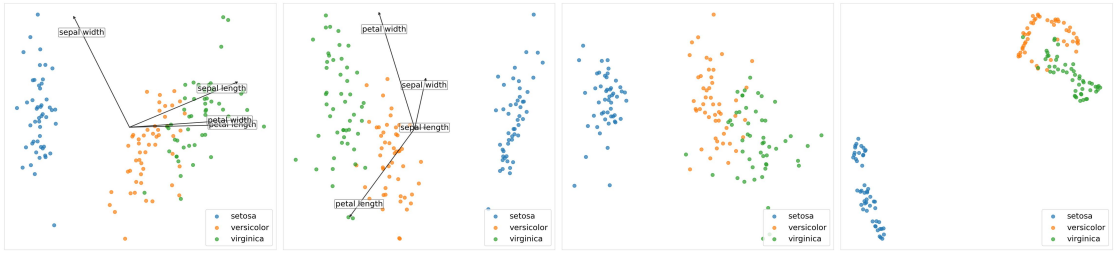
2.1.2 Strategies for Visualizing Multivariate Data

A variety of techniques have been developed to visualize multivariate data in a comprehensible and interpretable manner. These can broadly be categorized into *direct visualizations*, *dimensionality reductions* and *multi-view approaches* [43]. Figure 2.1 shows direct and dimension-reduced visualizations of the classic "iris" dataset [42], which contains 150 samples of flowers, each described by four numerical attributes, and grouped into three classes. The comparison highlights a trade-off: Each technique conveys specific aspects of the data's structure while introducing distinct interpretive challenges.

Direct representations map multivariate data to visual axes. These axes can be arranged within one plot as in parallel coordinates [44], heatmaps, and radar charts [45], or distributed in small multiples concepts [46], such as scatterplot matrices [47, 37]. The advantage of these techniques is that they use original data axes. However, multi-axis approaches scale poorly beyond 5–10 dimensions due to visual clutter and limited display space [37], as can already be recognized in Figure 2.1 (a) with just four dimensions. Moreover, the interpretation of heatmaps is highly dependent on axis ordering [48], an aspect that visualization research hardly considered from a practical side so far [49]. Chapter 5 addresses this gap by introducing an analysis framework for pattern-based ordering in application heatmaps, incorporating domain knowledge evaluation that previously was handled only via additional heatmap columns [41].



(a) Direct visualizations



(b) Dimensionality reductions

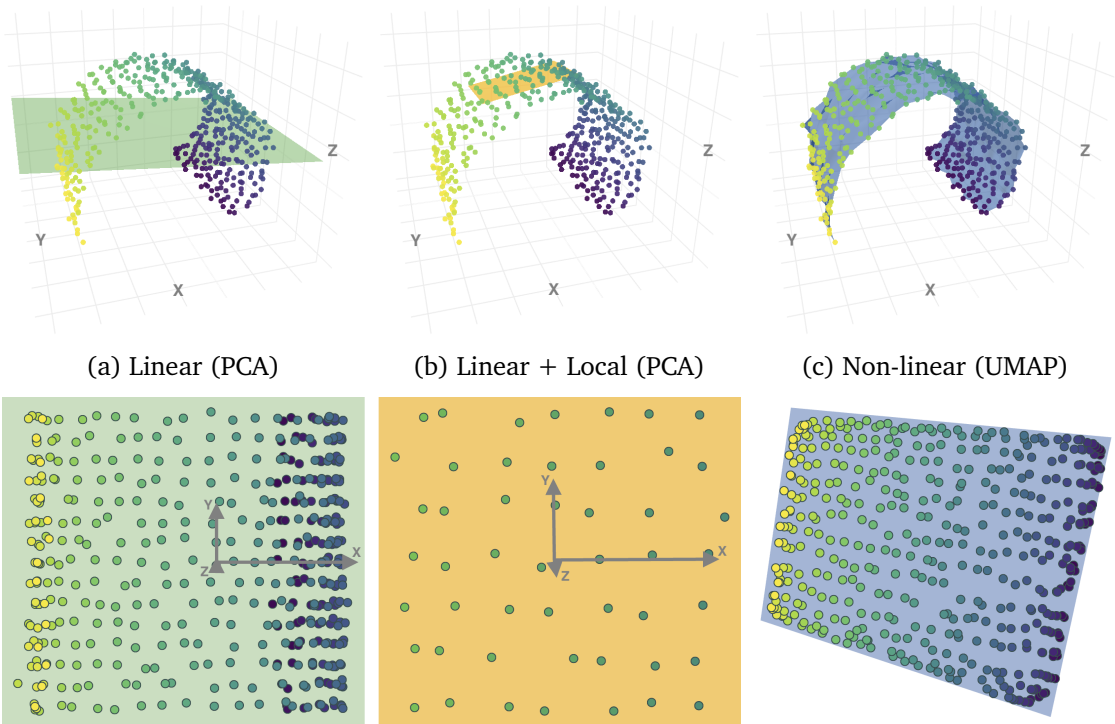
Figure 2.1: Multivariate data visualization strategies on a 4D *iris* dataset with three classes [42]: (a) Direct visualizations keep original axes but scale poorly with the number of dimensions. (b) Dimensionality reduction techniques can handle higher dimensions, but their projective distortion requires caution during interpretation. Multi-view approaches combine multiple data representations, each highlighting different aspects.

Dimensionality reductions avoid the scaling and ordering issues altogether by projecting data into a lower-dimensional space, typically 2D, while attempting to preserve its structure. They are commonly divided into *linear* and *non-linear* methods, which differ in their assumptions about the data structure and the relationships they preserve. To illustrate these differences, we use half a revolution of the "Swiss Roll" dataset [50], consisting of 450 points arranged on a bent surface in 3D space, in Figure 2.2.

Linear techniques construct projections based on linear combinations of the original variables, optimized for goals such as maximizing variance (PCA [34]) or class separability (LDA [42]), cf. Figure 2.1 (b) (left). Regardless of the optimization objective, the result is always a projection onto a flat 2D plane defined by the chosen linear components, as illustrated in Figure 2.2 (a). Such methods are computationally efficient and considered interpretable, since they preserve linear attribute axes (gray arrows). However, the Swiss Roll example also highlights their limitation. Points at both ends of the roll overlap, reflecting the inability of a single plane to capture curved structures. Linear methods are therefore only suitable when the data has an approximately linear structure; in more

complex cases, such as the Swiss Roll, they risk substantial information loss.
A common remedy is to confine the projection locally, which better adapts to complex structures [50], as shown in Figure 2.2 (b). While this local confinement comes at the cost of losing a global perspective, it is still useful for exploring local neighborhoods in context of original attributes. Chapter 3 employs this concept to create interpretable maps of a model's input space around a specific instance. Joining small local projections to a single 2D map, as in LLE [50], looses the context to original attribute axes, but forms the basis of the next class of methods.

Non-linear techniques assume that high-dimensional data lies on a lower-dimensional manifold embedded within the high-dimensional space, a topological concept known as the manifold hypothesis [51]. The most common assumption is a 2D manifold, where each data point has a small neighborhood of other points that locally resemble a 2D plane, forming a continuous, but potentially bent surface, as in the Swiss Roll. Figure 2.2 (c) demonstrates on the UMAP algorithm [36] that non-linear methods aim to recover the structure of this surface (highlighted in blue) while flattening it to two di-



(a) Linear (PCA) (b) Linear + Local (PCA) (c) Non-linear (UMAP)

Figure 2.2: Comparison of projection types on a "Swiss Roll" dataset [50] from 3D (top) to 2D (bottom): (a) Linear reductions project onto a 2D plane that preserves interpretable axes but fails to capture the curved structure. (b) In a locally confined area, linear projection can represent the structure correctly. (c) Non-linear projection projects onto a surface that preserves the manifold structure but looses interpretable axes.

mensions. Methods differ in how they define the neighborhood relationship between points, either using pairwise distances directly (MDS [52]), or using intermediate structures like graphs (UMAP [36]) or probability distributions (t-SNE [35]).
Nowadays, non-linear techniques are commonly used to represent latent spaces of machine learning models, e.g., in Autoencoders [53], because their assumptions are less strict. However, they are often computationally expensive and sensitive to their algorithmic neighborhood parameters [54]. Most importantly for this dissertation, they are harder to interpret because unlike linear techniques, they do not retain linear attribute relationships due to their non-linear transformation [33]. While solutions exist to recover these relationships through static augmentations [43] or interactive summaries [55], studies reveal that they still comprise a trade-off between conveying clusters, outliers and ambiguity [43, 38]. Consequently, Chapter 4 introduces an interactive augmentation technique to address this challenge.

Multi-view approaches integrate multiple linked representations of the same data, with each view highlighting different aspects or structures. They combine the strengths of multiple techniques, mitigating the limitations of individual methods, e.g., the information loss of projections in Figure 2.1 (b) by including the direct representations in Figure 2.1 (a). Multi-view systems have been successfully applied in various domains, including graph [56], cluster [57], and correlation analysis [58]. The main benefit is their ability to provide a more comprehensive understanding of complex datasets, particularly when no single visualization can capture all relevant aspects. This dissertation employs multi-view approaches in all three later chapters to deal with the complexity of multivariate model predictions.

In conclusion, we deduced three takeaways for the visualization side of interpreting multivariate model predictions that are addressed in this dissertation:

- **Linear projections are more interpretable, but build on assumptions** Chapter 3 explores a model's input space through careful creation of local linear maps, opening up the interactive exploration through a multi-view tool.
- **Non-linear projections require effective links to original attributes** Chapter 4 introduces a direct augmentation technique for non-linear projections to regain the interpretability of attribute relationships.
- **Heatmap analysis is sorting-dependent and lacks enrichment techniques** Chapter 5 links multiple direct views of domain knowledge attributes with an algorithmic analysis of prediction matrices advancing our data understanding.

2.2 Explainable Machine Learning

While visualization techniques provide valuable insights into the structure of high-dimensional data, in many practical scenarios the ultimate goal is to learn relationships between input and output variables, e.g., to distinguish flowers by their phenotypical attributes. Machine learning methods are well suited for this task, as they can extract complex, often non-linear dependencies directly from data. However, the resulting models are frequently opaque, making it difficult to understand how specific predictions are derived. This lack of transparency has motivated the fields of interpretable machine learning and explainable artificial intelligence (XAI), which aim to make model reasoning accessible and comprehensible to humans, enabling both a deeper understanding of the learned relationships and more trustworthy application in practice [59]. While this dissertation focuses on the subproblem of explaining machine learning predictions, our proposed approaches have to be considered within the broader context of XAI. In the following, we discuss the motivation for explanations, the key concepts and axes along which explanation methods can be classified, as well as the challenges that arise when explaining multivariate machine learning predictions.

2.2.1 Why Explanations Matter

A central challenge in machine learning is that models are typically trained to optimize predictive accuracy rather than to capture true causal relationships [60]. This optimization objective can lead to situations where a model generates correct outputs while relying on unintended or undesirable patterns [11]. Explanations aim to reveal these underlying patterns, allowing us to assess whether a model's reasoning aligns with our expectations. This is a crucial step in ensuring that a model is not only accurate but also trustworthy, i.e., aligned with domain knowledge. In the literature, this concept is often described as a model being *right for the right reasons* [61].

Explanations are thus essential whenever models make high-stakes decisions but may rely on embedded biases [62] or spurious correlations [11]. Consider the example of an automated loan approval system, where decisions are made based on data entered into an application form. Rather than simply presenting the final decision, providing an accompanying explanation offers benefits to multiple stakeholders. For instance, bank employees can verify that the model's decision was fair [63]. They might confirm that the loan was denied due to insufficient income, rather than being influenced by sensitive features such as the applicant's birthplace. In turn, the applicant benefits in that it potentially reveals actionable steps, such as increasing their reported income before

reapplying [29]. Finally, the system developer can use explanations to assess the fairness and robustness of the model or to reconsider which input fields should be included in the application form before deploying the system in practice [61].

To phrase this example more general, the need for explanations can be categorized into three scenarios. The first scenario involves *assessing the reliability of a specific prediction*: Stakeholders need to be able to trust that the correct factors led to a particular decision, especially in domains where automation can have significant consequences, like grading, justice, and medicine. The second scenario is similar, but focuses on *influencing or changing specific predictions*: Understanding the model's reasoning allows for targeted interventions or data modifications to achieve desired outcomes. The third scenario centers on *improving our prediction and data models*: Explanations can reveal weaknesses, biases, or unexpected behaviors that guide model development and refinement.

However, not every prediction is accompanied by a corresponding explanation. This limitation arises from the complexity of modern machine learning models. While most machine learning models from just a decade ago were still small enough to be explained through manual inspection of their parameters and decision logic [64, 65, 66, 67], basically all modern models can be characterized as non-interpretable due to their algorithmic complexity, architectural sophistication, or sheer scale [68, 69]. The rise of these huge and complex models has fundamentally shifted the landscape of model interpretability, introducing the field of XAI [29, 69].

2.2.2 Key Concepts and Terminology

Before presenting the solution strategies to these challenges, we briefly define the terms and concepts relevant to this dissertation. The field of XAI covers a spectrum of related terms that are often used interchangeably, but have slightly different connotations.

Interpretability is the ability to understand the decision-making process of a machine learning model. It refers to the degree to which a human can comprehend the underlying logic of the model [70, 71]. Interpretability can be understood as how *transparent* a model is.

Explainability is the process of providing reasons for a model's predictions. It refers to techniques that compute and present reasonings for a model's decisions in a human-understandable way [70]. Explainability can be seen as a subset of interpretability that focuses on the *communication* of decision reasoning, while potentially trading transparency for usability [71].

Explainability algorithms, so-called **explainers**, produce **explanations**. Individual explanations provide insights into how a model arrived at a specific prediction, though they may be summarized to broader statements. They aim to communicate the influential factors of a prediction to help users align this reasoning with the expected behavior. Therefore, they allow users to act adequately – by trusting or improving the model.

Ever since the introduction of the EU's General Data Protection Regulation (GDPR) in 2016 [72], the need for explanations has been enforced by law. The GDPR mandates that individuals have the right to an explanation when subjected to automated decision-making processes. The ensuing surge of XAI research has revealed the potential of explanations to improve transparency, trust, and model performance [71]. Consequently, there has been a growing demand to develop explanation methods for machine learning models that simultaneously grew in size and complexity [29, 69].

This quickly exposed the limitations of existing methods, that relied on the trade-off between predictive power and interpretability [73]. Simple models, such as linear regression or decision trees, are inherently interpretable. They are called **white box models** due to their transparent decision logic. However, their relatively simple architecture often fails to represent complex processes, rendering them insufficient for most real world problems [68]. On the other hand, models capable of solving realistic tasks, such as ensemble methods or deep neural networks, are too complex to be interpreted directly. Due to their opaque decision logic, such models are referred to as **black box models**.

We conclude that true interpretability is only possible for simple models, yet real-life problems are solved by complex, intransparent models. XAI research aims to bridge this gap by either providing intrinsically interpretable, yet competitive models or explaining the decision process of given models. This dissertation focuses on the latter: providing explanations for given machine learning predictions.

2.2.3 Axes of Explanations

The concept of explanations can be decomposed on four independent axes: *time of explanation*, *scope of explanation*, *dependence on model type*, and *type of explanation*. An overview and illustration of these axes is given in Figure 2.3.

Time of explanation distinguishes whether explanations are generated before or after the model has been trained.
Intrinsic explanations are explanations that are built into the model during its design and training phase. The explanations are sometimes also called *ante-hoc* explanations, as the

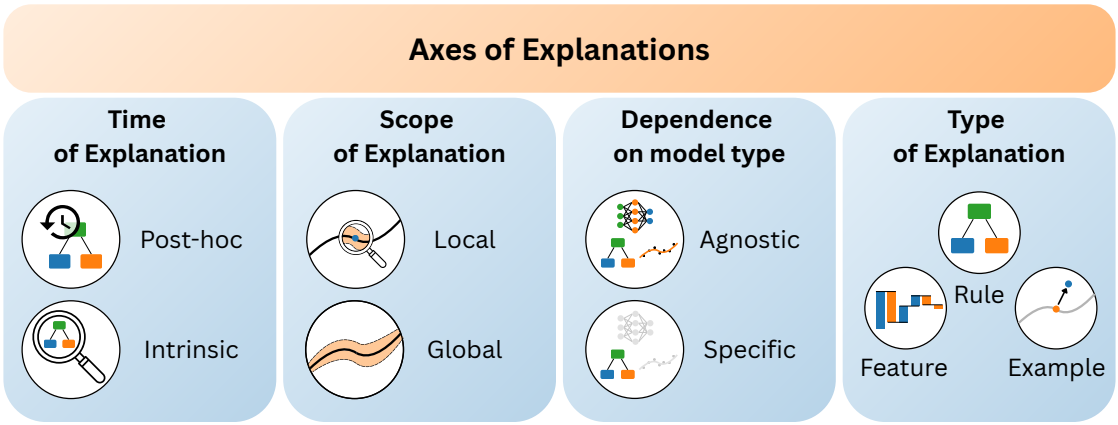


Figure 2.3: Axes of explanations: The four axes of explanations are time, scope, model dependence, and type. Each axis defines a different aspect of the explanation, which can be combined to create a wide range of explanation methods.

explanation is considered *before* evaluating the model. Most intrinsic explanations are inherent to the model due to simple architecture, such as Linear Regression, Decision Trees, and Generalized Additive Models [74]. As the field increasingly shifts towards massive black-box models, achieving intrinsic interpretability has become substantially more challenging without sacrificing performance [73].

Post-hoc explanations aim to explain the model's behavior *after* the training has finished. Post-hoc explanations do not compromise the model's accuracy, but rely on approximations of the model's more complex internal reasoning. Post-hoc explanations dominate XAI research, including this work, due to their flexibility in not requiring prior consideration during model design. It has to be noted though, that the approximation step common in post-hoc explanations means we can't always guarantee a complete or truthful representation [75].

Scope of explanation refers to whether an explanation targets an individual prediction or provides insight into the model's overall behavior.

Local explanations aim to clarify what led to a specific decision for a particular input instance. They answer questions such as *Which features contributed most to this loan denial?* or *Why did the model classify this email as spam?*, but the explanation can vary in complexity. Simple approaches trace the model's decision path for that instance, e.g., by analyzing the model parameters or decision rules [64]. More complex approaches approximate the model behavior in the vicinity of the instance with a simpler surrogate model, e.g., a rule-based one [76] or a linear one as in LIME [25].

Global explanations focus on understanding the model's reasoning in general, providing insights into the overall relationships between input data and model predictions across

the entire dataset. Algorithms often summarize the model's behavior by aggregating local measures across all instances, most prominently feature influence scores such as permutation importance [77], SHAP [26], and variants of partial dependence plots (PDPs, ALEs, ICEs [27, 78, 79]). While Chapter 3 focuses on local views of individual instances, Chapter 4 and Chapter 5 emphasize global prediction and data understanding.

Dependence on model type captures whether an explanation method is applicable across different types of model architecture or is designed for a specific one.
Model-agnostic explanations treat the model as a black box and can describe any machine learning model, independent of the underlying architecture or complexity. They are typically – yet not necessarily [80] – applied post-hoc to explain predictions of a finalized black box model. Model-agnostic algorithms include the previously mentioned surrogate models, but also simpler methods like Leave-One-Out importance [81] or representative instances [82, 29]. Techniques can be summarized in that they rely on statistical sampling with altered input variables. Chapter 3 employs such a sampling-based approach to explore the input space of any given prediction.
Model-specific explanations are tailored to specific types of models only. Algorithms leverage model-specific characteristics to provide more accurate and detailed explanations. Though intrinsically interpretable models lend themselves for model-specific explanations, they can also be specific to an architecture or training process [83]. Model-specific techniques are diverse, ranging from tree-specific variants of the model-agnostic SHAP [26], over the activation of individual nodes in neural networks [65], to how specific pixels travel through the network [84]. Chapter 4 introduces explanations specifically for non-linear projections, and Chapter 5 focuses on prediction matrices.

Type of explanation distinguishes between explanations based on their form and content [85]. **Feature-based** explanations highlight which input variables were most influential in driving a prediction. **Example-based** explanations provide representative data instances to illustrate model behavior. **Rule-based** explanations express the decision logic through interpretable *"if-then"* statements. Each type serves different stakeholder needs and use cases, from increasing trust to model debugging and improvement. Examples for each of the explanation types have been mentioned throughout the previous paragraphs. Feature-based explanations are used throughout Chapter 4 and Chapter 5, while example-based explanations play a central role in Chapter 3. Therefore, we further distinguish between *counterfactual* and *exemplary* instances. Counterfactuals provide a counterexample to a given decision, showing differences to an *opposite* outcome [29], while exemplars present a *positive* example, so a prototype of the outcome [82].

2.2.4 Challenges in Explaining Multivariate ML Models

Our analysis of studies revealed that explainable machine learning still faces many challenges, including the effective use of contrastive explanations, the composition of multiple explanation types at different levels of abstraction, and scalability to large models and datasets [86, 68]. While these will be discussed on the respective cases in later chapters, four challenges are central in context of this dissertation's overarching goal of explaining machine-learning-based predictions for multivariate data.

- *Fidelity of Explanations* [86, 71, 71]: faithfully conveying multivariate behavior requires accurate representations.
- *Conveying Feature Dependencies* [68, 70, 43]: multivariate data requires multivariate explanations.
- *User-Centered Explanations* [86, 68, 70]: tailoring explanations to the needs, expectations, and expertise of human users.
- *Integration of Domain Knowledge* [86, 87, 88]: incorporating expert insights to enhance explanation relevance and accuracy permitting multidisciplinary research collaboration.

Fidelity of Explanations An increasing challenge is ensuring that explanations are faithful to a model's true behavior rather than merely plausible or convincing [9, 89]. The recent success of large language models illustrates this issue: while they present explanations alongside outputs, these are rarely verified to reflect actual reasoning, often leading to hallucinations where outputs, explanations, or both are misleading [90, 91]. This problem extends beyond language models, as many explanation methods approximate already probabilistic ML models, introducing additional uncertainty through a chain of approximations [87]. To minimize this pitfall, this dissertation emphasizes explanation methods grounded directly in a model's input–output behavior, thereby maintaining higher fidelity and reducing the risk of false transparency [71].

Conveying Feature Dependencies However, direct inspection of input-output behavior is challenging due to the high dimensionality of the data involved. As discussed in Section 2.1, univariate analysis often fails to capture feature dependencies, yet the complexity of real data impedes straightforward multivariate visualization. Overcoming this requires advanced techniques from multivariate data visualization, which are specifically designed to reveal patterns, dependencies, and anomalies in high-dimensional spaces. This dissertation draws on such techniques to develop multivariate explanations that reflect the model's actual behavior. The following chapters examine three such approaches in detail.

User-Centered Explanations The third challenge arises from the fact that explanations are created for human users. However, users differ significantly in their needs and expectations, depending on their background, level of expertise, and the context in which explanations are used [92, 93]. As discussed above, explanations already serve various purposes – such as debugging, model understanding, data understanding, and validating trust – but they are often static and predefined [68, 87]. To address this, our tools incorporate interactive exploration, allowing users to engage with explanations in a visual interface, progressively extracting the information most relevant to their specific needs [86].

Integration of Domain Knowledge The last and least addressed challenge is the integration of domain knowledge into explanations. Domain experts possess valuable insights that can significantly enhance the relevance and accuracy of explanations [88]. However, effectively incorporating this knowledge into explanation methods remains a complex task, often requiring multidisciplinary collaboration [86]. This dissertation emphasizes the importance of domain knowledge in evaluating and refining explanations. In collaboration with thermodynamic modeling experts, each chapter includes case studies on domain data to ensure practical relevance. Chapter 5 explicitly integrates domain knowledge into the analysis of prediction matrices, demonstrating how expert insights can guide model interpretation and improvement.

In summary, these challenges build the foundation of the XAI side of dissertation, with each chapter contributing specific methods and case studies to address them.

- **Multivariate explanations of feature spaces** Chapter 3 examines how individual predictions can be changed to achieve alternative outcomes, revealing not only multivariate user-specific action paths but also structural weaknesses through direct model sampling.

- **Faithful analysis of features in distorted projections** Chapter 4 introduces an interactive approach to trace individual and cluster-level distributions of features in non-linear projections, enhancing transparency of model mechanisms and multivariate data dependencies.

- **Integrating domain-knowledge in prediction matrices** Chapter 5 presents a framework for uncovering feature dependencies in prediction matrices through aligning them with domain knowledge, thereby improving data understanding and guiding future model development.

2.3 Visual Analytics as a Bridge

The preceding sections have outlined the two methodological perspectives of this dissertation: visualization techniques for multivariate data and explanation methods for machine learning predictions. While each of these fields has developed its own strategies and tools, their synthesis is essential to address the challenges of understanding predictions in high-dimensional data spaces [69, 10, 70]. This synthesis can be achieved via *visual analytics*.

Visual analytics *"combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision-making on the basis of very large and complex data sets"* [94]. It integrates computational methods such as machine learning and statistical modeling with interactive visualization, leveraging the complementary strengths of automated analysis and human interpretation: machines provide scalability and computational efficiency, while humans contribute contextual judgment and sensemaking.

Visual analytics approaches are particularly valuable for high-dimensional datasets and studies show that they are widely used to enhance the explainability of machine learning [69, 10, 70]. They have been applied to various aspects of explainability, including feature importance [67], feature engineering [95], and model debugging [96]. However, other aspects, like *decision boundary visualization* [97, 98], *feature enrichment in projections* [38, 99], and *domain-knowledge evaluation in matrices* [41, 100], are still predominantly analyzed in static or single-view settings. We introduce visual analytics tools for all three of these latter aspects in the following chapters, demonstrating how they better account for the complexity of high-dimensional ML data.

In this chapter, we identified weak support for interpreting machine learning predictions for multivariate data that this dissertation addresses through structured approaches in three areas:

- **Multivariate analysis of decision boundaries in input spaces** Most approaches emphasize single-attribute explanations or aggregated feature rankings, rather than explicitly visualizing how multiple attributes jointly shape predictions. Chapter 3 introduces a visual analytics tool including local linear maps to provide interpretable views of decision boundaries, thereby revealing not only multivariate user-specific action paths but also structural weaknesses.

- **Effective links between projections and original features** Dimensionality reduction views are frequently used to show model embeddings or latent spaces, but these are often detached from the original attributes, making it difficult to interpret the meaning of structures in low-dimensional projections. Chapter 4 presents topology-based augmentations to effectively trace feature distributions in nonlinear projections, regaining the interpretability of multivariate feature relationships.
- **Knowledge-based analysis of prediction matrices** Matrix completion models produce valuable interaction matrices that lack systematic methods to relate patterns to domain knowledge. Chapter 5 proposes a hierarchical evaluation framework that connects direct views of domain knowledge with an algorithmic analysis of prediction matrices, improving data understanding and guiding future model development.

Chapter 3

Visualizing Decision Boundaries for Counterfactual Reasoning

An emerging direction of research aims at generating explanations through visual representation of local or global model behavior [101, 28, 102]. Without limiting the explanation to a certain architecture, it has to be based on probing the black box model's decision function. In classification models, the interesting section of the decision function is where the predicted class changes, which is called the *decision boundary*. This decision boundary can be described by samples [29, 28, 103] or continuous maps [104, 105, 106].

For a given input to a prediction model, similar inputs for whom a different outcome is predicted by the model are called counterfactual examples [29]. In particular, local samples of the decision boundary with regard to a given data instance are counterfactuals. While counterfactuals do not explicitly shed light onto model-internal factors leading to the prediction, they provide insight into what would need to change to generate a different outcome. As humans inherently deduce their internal explanations from comparisons [107], counterfactual- and therefore decision boundary-reasoning is a preferential explanation approach [29, 108].

Global samplings of decision boundaries extract scatterplot projections [15, 109, 110] or subspaces [28, 103, 31] from the high-dimensional input space that visually separate a given dataset color-coded according to class affiliation. The decision boundary then resides in the interval between instances having different class labels.

By regularly sampling the decision function, continuous maps provide an explicit insight into decision zones instead of just samples. Therefore, the decision boundary can be read exactly, even if no samples are nearby. Currently, these maps cover either univariate changes to an instance [111] or the whole dataset under multivariate changes mapped to 2D. [104, 112, 113]. The multivariate maps employ non-linear dimensionality reduction, since linear projections fail to capture non-linear manifolds of full datasets [114]. While the topology of the decision boundaries is preserved by the non-linear reductions, their shape and their distance to explained samples are distorted [33]. However, when

limiting the reduction to *local* subsets, the interactive exploration with *linear* embeddings has been shown to produce excellent explanations [115, 28].

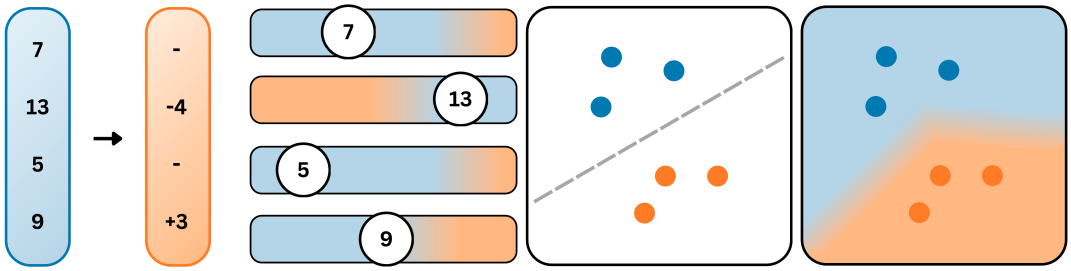
In this chapter, we propose a framework for the visual exploration of high-dimensional decision functions that enables the visual identification of feasible instance-based explanations through counterfactuals. Local linear maps are created around an instance for answering *"What-If"* questions about the shape and distance of nearby decision boundaries. The maps are complemented with the mentioned non-linear and one-dimensional decision boundary techniques to create a comprehensive interactive framework aimed at classifiers with 3 to 30 input variables.

3.1 Related Work

Research on explaining decision functions in machine learning models has produced a broad range of solution approaches. They can be summarized by their form of explanation: explanation through counterfactuals, visual model evaluation, boundaries in labeled datasets, and decision maps. In addition to a discussion of existing techniques comparable to our work, schematics of the approaches are presented in Figure 3.1.

Counterfactuals Currently, most of the machine learning literature is united under the concept that computing a counterfactual is an algorithmic optimization problem [29, 116, 117, 118]. However, identification of optimal counterfactuals is NP-hard [119] and the definition of optimality varies on a case-by-case basis [120]. Presenting a diverse set of counterfactuals instead increases the chances that an applicable example is found [121, 122]. Still, explanations through algorithmic counterfactuals are missing the flexibility, interactivity [120] and context [123] a visualization can provide.

Visual Model Evaluation In recent years the analysis of machine learning models has shifted from raw statistical measures to interactive tools that present the model's decision behavior. Ming et al. [124] approximate the decision space of black box models with global linear rules that can be visually aligned with human understanding. The approach was extended to provide both local and global explanation with visual rules [101], which limited the application to random forest models. Cheng et al. [125] proposed a similar iteration of scoped rules [82, 126, 127] on interactively refined subgroups that are evaluated over univariate counterfactuals. They also provide an interface to communicate and influence diversity in instance-specific counterfactuals.



(a) Counterfactual (b) Univariate Analysis (c) Labeled Dataset (d) Decision Map

Figure 3.1: Approaches of explaining decision boundaries.

While a common approach is to abstract from the complex model to an easier surrogate model [25, 124], the decision space can also be probed explicitly starting from an instance. The What-If-Tool [128] and Prospector [111] let a user probe the model response under manual perturbations to an instance. The former's focus is on model evaluation through a test set and therefore requires trial-and-error probing in text fields while the latter aids probing by showing model predictions under univariate changes in a colormap as illustrated in Figure 3.1 (b). We extend this analysis to multivariate changes.

Boundaries in Labeled Datasets Prediction models are typically evaluated on a discrete test set of labeled data instances. Hence, explaining boundaries between these labeled instances is parallel to the problem addressed in this chapter. As datasets usually comprise many dimensions, this task is often reduced to an interactive scatterplot exploration through dimension reduction for which both linear [110] and non-linear [31, 129] tools have been proposed. Ranking the possible perspectives allows filtering for interesting ones [130]. Returning to the issue of boundaries, Ma et al. [28] analyze a labeled dataset to compute a set of local linear boundaries that approximately separates the sample classes, cf. Figure 3.1 (c). While they generate sparse abstractions of the boundaries between two classes through sub-sectioning, this chapter focuses on instance-based interactive maps to support explanations through contextual counterfactuals.

Projections of labeled datasets have also been applied to evaluate prediction models in linear [128] and non-linear embeddings. The relevant non-linear embeddings are integrated into application-specific frameworks. Their aims vary from improving class separability and thereby model performance through feature selection [103], over latent space interpolation between two high-dimensional samples [15], to inspecting model behavior on new samples during transfer learning [30]. Mazumdar et al. [131] extend on the concept by basing their dimensionality reduction directly on the internal decision paths of instances in random forests.

While approaches based on labeled datasets often times provide sufficient and interpretable explanations, they evaluate a model solely on a discrete set of instances. As a result, the decision function can only be approximated from a sparse sampling of the input space, even when the local projections are chosen to show a clear separation between instances [28, 30]. However, the actual decision function may have arbitrary shapes between these instances which is not derivable from the instances alone, cf. Figure 3.1 (d). Therefore, our approach moves the emphasis from a sparse sampling to a continuous evaluation of the decision function in input space.

Decision Maps Sampling the input space on the basis of a two-dimensional embedding creates a continuous explanation of the decision function. Espadoto et al. [98] perform an extensive comparison for suitable projection techniques, which they later apply to visualize agreement between classifiers [32]. They come to the conclusion that non-linear dimension reductions are suited best for this application, which is affirmed by several other articles [105, 113, 112]. In case that the classifier explicitly defines a reduction function, e.g. a support vector machine, this mapping should be used [106]. However, such a genuine mapping usually does not exist and Rodrigues et al. [33] point out three problems with using generic non-linear embeddings instead. First, non-linear dimension reductions typically do not feature an inverse projection. Learning an approximate inverse projection can take significant time [102, 132, 133], except if it is integrated in the reduction process already [97]. Second, they tend to overfit in confusion zones leading to uninterpretable noise. Third, the distances to the visible decision boundary in the map and the real decision boundary in feature space may not match.

The approach presented here uses linear projections to create maps that inherently do not suffer from these problems. While linear projections have been considered for this application before [134], they were dismissed due to their poorer performance in cluster separation [105, 98] and possible data point overlap as compared to non-linear methods [32]. Section 3.5 demonstrates that by providing complementary interactive selection and interpretation tools, this weakness can be alleviated.

3.2 Problem Definition

The central idea of a counterfactual explanation is to describe the local structure of the decision boundary of a classifier, i.e., the borderline in input space that separates the predicted class from a different one. In order to fully understand and explore this concept, we will first introduce decision boundaries formally and motivate the approach

(Section 3.2.1). This section continues with a discussion of desired properties of map explanations (Section 3.2.2) and concludes with the construction of an embedding to explore the decision boundary around an instance (Section 3.3).

3.2.1 Decision Boundaries

Consider the point cloud in Figure 3.2 (a) which depicts a synthetic dataset of three anisotropic clusters in 2D. The colors indicate the class membership. The task of a probabilistic classifier model is to compute the probability of class affiliation for any given point in 2D space. We consider a sample point x to be predicted class A by classifier f if f predicts a probability for A higher than a target threshold t. Assuming a continuous input space, the decision boundary is then formed by the set of sample points that lie exactly on the threshold t:

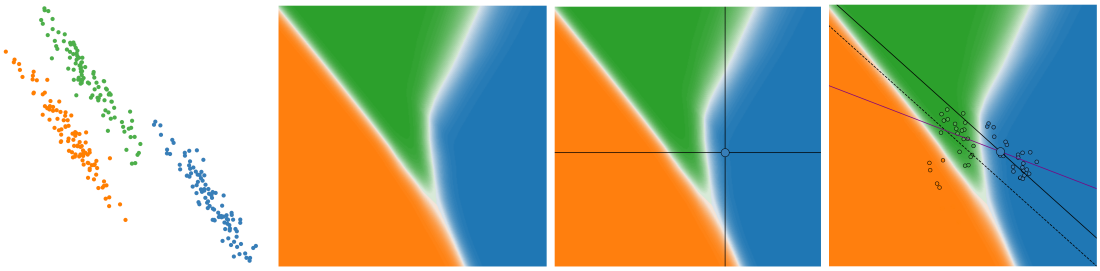
$$B(A) = \{x \mid f_A(x) = t\} \tag{3.1}$$

Without loss of generality $t = 0.5$ lends itself as a suitable threshold for binary classification [106] and is therefore used throughout this chapter. This threshold can be chosen arbitrarily to match the respective application scenario. In multi-class classification, we consider $t = 0.5$ equally applicable since in our experiments boundaries locally collapsed to only two neighboring classes. For regression analysis, the definition follows analogously with the threshold t set to separate the value range into meaningful sub-ranges [4]. In the visualization of the decision space in Figure 3.2 (b), the hue represents the highest predicted class with saturation fading to white at $f_A(x) = t = 0.5$. The white band in-between class-areas indicates the decision boundary.

As all counterfactual explanations are points that barely flip the current prediction to an opposing one, they lie on the decision boundary. In other words, counterfactuals are a sampling of a decision boundary. If one knows the decision boundary of a class A as the points where the predicted class changes, they have found all close counterfactuals. The explanation can take the form of a single counterfactual point, a set of counterfactual points, or, even more general, a representation of the decision boundary. We aim at the latter approach, as it still allows the more specific explanations.

While this concept is intuitive in our synthetic 2D example, a decision boundary can be difficult to imagine in higher dimensions. In 3D, it forms surfaces that can still be rendered using scalar field techniques such as isosurfaces [135]; for higher dimensions, we already determined that direct visualization is no longer feasible.

A prediction model forms a continuous function that outputs a value for any point in



(a) 2D dataset (b) Decision space (c) Univariate analysis (d) Our approach

Figure 3.2: Decision boundary sampling on a synthetic 2D dataset shown in (a). (b) Decision space colored by most probable class. While visualization innately works in 2D here, it becomes challenging for more than 2 dimensions. (c) Partial dependence plots sample parallel to the axes, as represented by black lines. (d) Our method samples along the purple line. The black lines indicate the construction process as explained in Section 3.3. Small circles indicate the nearest neighbors.

the input space. Therefore, the input space can be considered a high-dimensional scalar field with the scalar being the output value. A common approach that is well established for the interpretable rendering of high-dimensional scalar fields is the use of cutting planes [136, 137]. The idea is to show a representative slice of the scalar field, commonly a straight line or a plane. If the scalar field is defined by a function as in machine learning, the cutting plane can be sampled directly from this function.

This concept has already been applied for the explanation of classifiers, most prominently in the form of partial dependence plots (PDP) [27]. While PDPs average the sensitivity over a full dataset, the focus of this chapter is on explaining instance-specific behavior, and therefore we follow the notation of Krause et al. [111] to inspect the partial dependence of a single instance. A PDP samples the decision function along one axis, keeping all other attributes constant. In the 2D example of Figure 3.2 (c) this corresponds to sampling along the black lines. They start in one of the data points and vary only in one axis each.

Partial dependence can be extended to 2D analysis by changing two attributes instead of one. This constructs an axis-parallel plane that can be sampled. In our 2D synthetic dataset, this plane perfectly describes our scalar field already. Figure 3.2 (b)-(d) show this 2D sampling as the colored background. In practice however, machine learning is typically applied on multivariate input spaces >2D. The resulting nD dependency is significantly harder to visualize meaningfully. Following the PDP-approach, 2D planes could be drawn for all pairs of axes, but we determined in Section 2.1 that the amount of plots would be overwhelming for more than a single-digit number of features.

Dimensionality reduction was introduced as a possible solution to minimize the visual overload. Even though these algorithms are designed to approximate the distribution of a set of data points, contrary to our goal of exploring a continuous decision function, the 2D representation can also be used to explore the continuous decision space [132, 102]. These methods work by inversely projecting the 2D points back to input space, where they can be evaluated by the machine learning model. The result is a densely sampled embedding with only minimal remaining uncertainty in-between samples [32]. Geometrically, this back-projection is analogous to dimension reduction in our Swiss Roll example: in the linear case, it corresponds to embedding a hyperplane in the nD input space, and in the non-linear case, a hyper-surface.

3.2.2 Desiderata

After a comprehensive review of existing work on decision boundary maps and counterfactual explanations in Section 3.1, we distilled the following requirements for an interpretable explanation via decision boundary maps:

- **R1: Convey distances in decision space** The aim of a decision map is to convey the range of scenarios that keep or flip a decision. Therefore, the visual distances of test samples to the decision boundary on the map need to be comparable, i.e. there should be a monotonic relationship between visual and actual distance to the boundary. As this is hard to achieve for high-dimensional datasets [33], focusing on specific instances is sufficient for counterfactual reasoning. The visual distance measure should reflect the expected distance between inputs (Euclidean distance).
- **R2: Favor likely alterations** Reciting the goal of counterfactuals to provide an explanation via an expressive comparison, not all counterfactuals are equally helpful as explanations. A decision boundary that is reached via likely changes is more realistic and therefore more helpful than one with unlikely changes [116, 138, 118]. The distances in the embedding should reflect the likelihood of a change in reality.
- **R3: Show a close decision boundary** Reducing dimensionality is always a trade-off between many possible optimization criteria. The embedding can only cover a small subset of the high-dimensional space. Therefore, the optimization of the embedding should focus on providing explanatory value in the sense that the shown decision boundary actually is close to the explained instance [29, 139, 121]. Note that suitable counterfactual reasoning just needs an actionable close boundary, not necessarily the mathematically closest one [122].

3.3 Method

On the basis of the requirements *R1* - *R3*, we now construct a suitable embedding. The construction process is illustrated on the synthetic example in Figure 3.2 (d). The concept is illustrated in 2D, since plotting an example with an n-dimensional input space directly is infeasible. In Figure 3.2 (d), the reduction from 2D to 1D represents a real reduction from nD to a 2D map. A 1D line therefore represents the 2D hyperplane in nD.

3.3.1 Local Linear Maps

A major decision for creating an embedding with dimensionality reduction is whether to stay linear or allow non-linear distance transformations. As described in Section 3.1, in non-linear maps, the 2D distances between embedding points and decision boundaries are not matching the distances in input space [33], regardless of the projection technique [98]. Therefore, non-linear embeddings are violating *R1*. On the contrary, the axes of linear projection techniques are based on linear feature combinations, therefore keep interpretable distances, and can fulfill *R1*.

From the plethora of linear transformations, we choose PCA [34] to build our embedding visualization, an approach also utilized by OptMap [140] in a related scenario. In contrast to OptMap, we focus on explaining classifier outputs instead of optimization paths in the domain space of real-valued functions. Further, in OptMap the PCA is computed from samples on a regular grid, while *R2* leads us to base it on the distribution of real data instances to favor likely alterations. This training data is not restricted to the training or test dataset of the model and can be any realistic distribution of samples as long as it is not strongly biased.

Why not use a linear embedding that maximizes distance between classes? PCA is chosen over more discriminant methods like LDA [42] for two reasons: (1) PCA-axes are optimized to capture the variance in the data. Assuming a higher variance in a feature means that it is more likely to change, the embedding implicitly accounts for feature variability (*R2*). Since this assumption may only hold for a theoretical dataset as in Figure 3.2, we provide additional tools to control feature mutability in Section 3.4.5. (2) Since we standardize all features before PCA, the analysis is an eigenanalysis of the correlation matrix. Thus, the covariance of two axes signals the correlation between the features. As a result, we get the convenient property that points in our map adhere to the linear feature correlation in the dataset (*R2*). While we cannot expect a dataset to

only contain linear correlation, the manifold hypothesis [51] suggests that this assumption holds on local neighborhoods. Hence, we restrict PCA training samples to a set of nearest neighbors in nD.

Lastly, *R3* aims to show a *close* decision boundary. However, the PCA-hyperplane is not guaranteed to find a decision boundary at all, especially not a close one, as it is trained on the variance of the data and not on the decision function. To satisfy *R3* we align the hyperplane orientation towards a close decision boundary by restricting the training samples to a relevant local subset. Therefore, we construct a balanced set of nearest neighbors for the explained point, where half the samples share the predicted class and the other half is predicted to be in another class. The neighbors are determined on Euclidean distance of standardized feature values. To speed up the neighborhood search, a ball tree data structure is set up for each class to achieve logarithmic search time scaling. A default of 100 neighbors, 10-15% of the data in our case, worked best in the experiments, but is adjustable in the interface header.

The resulting map conveys interpretable distances (*R1*), favors likely alterations (*R2*), and is oriented towards a close decision boundary (*R3*). However, our goal was not only to show the decision boundary, but using it to explore counterfactual explanations. A counterfactual explanation is only sensible in reference to a decision on a baseline instance $x_i = (x_{i1}, x_{i2}, ..., x_{id})$. Therefore, we guarantee that the hyperplane is centered around the explained instance by shifting it to this instance. Traditionally, the PCA-hyperplane is anchored in the mean of the training data, yet our translation puts the explained instance to the center of the map.

3.3.2 Sampling

Figure 3.3 repeats the incremental sketch of our proposed embedding: The dashed line represents how a regular PCA would cut through high-dimensional space, the solid black line shifts towards the explained instance and the purple line is further oriented towards the decision boundary. The shown points indicate the balanced nearest neighbors used for creating our custom principal components pc_1 and pc_2. The mapping from a point at 2D coordinates (x, y) in embedding space to feature space is computed with:

$$Inv(x, y) = [x, y][\vec{pc}_1, \vec{pc}_2] + x_i \tag{3.2}$$

As this mapping is computationally simple and can be vectorized, the embedding is sampled once per pixel (x, y) (Figure 3.3: along the purple line) and the corresponding points in input space $Inv(x, y)$ are classified in the model. The predictions form a

multivariate partial dependence plot colored by the most probable class. The neighbors used for training are projected into the plane as colored circles to identify practically occurring feature combinations, though these could be omitted to reduce complexity for casual users.

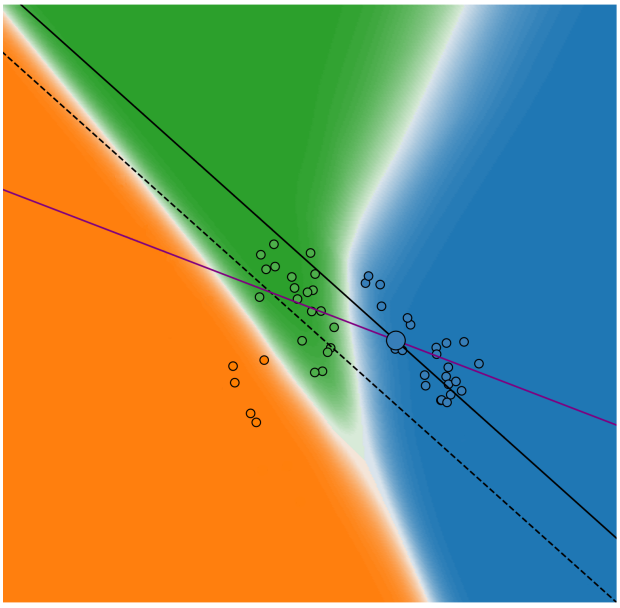


Figure 3.3: A theoretical example of sampling a 2D decision space to 1D. A regular PCA embedding would sample along the dashed line. Shifting this embedding to the explained data point is represented with the black line. The purple line illustrates the local embedding based on nearest neighbors. Small circles indicate these nearest neighbors.

3.4 System Design

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

3.4.1 Design Overview

The review of related work in Section 3.1 revealed design requirements for a holistic analysis of decision boundaries, which can be ordered from overview to detail.

- **DR1: Provide an overview** Show the overall class separability and data distribution [98, 105] as well as the focused instance's placement therein [112] to give an overview.
- **DR2: Present simple explanations** Allow univariate sensitivity analysis to support sparse and simple explanations [29, 141, 111]
- **DR3: Support multivariate explanations** Provide interpretable, direct analysis of the decision function under multidimensional changes (Section 3.2.2) to support multivariate explanations.
- **DR4: Account for feature significance** Retain flexibility to account for uneven feature model importance [103] or mutability [120]

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

- **DR5: Independence of the underlying model** With the ever-changing search for superior model architecture, fitting a visualization technique on a specific model type severely limits its usability. As the presented methods is a black box approach, we only require a *predict* function and sample data, thus are model-agnostic.
- **DR6: Applicability to common data types** The most common types in machine learning are texts, images and tables. For texts and images, the exploration of decision boundaries is currently still restricted to exemplary counterfactual generation [142, 127, 143], so the focus in this dissertation is on tabular data with a number of dimensions displayable as a list, i.e. less than 30 [125, 128, 123, 111].
- **DR7: Applicability to a variety of model outputs** For the design overview, we focus on binary- and multi-classification problems. The translation to regression analysis is described in Section 3.4.6.

Our interface, shown in Figure 3.4, is split into four major components: The *topology view* (A), the *partial dependence view* (B), the *embedding view* (C), and the *feature selection* (D). Each of the components fulfills one of the requirements *DR1-DR4*. The topology view grants insight in the separability of the dataset (*DR1*). The partial dependence view shows the univariate analysis of the local decision behavior (*DR2*). The embedding view extends the exploration to multivariate space (*DR3*). The feature selection provides a guided way of reducing the search space (*DR4*). Completing the interaction, data points can be selected and compared in a data table (E). All components are linked and update to the current selection. A prototype of the presented framework is implemented using Python, Bokeh [145] and Panel [146] and is publicly available on GitHub[1].

[1] https://github.com/Jan-To/COFFI

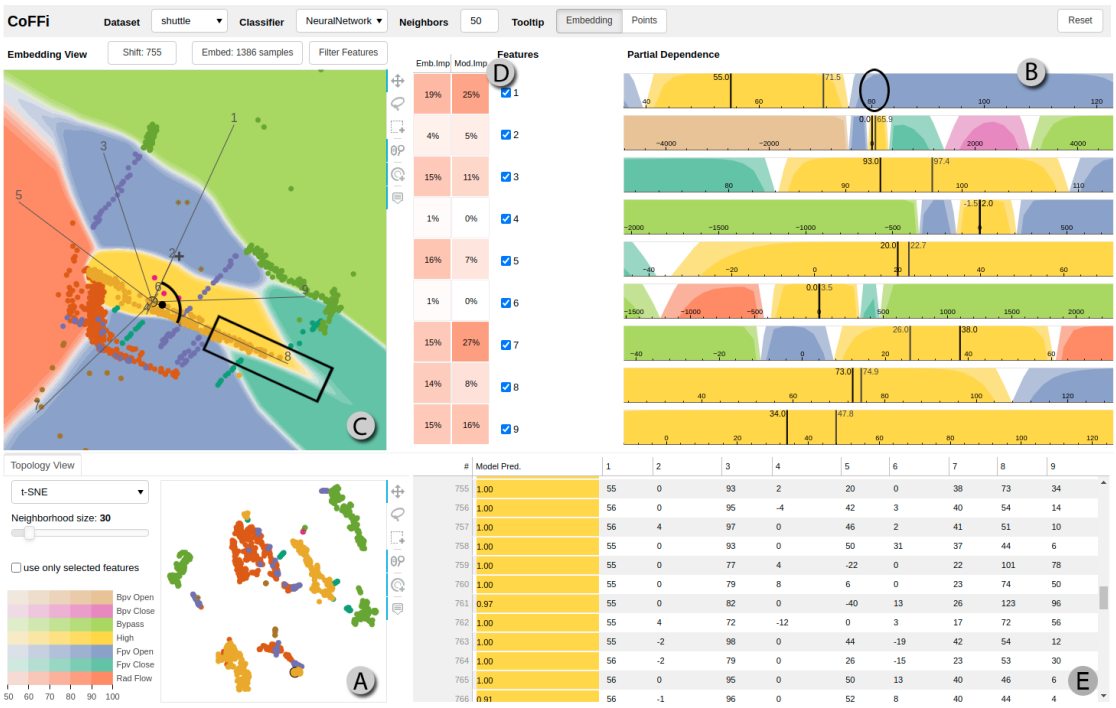


Figure 3.4: CoFFi interface on a space shuttle dataset [144], explaining decision boundaries of sample 755. (A) The topology provides an overview of class cluster distribution. (B) The partial dependence view shows single parameter change analysis. (C) The embedding view shows predictions under multivariate changes. (D) Feature importance can be aligned between model and embedding for quality analysis. (E) Specific points can be searched and selected in a data table. Black annotations in (B) & (C) highlight conclusions of Section 3.4.3 & Section 3.4.4. A higher-resolution version of this figure is available in the Appendix.

We explain the interface on the example of a dataset from NASA [144] shown in Figure 3.4, where nine radiator sensor measurements of a space shuttle are given and a fully-connected neural network with three 100-neuron-layers is used to predict the corresponding radiator position. The example model achieves 99% accuracy, and we want to understand how the input space is distributed into the seven classes. The dataset contains 58 000 instances, hence we reduce overplotting by limiting the visible scatter glyphs to maximal 400 instances per class. We also tested with using full data for embedding computation and achieved similar results while also remaining interactive. A more detailed analysis on scalability is provided in Section 3.6.1.

3.4.2 Topology View

The exploration of decision boundaries in high-dimensional space is only sensible if we know what we are looking for. For a clustered, clearly separable data manifold as in Figure 3.5 (a), we can expect the model to form similarly simple decision boundaries. For a complex, hardly separable data manifold later discussed in Section 3.5 and shown in Figure 3.5 (b), the decision boundary could be equally complex. Hence, the first component to look at after loading in a dataset and a trained prediction model is the topology view.

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

Two state-of-the-art non-linear dimensionality reduction techniques – t-SNE and UMAP – are available to plot the data points based on their feature values. The positioning of the data points approximates the intrinsic manifold of the dataset, while the color distribution indicates the class separability. Classes that have distinct feature combinations show in a clear separation into uni-colored clusters as in Figure 3.5 (a) ■ orange, ■ yellow, ■ green, while the other classes seem harder to differentiate. In our experiments, t-SNE proved better for finding clusters, while UMAP excelled at capturing the intrinsic manifold. Nevertheless, the outcome of both algorithms strongly depends on

(a) Shuttle data [144], t-SNE (b) Diabetes data [147], UMAP

Figure 3.5: Topology view with (a) separated and (b) overlapping classes. Misclassified samples are marked with a cross in the color of the ground truth class.

their hyperparameters, so we included sliders to experiment until a satisfying setting is found.

The downside of these non-linear embeddings is that most of the information about global distances between points is lost. Conclusions can only be drawn about immediate local similarity. Due to the focus on immediate locality and the resulting distortion in the non-linear embedding, points that are distant in the plot still appeared as nearest neighbors. In preliminary versions, the scatterplot was augmented with inverse mappings of non-linear dimensionality reduction as in [112]. However, the background embedding did not provide additional information about class separability compared to the glyph-colored scatterplot alone. For that reason, we chose to not add background coloring here.

The aim of this workflow is to explain the decision boundary for a specific data point. When deciding for a point of interest, the topology view provides hints about atypical instances. A single different-colored dot in a uni-colored cluster or a misclassified instance indicated by a cross are good starting points. Custom points can be added in the data table. In the shuttle example an instance of the central class ■ High is chosen, demonstrating the possible variety of adjacent classes. A click selects the instance and updates the other components to this point.

3.4.3 Partial Dependence View

The most easily understood way to describe decision boundaries is by illustrating univariate behavior. In the context of model predictions, this approach is typically called partial dependence analysis [27]. The class prediction is observed altering one feature at a time while the other dimensions are fixed. By design, partial dependence perfectly captures the univariate behavior around the examined instance. As partial dependence is a well-established technique that still marks the state-of-the-art for inspection of feature-model relationships [67, 111, 128], we include them with a new look depicted in Figure 3.6. Partial dependence is typically plotted as line charts [128, 67, 148, 27] or colormaps [111]. We choose to use a hybrid of quick-to-read colormaps and precise-to-read line/area charts called horizon chart [149], which combines the advantages of both previous visualization methods.

Horizon charts are vertically condensed and color coded area charts. The area is segmented into ideally two horizontal bars [150] which are color-coded and shifted over each other. The color component facilitates highlighting regions of interesting value

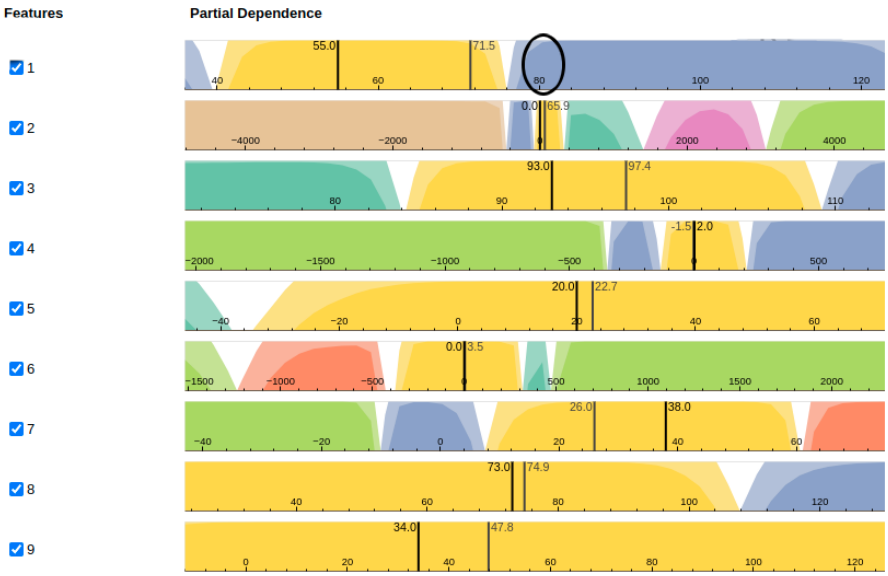


Figure 3.6: Partial dependence view of Figure 3.4. Black lines indicate the selected instance, while gray lines indicate the counterfactual probed in the embedding view.

range, while the slope conveys an accustomed way of reading gradients as well as the possibility to read exact values. The lowered space constraints of horizon charts over line charts allows us to show significantly more dimensions at once than previous methods [128, 67, 148].

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

The message conveyed by horizon graphs depends on the axis ranges. The x-axis should cover all possibly occurring feature values, hence, we set it to the range of the training set. The vertical baseline is the decision boundary threshold $t = 0.5$, since our goal is to put emphasis on the pivot point between predicted classes. The vertical axis is set to 25% prediction change, so that over 2 positive bands the horizon covers the prediction range between 50% and 100% per prediction per class. The classes are color-coded with richer colors representing more confident classifications. We can therefore read the model prediction under all possible individual feature changes from the horizon graphs.

The feature values of the currently selected data point are indicated with black lines, while the currently explored counterfactual, which we will learn about in the next section, is shown with gray lines. We select point 755 in Figure 3.4 and enlarge the partial

dependence view in Figure 3.6. None of the sensors measure an extreme value. Therefore, the neural network predicts 100% class High, which can be read from the full height of the saturated yellow area at the black lines. The chart provides hypothesis testing such as if sensor 1 had been 80 (black ellipse) and all other sensors were the same, the model would have predicted 99% Fpv Open.

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

3.4.4 Embedding View

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

The Embedding View depicted in Figure 3.7 shows the prediction probabilities on a linear hyperplane based in the inspected point. The necessary cutting plane is generated as introduced in Section 3.3. The shown parameter space can be sampled at every position, as the map induces a continuous bijective mapping between the plane and the parameter space. A dense regular grid is sampled on the cutting plane to create a smooth visual impression. For each sample the class probabilities are predicted in the machine learning model. From these probabilities, we create a map of the model predictions on the hyperplane, with each pixel colored according to the most probable class. The saturation is increased with certainty of the prediction. Hence, decision boundaries will show as white areas or flips in hue.

A gray biplot [151] of the high-dimensional axes is added to indicate how feature values change when moving in the sampled plane. The biplot is centered in the focused data point and the axes point in positive feature direction. Since the hyperplane is created on centered and normalized training data, the covariance of two axes in the biplot signals the correlation between them. Axes in similar directions can be assumed to be positively correlated, axes in opposite directions to be negatively correlated, and the other axes to

Figure 3.7: Embedding view of Figure 3.4 with black annotations illustrating the conclusions drawn in Section 3.4.4.

be not correlated. Further, the axes scaling attends to the variance of each feature in the training data, which is encoded in the relative length of the axes.

The colored map lets us draw conclusions about multivariate decision boundaries. In our linear embedding feature values increase linearly in the direction of their respective axis. Moreover, the respective feature value does not change when moving orthogonal to an axis. We enforce an 1:1 axis-ratio in the embedding plot so that when a variable strongly influences a decision boundary, it is visible as its axis being (close to) orthogonal to that boundary. Hence, a decision boundary that is linear and orthogonal to an axis, relies on this feature to cross a certain threshold. In Figure 3.7, the lower part of the decision boundary between High and Fpv Close (black box) is orthogonal to the feature axes 1 and 2 (angle with dot), so one of them or both are the deciding feature here.

From the extent of the linear boundary, its generalizability can be estimated with regard to the other features, as these change when moving along the boundary. In Figure 3.7, the linear boundary ends in a sharp curve when one of the similar-oriented features 5, 8 or 9 reach a certain value. Curved decision boundaries indicate that a combination of feature values are necessary to flip a decision. The partial dependence view in Figure 3.6 hints towards a further exploration of sensor 5 as it shows a teal range on the lower end. The necessary workflow is described in Section 3.4.5.

To annotate the embedding, training samples are projected into that plane and shown as circles in the plot. The same coloring schema as in the topology view is used which uses slightly more saturated class colors to make circles distinguishable from the background. The sample points work as an indicator for the spatial distribution of real data as well as for the descriptiveness of the embedding. Regions where samples share their color with the embedding are regions where the cutting plane is illustrating the behavior of the model on real data well. Regions where the colors don't match the background imply that the cutting plane does not generalize to these points. In the example embedding of Figure 3.4 C the ■ Fpv Open, ■ High and ■ Bypass match with the linear embedding, while the projection is inaccurate for the other classes.

A point of interest can be focused by clicking in any view, increasing the circle size. The embedding and Partial Dependence View can then be updated to the selected point with the *Shift*-Button. This updates the embedding to the new point and re-computes the inverse mapping. The embedding is then re-colored according to the new prediction probabilities. Note that the view stays rather consistent, as only the embedding colors change, while the axes orientation and the location of the sample points stay the same. This is a result of our embedding construction. Since the samples are orthogonally projected onto the linear embedding, shifts with the same training data necessarily are parallel slices of the parameter space. In other words, shifts occur only orthogonal to the initial hyperplane, cf. Figure 3.3 black vs. dashed line. This is a desirable property, as it allows for a consistent comparison of different points in the embedding space.

On startup and initial selection of a data point, the embedding is based on a (shifted) PCA of all available instances for overview and consistency, cf. Figure 3.3 black lines, Figure 3.4 C, or Figure 3.10 A. However, based on the approach in Section 3.3, the training data of the embedding should be reduced to the nearest neighbors to better adhere to the local manifold around an instance. This is achieved with the *Embed*-Button, which then reflects the nearest neighborhood, as in Figure 3.3 purple line, or any custom selection drawn in the embedding or topology plot. All non-selected samples are rendered transparent, as their projection to the embedding is less meaningful. In our experiments, the closest 100 nearest neighbors provided a good balance between locality and generalizability, thus, we use this number in our case study in Section 3.5.

Lastly, the embedding can be probed at any position by hovering, generating a contrastive explanation for the prediction. Clicking in the embedding creates a gray cross that marks the probed position. Simultaneously, the partial dependence view shows the feature values of the inversely projected point as gray lines. This serves two purposes.

First, precise readings of feature values at the decision boundaries and beyond can be taken. Second, the comparison with the black markers of the focused point reveals the necessary changes that can serve as a counterfactual explanation. In Figure 3.7, we selected a multivariate counterfactual beyond the ■ Fpv Open boundary that was not evident from individual feature changes in the partial dependence view. By freely choosing points along the decision boundary, the user can explore a plethora of possible counterfactual explanations visually. Thus, the explanation can be steered to personal preference without knowledge about coding, machine learning models or dimensionality reduction beyond PCA. The steering of the exploration is extended in the next section.

3.4.5 Feature Selection

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

In the feature selection component, features can be disregarded for the analysis by fixing them to the current value. The embedding view is then read as: *What are the predictions of likely changes to the focused instance under the assumption that unchecked features remain static?* In case the user has no preferences on which features are immutable or irrelevant, guidance on how to filter features is required.

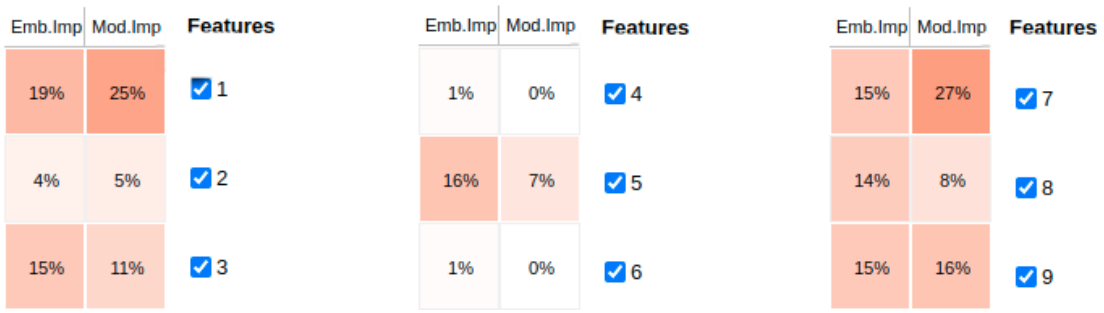


Figure 3.8: Feature selection component comparing embedding importance, model importance and mutability. Image is sliced from Figure 3.4.

The feature component is also a way to assess and improve the quality of the embedding. We assume that an expressive embedding is one that captures the model sensitivity. Therefore, we compare the feature influence in the *model* to the one in the *embedding*. A high accordance between the two indicates that the embedding is a good representation of the model. An annotated heatmap in Figure 3.4 D, which is shown in detail in Figure 3.8, juxtaposes the respective values. In this case, the feature importance in the model mostly lines up with the variance in the embedding. Thus, the embedding can be assumed to show a representative explanation of decision boundaries.

Embedding influence is computed over the length of the feature vectors spanned by PCA. Model sensitivity is computed via the permutation importance measure [77, 152]. We made this choice based on implementation availability and because true model-agnostic feature importance is still a topic of active research. We restrict the model importance measure to the training data of the current embedding, thereby keeping the measures comparable at all times, implicitly adjusting to local differences. For easier comparison, both measures are normalized. The embedding quality can be assessed by comparing the influence of features in embedding and model, and improved by adjusting the embedding to better mirror the model.

3.4.6 Extension to Regression

The fundamental basis of counterfactual reasoning is the comparison to an opposing outcome. In regression analysis the target variable is continuous, hence an opposing outcome is no longer clearly defined. Moreover, it is highly subjective what an opposing outcome would be. We therefore suggest transforming the continuous outcome variable into a categorical one by segmenting the continuous space into meaningful ranges. This allows us to define decision boundaries as the transition between target segments.

However, the transformation requires some precautions. As the comparison between segments only holds explanative value if the segments have contrastive meaning. Sometimes choosing segments is obvious due to underlying data, e.g., water boils at 100°C and freezes at 0°C. In other cases, the segments are not so clear and need to be defined by the user. Segments should be chosen such that they are meaningful in the context of the data and the model. To aid the user in this process, we provide a histogram of the target variable distribution in the bottom left of the embedding view. Figure 3.9 shows the CoFFi interface for a regression analysis on the chemical dataset of Chapter 5, where

the segments are chosen based on the distribution's modality into low, medium and high values. A detailed analysis of the dataset in CoFFi can be found in [4], as such in-depth examination lies beyond the methodological focus of this chapter, where we instead concentrate on the more straightforward use case of classification models.

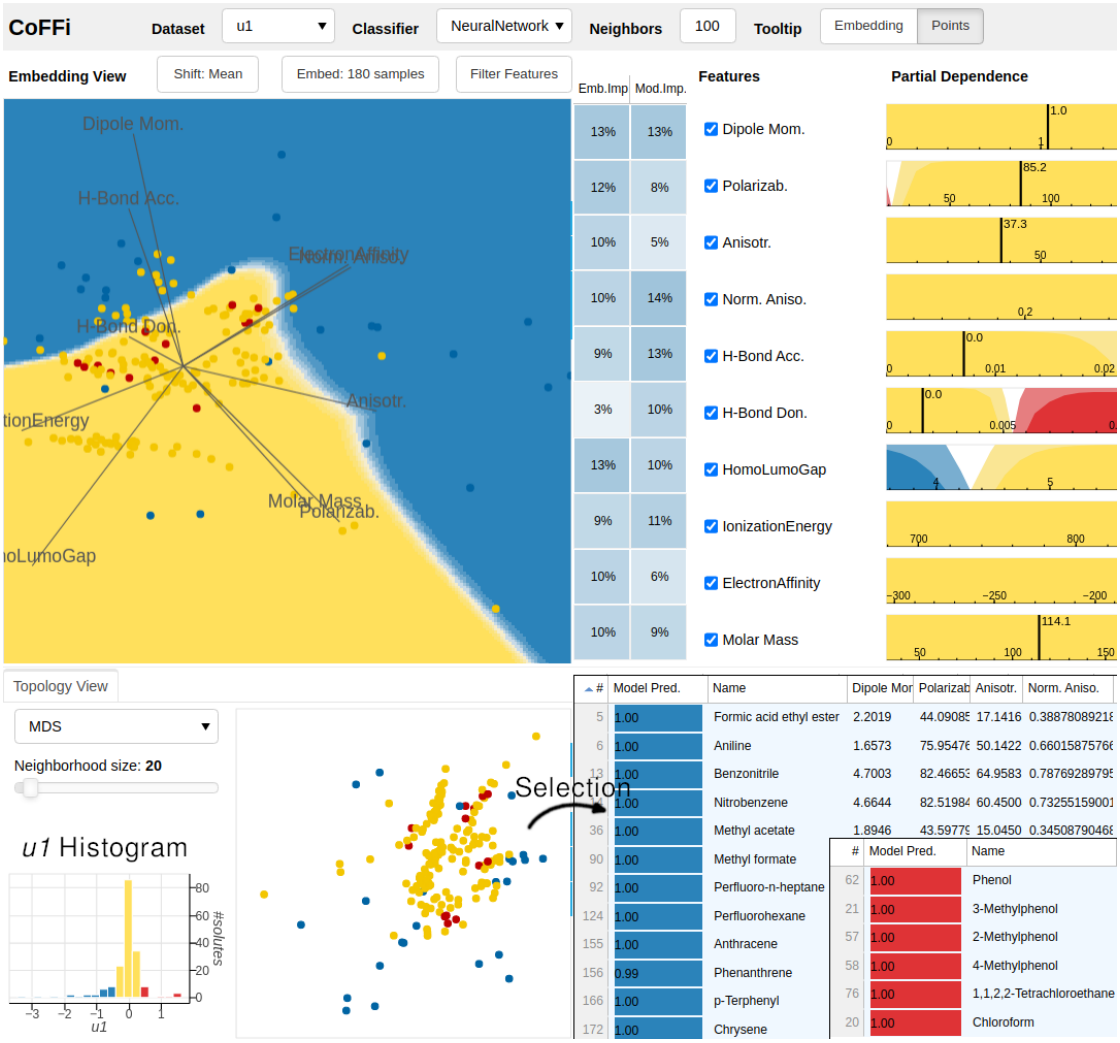


Figure 3.9: A snippet of the CoFFi interface for regression analysis on the chemical dataset of Chapter 5. The target variable is segmented based on modality into low, medium and high shown in the histogram on the bottom left. Selection in topology view reveals name similarity within groups. The embedding view shows the decision boundaries between the groups. A detailed analysis of the dataset in CoFFi can be found in [4].

3.5 Case Study

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

The diabetes diagnosis of members of the Indian Pima tribe is to be automated. The dataset is preselected to contain only females above 21 years [147]. The dataset contains 768 samples with 8 numerical input features. The output variable signals whether the patient was diagnosed with ■ diabetes mellitus or ■ not. We train a random forest classifier with 100 internal trees and achieve 80% cross-validation accuracy.

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

After seeing no clear class separation in the topology view in Figure 3.10 (E), we select f667 by clicking it and shifting to (A+D). The partial dependence view (D) updates to show the model's response under feature changes in one dimension. The ■ green areas left of the black instance-lines in (D) indicate that a lower pregnancy, glucose, BMI or age value lowers the prediction for diabetes. Apart from this expected behavior, we mark multiple unexpected patterns in Figure 3.10 (D) that raise suspicion about model robustness. First, the chances to be healthy would have been significantly improved had the female been older than 55 years (bottom-most horizon). Second, a higher glucose value from 111 mg/dl up to 140 mg/dl would improve her prediction (2nd horizon) even though the healthy range is anything under 140 mg/dl according to the American Diabetes Association. Lastly, the model would predict her as a diabetes patient would her BMI be just one higher, but an BMI increase to between 32 and 42 would have improved her prediction (6th horizon). It is likely that this is a model artifact from overfitting to a biased training set. We will find out in the following exploration.

While one-dimensional hypotheses are insightful, the medical measures considered here are unlikely to be changed independently. Figure 3.10 (A) shows the global linear embedding shifted to be centered at our sample. We can therefore explore the prediction results under multidimensional alterations to f667's values respecting the global correlation and variance. The axes show three correlated groups of features: Pregnancies and

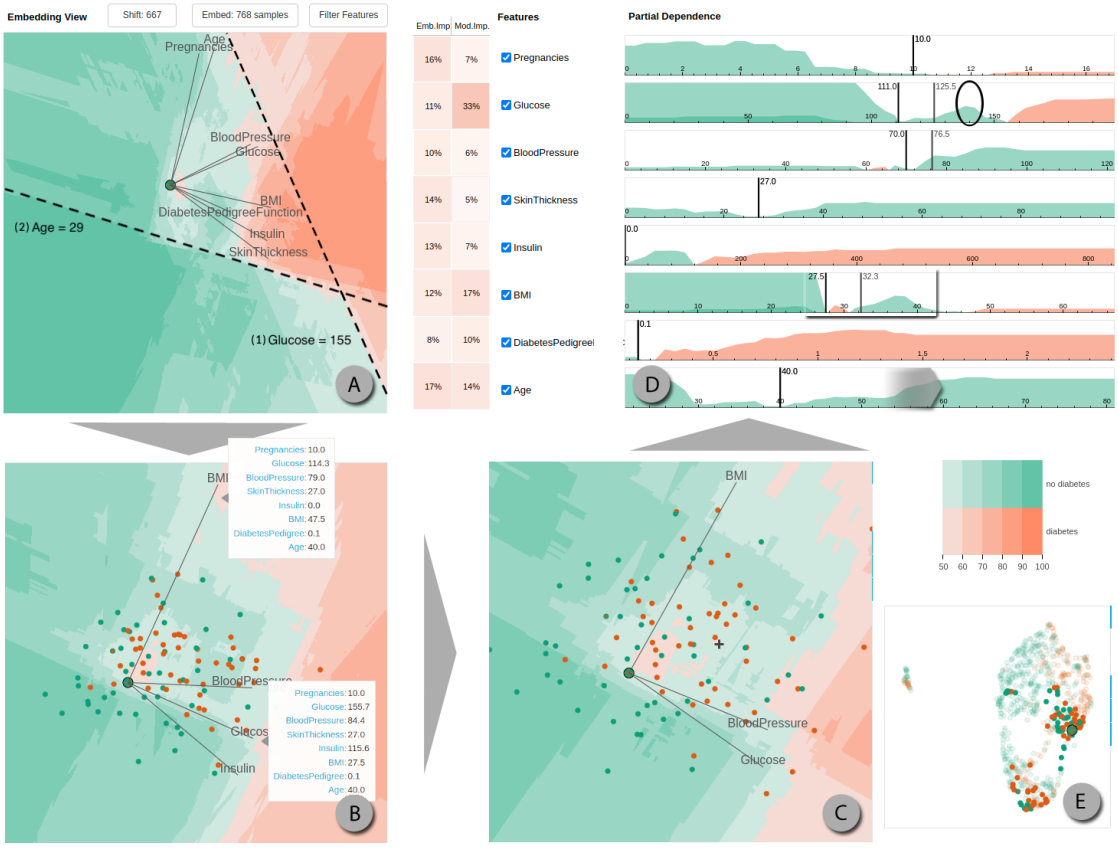


Figure 3.10: Workflow of local analysis of f667 in the diabetes dataset. (A) Inspecting global map. (B) Restricting to local neighborhood and fixing the values of immutable and less-influential features. (C) Further reduction of axis for closer inspection of boundary artifacts. An unusual counterfactual is found. (D) Comparing the multivariate counterfactual with the expected sensitivity under individual feature changes. (E) Topology overview shows no class separation on non-linear manifold.

age, glucose and blood pressure, and the rest. While we cannot explain the last group, the first two are expected since females with more children tend to be older, and glucose increases the blood pressure. It is therefore sensible for our embedding to assume that these features change proportionally.

Decision boundaries orthogonal to the respective axes indicate that (1) glucose levels above 155 mg/dl lead to higher prediction; even if the blood pressure rises accordingly and regardless of the other features; (2) being younger than 29 with fewer pregnancies helps to reduce the diabetes chance, but is overruled by Glucose values. We annotated both observations with dashed lines in (A). The exact values of embedding points can be probed with tooltips in the embedding and gray lines in the partial dependence view, visible in Figure 3.10 (B), (C) & (D).

Still in Figure 3.10 (A), we notice a small orange patch immediately to the right of f667. We conclude that it corresponds to the same phenomenon of BMI sensitivity in the horizons. All points on the hyperplane within this patch are predicted to have diabetes. The important difference to a well-formed decision boundary here is that this model behavior is not generalizable. An algorithmic counterfactual explanation would have been: *If your BMI was slightly higher, you would have been correctly diagnosed with diabetes.* While this is correct, it leads to the wrong assumption. To overturn her AI-doctor, f667 gains weight and increases her BMI by three. She still is rejected even though she followed the explanation. With our visual inspection, she would have known that this behavior does not generalize to higher values. In our multidimensional linear map, the small size of the patch immediately shows that this behavior is not applicable for similar feature combinations. It can therefore be assumed that the patch is a model artifact from training data. We investigate this further in the next paragraph.

Until now the linear embedding is chosen to approximate the global covariance. However, a dataset can have non-linear associations between variables, so we localize the embedding. We restrict training data of the projection to the nearest 50 neighbors per class. Our patient further cannot readily change skin thickness, pedigree function, age or number of pregnancies. Therefore, we mark these features as immutable as described in Section 3.4.5. The resulting plot after both changes is shown in Figure 3.10 (B). The orange patch is still small, hence our initial suspicion is confirmed. The class shift by increasing BMI only applies if the other feature values are within a small, specific range. The orange main area, however, is much larger, indicating that the model is more robust there. We formalize a generalizable counterfactual explanation by probing the embedding in the orange main area via hover or click: f667 needs to change her glucose level above 155 mg/dl or her BMI above 47.

Lastly, we demonstrate the benefit of multivariate explanations over classical univariate ones. A common misconception is that multivariate probing can be achieved implicitly by adding individual partial dependencies. This is generally not the case, and we can demonstrate on this example. First, we simplify and fix the insulin value to get to Figure 3.10 (C). Clicking in one of the small ■ orange patches in the embedding view marks the counterfactual with a gray cross. The corresponding gray lines in the partial dependence view (D) show that each individual change ■ reduces the likelihood for diabetes. Consequently, their combination is expected to reduce it even further. However, we started in an orange patch, so we know that changing all three at once results in flipping the prediction towards ■ diabetes. The need to assume such interactions vanishes with our linear embedding visualization as the likely ones are presented already.

3.6 Discussion

The case study demonstrated how the proposed design can be used to explore decision boundaries and draw conclusions. The analysis of the diabetes dataset shows that CoFFi can help identify model artifacts and understand the decision-making process of machine learning models. The embedding view allows for a more detailed exploration of the decision boundaries, while the partial dependence view provides insight into the univariate behavior of the model. We can conclude that the proposed design fulfills the requirements set out in Section 3.2.2 and provides a useful tool for exploring decision boundaries in high-dimensional datasets. To provide a more comprehensive understanding of the proposed design compared to the state-of-the-art, we will discuss the scalability and quality of the approach in the following sections.

3.6.1 Scalability Analysis

CoFFi is designed to be a tool for exploring decision boundaries interactively. As such, it needs to be able to handle sufficiently large datasets in a reasonable time. The scaling of our approach is determined by four factors: The number of features, the number of data instances, the sampling resolution and the speed of model evaluation.

In our experiments, CoFFi handled up to 30 dimensions, which is the maximum number of features that can be reasonably displayed in a list – a common limitation in visualization tools [125, 128, 123, 111]. The embedding view can be computed for any number of dimensions, but the axes become cluttered and less interpretable as the number of dimensions increases. For us, the embedding view remained interpretable up to 10 dimensions, beyond which the visual complexity increases significantly.

The number of samples is less of a concern computationally as visually. The presented embedding technique is based on PCA. This means that the computational complexity of the embedding is $O(n^3)$, where n is the number of samples in the training data. However, in absolute numbers this is still significantly quicker than previous non-linear maps such as t-SNE or UMAP [153]. Non-visual counterfactuals are typically computed algorithmically via optimization [154], which is computationally expensive while only giving a spot check. Nonetheless, overplotting is a serious concern for all scatterplots, hence we recommend using only a subset big enough to provide insight into data distribution, e.g., less than 3000 samples. In our experiments, the projection and inverse mapping required under 10ms, which is negligible compared to previous non-linear maps.

The dominant factors for interactivity are the sampling resolution and the model evaluation time. The sampling resolution is determined by the number of pixels in the embedding view. In our experiments, we found that a resolution of 300x300 pixels was sufficient to provide a smooth visual impression while keeping the computation time reasonable. The model evaluation time is determined by the complexity of the underlying machine learning model. In our experiments with a fully-connected neural networks and a random forest, the embedding generation took under 300ms for 300x300 pixels, scaling linearly with both pixel count and ML model evaluation time. Hence, it is possible to zoom and pan interactively in the embedding, exploring regions of interest in more detail.

3.6.2 Quality Analysis

To assess the quality of our embedding technique, we compare it to state-of-the-art decision boundary maps, iLAMP [132] and iNN [102]. We conduct our benchmarks on the breast [144], diabetes, robot [155] and shuttle dataset. To simulate interactive use of our tool CoFFi, we also implemented a naive feature filtration where only features with more than average feature importance are kept. Note that this case is just to show the potential of the tool, since a direct comparison to full-feature maps is unfair. We base our analysis on the general map desiderata *R1-R3* in Section 3.2.2.

Table 3.1: Comparison of decision boundary maps iLAMP [132], iNN [102], CoFFi and filtered CoFFi. Shown are the average L1-distances to closest shown counterfactual in normalized feature space $\overline{d_{shown}}$ and Pearson correlation between shown and optimized counterfactual from alibi ρ. The best results of the all-feature approaches are marked in bold.

		iLAMP	iNN	CoFFi	fil. CoFFi
breast [144]	$\overline{d_{shown}}$	12.41	12.30	**11.92**	5.58
	ρ	0.14	0.14	**0.47**	0.29
diabetes [147]	$\overline{d_{shown}}$	7.56	7.18	**5.07**	3.84
	ρ	-0.07	-0.05	**0.37**	0.48
robot24 [155]	$\overline{d_{shown}}$	22.60	22.33	**14.97**	5.61
	ρ	0.03	0.04	**0.17**	0.21
shuttle [144]	$\overline{d_{shown}}$	9.96	9.90	**7.57**	3.46
	ρ	0.16	**0.19**	0.10	0.16

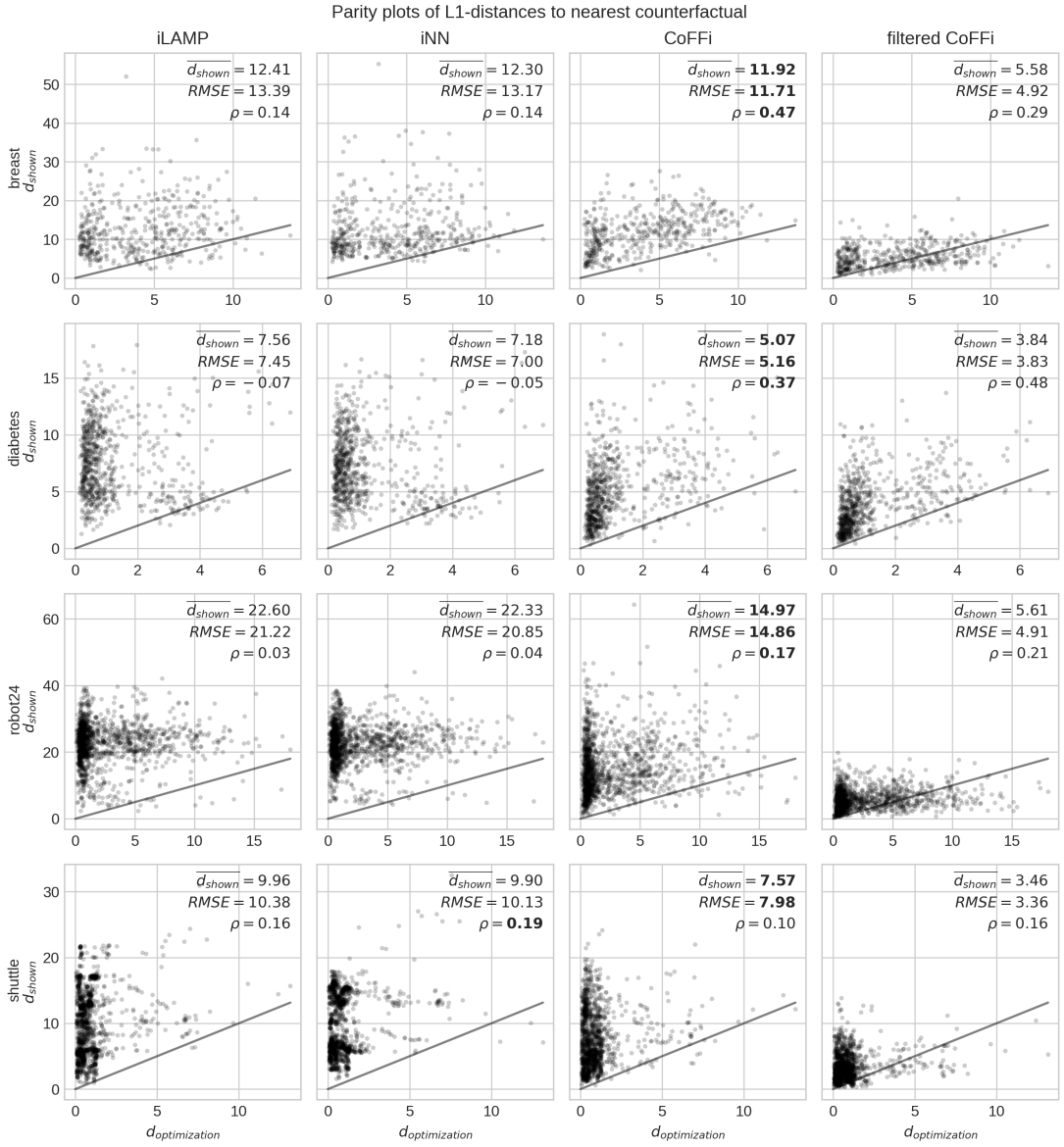


Figure 3.11: Parity plots for L1-distances in feature space to the nearest counterfactuals in three mapping techniques d_{shown} drawn against $d_{optimization}$ the L1-distance found by optimization [154]. Each dot is an instance in our dataset and the line marks the bisector (the theoretical target). In each plot, we give the mean of visual counterfactual distance $\overline{d_{shown}}$, the $RMSE$ of d_{shown} compared to $d_{optimized}$, and the Pearson correlation coefficient ρ of the two measures.

Our map is explicitly defined to convey linear distances based on likelihood of change as discussed in Section 3.3 and therefore fulfills *R1*. On the other hand, Rodrigues et al. [33] confirms that the distortion problems of non-linear embeddings translate to non-linear decision maps.

As a measure to favor likely alterations (*R2*), we assume a likely alteration is a close one. We compute the closest shown counterfactual per sample, as if a user clicked on the closest differently colored pixel in each of the approaches. In our approach the average L1-distance in feature space, which is the common measure for counterfactual distances [29], is lower than with iLAMP and iNN in all our experiments (Table 3.1 $\overline{d_{shown}}$).

To confirm *R3*, we test how well the shown decision boundary is actually matching with the closest decision boundary. As finding the closest decision boundary is NP-hard [119], we rely on the closest counterfactual found through algorithmic optimization as a baseline, which we compute with *alibi* [154]. In a perfect embedding the shown counterfactuals should coincide with the optimized ones. Therefore, we compute the Pearson correlation coefficient between the counterfactual distances found through optimization and the embeddings (Table 3.1 ρ).

In the experiments our approach found closer counterfactuals on average and in most scenarios it also showed higher correlation with the "optimal" counterfactuals than previous non-linear approaches. At the same time, our approach relies on a simpler concept with lower computation time. Detailed plots of the benchmark results can be found in Figure 3.11. While not a fair comparison to the other methods, the naively-filtered CoFFi creates even closer counterfactuals than the full-feature CoFFi. To our surprise, in some cases it even improved upon the ones found through optimization. This behavior persisted when varying the search parameters of *alibi*.

3.6.3 Limitations and Future Work

Our approach remains with limitations regarding scalability, interactivity and accessibility that are planned directions for future work. As in previous work, visualizing too many sample points in a scatterplots leads to overplotting. While, the full dataset can be used for computation and instance-selection, only a subset of points should be plotted. Though we successfully experimented with up to 30 input dimensions in our list, we suggest moving the feature selection process to a plot representation [103] or a recommender system. Due to the limited number of perceived colors [156], only about eight output-classes can be inspected at once and more classes would need to be grouped. While the proposed projection is significantly faster to compute than previous maps, creating a densely sampled map still depends on the excessive probing of the decision function ($\mathcal{O}(resolution)$). Hence, the interactivity of any map depends on a rapid model

evaluation. Finally, the orientation of our hyperplane is fixed via dataset correlation. This is a reasonable choice for most datasets, but it may not be the best choice for all datasets. In particular, if the dataset has a non-linear manifold, the linear embedding may not capture the true structure of the data. We rely on neighborhood-balancing to improve the local manifold, but this may not always be sufficient. Explanations could increase the adherence to the model by class discrimination [157, 158, 105] with the cost of losing current variability information. Customization can be advanced by introducing movable axes with drag-and-drop [159]. As we are aware that especially the axes and hyperplane interpretation acts as a high entry barrier, further evaluation of the accessibility is required. A user study with novices and experts could indicate the usefulness of an integration into an ML engineering pipeline in practice.

3.7 Conclusion

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

Parts of this chapter have been previously published in:

J.-T. Sohns, C. Garth, and H. Leitte. “Decision Boundary Visualization for Counterfactual Reasoning”. *Computer Graphics Forum* 42.1 (2023), pp. 7–20. DOI: 10.1111/cgf.14650

J.-T. Sohns, D. Gond, F. Jirasek, H. Hasse, G.H. Weber, and H. Leitte. “Embedding Space Explanations of Learned Mixture Behavior”. *Proc. 3rd Conf. on Phys. Modeling for Virtual Manufacturing Systems and Processes*. 2023, pp. 32–50. DOI: 10.1007/978-3-031-35779-4_3

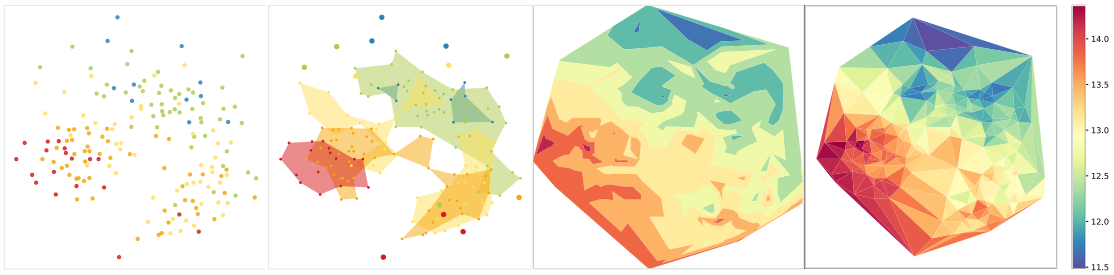
Chapter 4

Attribute-Based Explanations of High-Dimensional Data

The principle of exploratory data analysis requires presenting a raw data plot that convincingly supports each drawn conclusion [160]. In many applications this raw data plot is an embedding of the high-dimensional data using methods like linear projections (e.g. PCA) or non-linear techniques like multidimensional scaling (MDS) or t-distributed stochastic neighbor embedding (t-SNE). As we have seen in Section 2.1, linear techniques have the advantage that the resulting axes still have meaning, but the advances in the previous chapter affirmed that they are too hard to read for more than a few dimensions. As such, they are unsuitable to uncover structures in higher-dimensional space. Non-linear projections often nicely reveal complex structure in high-dimensional space, but no longer offer direct annotation of the projected space. Hence, it is paramount to equip these widely used techniques with mechanisms that help users properly read the projected data and relate original data attributes to the computed features.

Going through examples in high-dimensional data analysis libraries [163] and state-of-the-art reports [43], we observed that the gold standard for this task is still color-coding glyphs and 2D interpolation-based scalar field reconstruction [164, 38]. Figure 4.1 compares these standard techniques for an MDS embedding of an established example dataset [161, 162]. Color-coding assigns each projected data point a color using one of the original attributes, cf. Figure 4.1 (a). This technique is easy to implement and to comprehend, but suffers from occlusion and visual clutter and makes outlier detection difficult [37]. Scalar field reconstruction techniques [164] reconstruct a 2D scalar function for a given attribute, which serves as input to a spatial color-coding, cf. Figure 4.1 (c) + 4.1 (d). This results in readily visible spatial patterns, but cannot account for the fact that data points with different attribute values are projected onto the same position in 2D space.

Nonato et al. [38] survey the state-of-the-art in layout enrichment techniques that augment the embeddings with additional information. Many powerful augmentation techniques were presented in the last decades, focusing particularly on enrichment of the point cloud with additional information such as cluster annotation and automatic la-



(a) Glyphcoding (b) Rangesets (c) Contours (d) Triangulation

Figure 4.1: Same data, different chart types: All charts present the same MDS embedding of the wine dataset [161, 162] with various augmentation strategies for the attribute alcohol. (a) Colorcoding correctly represents that data but suffers from occlusion, clutter, and poor outlier highlighting. (c+d) Field-based techniques cannot correctly depict projection ambiguity. (b) The proposed rangesets combine the advantages of both techniques.

beling. In our new technique, we combine the strengths of these two directions. Our goals are to design a technique and system that are easy to use and comprehend, that are applicable to all types of embedding techniques, that can be directly integrated into existing analysis pipelines, and that enable the user to quickly and correctly understand attribute value distributions in the embedded data. We also remark that the goal of this approach is not to explain the projection, i.e., the precise nature of the data transformation from high-dimensional to 2D space, as is done for example in [165, 166] who visually encode least-varying dimensions, but only the final outcome, i.e., the embedding of the data points in the plane.

In this chapter, we present NoLiES, an interactive system that enables the user to explore linear and non-linear embeddings with respect to the original attributes. NoLiES inherently relies on a novel augmentation technique for multidimensional projections, which we call rangesets. Rangesets, as presented in Figure 4.1 (b), build upon non-convex α-hulls to visually group data points with similar attribute values, allow for outlier filtering, and user defined adjustments. Section 4.3 gives details about the methodology. To answer questions relating to multiple attributes, an interactive analytics system is required, which will be presented in Section 4.4. Several use cases, a real-world application problem of machine learning in thermodynamics (Section 4.5), and feedback from an informal expert user study (Section 4.6) are provided to demonstrate the capabilities of the novel technique. In summary, our contributions are as follows:

- We review the state-of-the-art of direct augmentation techniques for multivariate projections and highlight strengths and interpretation challenges.
- We detail a novel visualization technique (rangesets) to augment embeddings.
- We detail the connection of rangesets to algebraic topology and provide first steps on how this theory can be used to further improve augmentations of embeddings.
- We present an interactive analysis system and detail the analysis workflow to interpret linear and non-linear embeddings, along with several case studies using examples from machine learning databases and real world applications.

4.1 Related Work

NoLiES builds upon concepts for augmented multidimensional projections, improved scatterplots, and topological analysis of high-dimensional data, which we will review in the following.

4.1.1 Augmentation of Multidimensional Projections

Non-linear dimensionality reduction techniques are widely used for data exploration [38]. Much of the work centers around finding better projections, controlling and communicating error, and automatic detection and visualization of features. A critical aspect that receives far less support is the interpretation of projected data. Nonato and Aupetit [38] structure the augmentation techniques in *direct enrichment*, *cluster-based enrichment*, and *spatially-structured enrichment*. Further, we will discuss the directions of trying to reconstruct *continuous scalar fields* or *descriptive axes*, which are orthogonal to the three categories above.

Direct techniques augment the embedding for example with attribute-based color and text labels to provide additional information [43]. A variety of techniques exist using color-coding directly on the glyphs [167, 168, 55, 169], hexbin visualizations to estimate local densities [170], and multivariate glyph-based approaches [171, 172]. All these techniques share that they can represent only attribute values of a single data point per pixel. However, most multidimensional projections (linear and non-linear alike) suffer from projection ambiguity, i.e., data points with different attribute values are projected onto the same 2D coordinate, which needs to be communicated.

Cluster-based techniques cluster points in the embedding by proximity into non-overlapping regions and enrich the visualization with information about the aggregated cluster [173, 172]. As the clustering is performed in visual space, most techniques fail to account for distortion and resulting false neighbors. Clustering in high-dimensional space and enriching the projection accordingly has also been researched, but has to deal with uncontrolled fragmentation due to the non-linearity of the projection [169].

The third kind of methods, **spatially-structured enrichments**, first partition the space using techniques such as Voronoi diagrams or treemaps and afterward enrich these regions [174, 175, 176]. Most of these applications, however, aim at a non-overlapping partitioning, which we specifically want to integrate to correctly reflect the nature of the data.

Several techniques aim at the reconstruction of **continuous scalar fields** that can be directly visualized using field visualization techniques, e.g., probing projections [55] or data context map [177]. These spatial techniques directly avoid overplotting and group coherent regions [37], but need special strategies to cope with projection ambiguity as will be discussed in more detail in Section 4.2.1.

The last line of research that builds upon the concept of a continuous field tries to recover the **non-linear axes** or illustrate regions of maximal attribute values. The data context map [177] enriches the embedding with additional data points that locate regions of high attribute values and augment the visualization with additional attribute-based contours on the reconstructed scalar field [177]. DimReader [178] augments the embedding with non-linear grid lines and prolines [179] display the non-linear axes. The t-viSNE system [180] presents an analytics tool to explore t-SNE projections [54] using for enrichment color-coded glyphs augmented with interactive exploration. An excellent overview of interaction with dimensionality reduction is given by Sacha et al. [99]. The techniques in this last category are complimentary to our approach and can be combined with the here proposed rangesets.

4.1.2 Improving Scatterplots

Scatterplot visualization and its challenges like overplotting and clutter have been researched in their own right [181]. Micallef et al. [182] present techniques to optimize parameter settings for scatterplots like size and opacity to automatically improve the visual results.

Contours have been applied in a variety of approaches to represent set relationships in scatterplots to improve perception and to lower cognitive load [183, 184, 177, 37]. Bubble Sets [183] employ density estimates and contour lines to determine outlines for a set. Butterfly Plots [185] use convex hulls and refined convex hulls to enclose data points of the same class. Simonetto et al. [184] start with a planar graph connecting the points of a set for which a geometric hull is computed. From a theoretical perspective, this approach is similar to ours, though we assume more data points per set, which allows us to go for geometry directly generated through a triangulation.

Mayorga and Gleicher [37] present Splatterplots that combine density plots with contours to solve the overplotting problem for large numbers of points and at the same time be agnostic to outliers. This approach is conceptually very similar to ours and inspired the presented rangesets. Challenges that we wanted to further improve are locally non-uniform density and cognitive load. A more detailed discussion is given in Sect. 4.2.1.

4.1.3 Topological Methods for High-dimensional Data

Topological methods have a long history in high-dimensional data analysis [43]. Commonly, they are applied on high-dimensional data in a preprocessing step to extract relevant structures, e.g. the contour tree [186, 1] or topological features [55, 1, 187]. This helps to simplify the data and create abstractions that are easier to represent [187, 55, 188]. Topological methods have also been used to control and evaluate the projection process [189, 176, 190]. One of the more prominent approaches is integrating Voronoi cells for quality control [176]. By using the dual-graph of the Voronoi diagram, the Delaunay triangulation [191], we can draw from the well researched theory of algebraic topology [192, 193]. A summary of this theory and how it relates to our approach are given in Section 4.3.3.

4.2 Problem Definition

In this section, we discuss the challenges we faced when using state-of-the-art techniques to explore non-linear projections and how we overcame them. Starting point was the data displayed in Figure 4.2, which consists of 240 points in 4D. Using standard approaches it was hard to relate the structure in the point cloud to the input variables or domain knowledge. Therefore, we first discuss the existing approaches in an applied way and then formulate our proposed system.

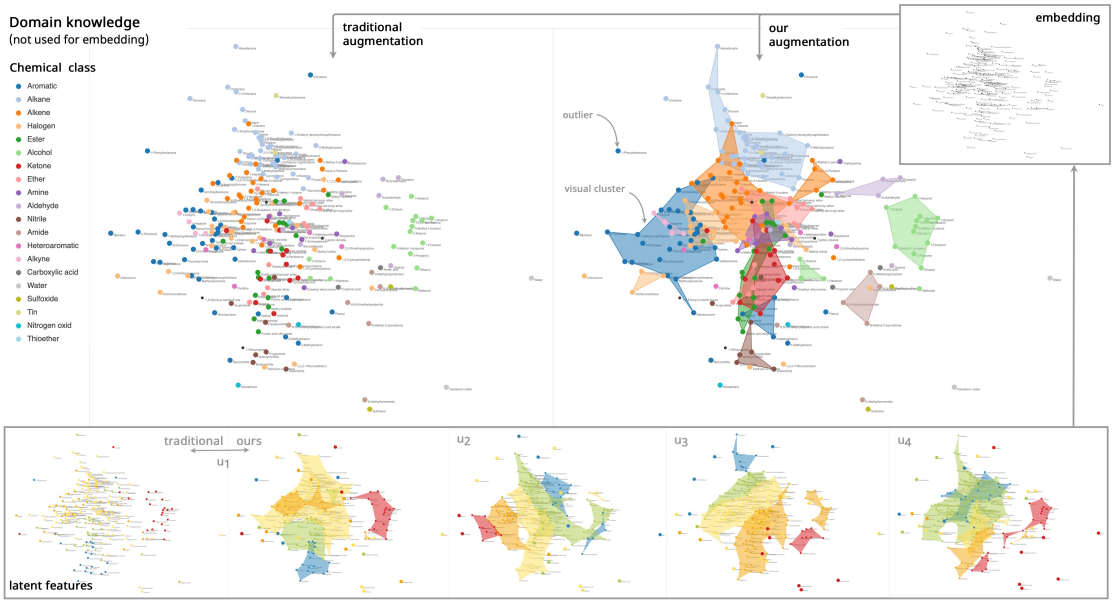


Figure 4.2: Explaining machine learning: The embedding (MDS, top right) shows a non-linear projection of the 4D latent feature space as computed by machine learning for 240 chemical compounds [20]. Color-based augmentation helps relate structures in the embedded point cloud to input attributes (bottom, continuous variables) and domain knowledge (top, categorical variables). Our new approach enabled domain scientists to detect previously unrecognizable patterns and outliers in their data. A higher-resolution version of this figure is available in the Appendix.

4.2.1 Discussion of Existing Approaches

Giving a practical view over the enrichment options, we review the three most widely-used concepts for scatterplot augmentation: Glyph-based colorcoding, scalar field reconstruction, and set-based visualization. For illustration purposes, we use the UCI wine dataset [161, 162], which features the same challenges as our domain dataset, but is more widely known in context of model predictions. With this dataset, we have a reproducible foundation to discuss the enrichments shown in Figure 4.1.

Glyph-based colorcoding, cf. Figure 4.1 (a), is the easiest to implement and gives an intuitive sense of value locations, but suffers from occlusion and overplotting as detailed before [37]. It is difficult to assess the amount of overlap and to detect of outliers rapidly.

Another approach reviewed earlier is reconstructing a scalar field from the point data. Two popular methods are levelsets and triangulation-based [191] renderings. As both methods are implemented in *matplotlib* [194], we show the respective plots in Figure 4.1 (c) and Figure 4.1 (d). While both techniques give a good sense of attribute value distributions, they have difficulties in representing regions that suffer from projection ambiguity, i.e., where points with different attribute values are projected onto or close to each other. Here, field-based approaches either have to work with local averages (isocontouring) or create many small color patches (triangulations).

A third concept that is used particularly for categorical attributes on scatterplots are cluster- or set-based visualizations [38]. The two primary directions in which the outlines of sets can be obtained are geometric/algebraic approaches and statistical approaches. The geometric approaches operate on a simplicial complex (graph or triangulation) and derive the boundary from this construct through filtering or additional geometric operations like dilation. The convex hull of a set of points is an example of a geometrically created boundary. Statistical approaches rely on a density estimate for which an isocontour is drawn. Both approaches, statistical and geometric, rely on parameters that control the outlier filtering. In the statistical case, this is achieved through the isovalue of the boundary contour. A known problem here is the handling of point clouds with locally varying density, which are treated, for example, with adaptive-KDE-methods [195], which, however, are hardly implemented in data analysis software packages. Geometric approaches, on the other hand, are commonly controlled by geometric criteria like maximal distance between two points to identify outliers. The quality of the extracted structures (contours and outliers) strongly depends on the quality of the data.

We can conclude that all methods have use cases where they are best suited: Glyph coloring is easy to implement and works well with a limited number of data points; statistical approaches carry the interesting notion of probability that points may be located in a certain area of the plot; and scalar-field techniques can handle an arbitrary number of color-levels if this is requested. In Section 4.3, we present rangesets, a geometric set-based approach with outlier highlighting that uses the strengths of spatial augmentations, can handle projection ambiguity, and is intuitive to control by a single distance parameter with good default value heuristics.

4.2.2 Objectives

The survey by Nonato and Aupetit [38] gives an excellent overview over analytical tasks performed using multidimensional projections. NoLiES supports tasks relating to the two main categories *Explore Dimensions (Axes)* and *Explore Items in Enriched Layout*, i.e., it helps understand the mapped data and explore structures in the projected data respectively. In particular, we support the following fine-grained tasks (names in italic are tasks as defined by Nonato and Aupetit):

- **Explore dimensions** (*Map Synthesized Dimension to Original Dimension, Discover Relation btw. Visual Pattern & Original Dim.*): We provide a visual link between the position in 2D space and the original data attributes, which explains the attribute value distribution in the projected data.

- **Cluster-based analysis** (*Name Cluster, Discover Clusters in Map, Match Clusters and Classes in Map, Brush in Data Space*): The user observes regions in the embedding that are denser than their surroundings, i.e., clusters. They now want to understand what discriminates the cluster from the surrounding data and what are characteristics that points in the cluster have in common. Small multiples and interactive selections that are shared between all plots support these tasks.

- **Outlier analysis** (*Discover an Outlier in Map, Discover Class-Outlier in Map*): Outliers are points that are unusually far away from other points compared to average point distance. Here, the user would like to understand what sets this point apart from the surrounding points. We support this task by outlier highlighting and the cluster strategies as above.

4.3 Method

The centerpiece of NoLiES are the proposed rangesets, which we introduce in this section. We first describe the contour extraction algorithm, then detail the relevance and benefits of their relation to levelset computation in algebraic topology, and close with the design decisions for visual encoding.

4.3.1 Rangesets

We propose what we call rangesets, a geometric set-based approach to combine the strengths of spatial and point-based augmentations, can handle projection ambiguity and can be controlled intuitively by a single distance parameter.

Rangesets use geometric contours to visually group data points with similar values while still highlighting outliers as individual points. To explain attributes in an embedding that are inherently categorical, a colored contour is drawn per category, cf. Figure 4.4. For continuous attributes, we first discretize the value range of the attribute and then draw a contour for each bin, i.e., for each range of attribute values, cf. Figure 4.8. In the following, we will describe the algorithm in the case of a continuous attribute, as this includes the extra discretization step. The algorithm for the augmentation of a categorical attribute follows analogously. The algorithm consists of five steps as illustrated in Figure 4.3:

1. Select the attribute bin
2. Filter for data points in the respective range
3. Compute Delaunay triangulation of the filtered points
4. Remove triangles with unwanted properties
5. Compute boundary of the triangulation and find all points in the current range that do not belong to the contour for highlighting

We describe the algorithm from an overview to details. We start with the target contour and then discuss the filtration parameter ϵ, alternative filtration approaches, and the discretization.

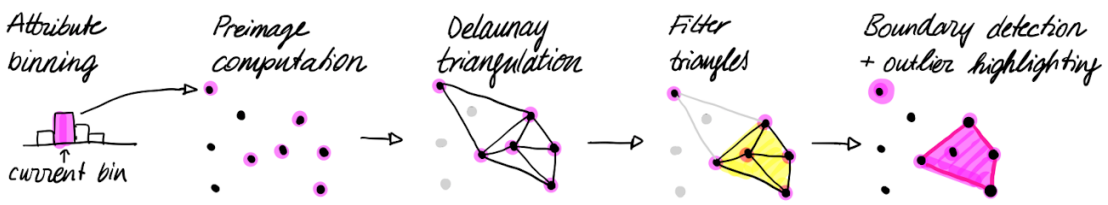


Figure 4.3: Rangeset contours are computed in five steps. The first step of attribute binning is only necessary for continuous attribute data.

4.3.2 Non-Convex Hulls

Given a set S of n points in the plane (n being a positive integer), we are looking for a hull that tightly encloses the points of S and allows for filtering of outliers. Non-convex or minimum area hulls [196] feature these properties. Following the discussion for the choice of a non-convex hull for scagnostics [197], we use α-shapes [192] as they can be computed efficiently and allow for filtering of outliers. This concept was previously used to partition projections in 2D by Joia et al. [198], though we propose to extend it to attributes of the higher-dimensional space, utilizing more of its benefits. We first revisit α-hulls as the mathematical base for α-shapes, then demonstrate their efficient computation and finally detail the implications of parameter choice.

α-hulls are a generalization of convex hulls with the α parameter defining the "tightness" of the hull around S. For $\alpha = 0$ the α-hull of S is the convex hull of S. Positive α-values result in "loose" hulls that extend beyond the convex hull. Negative α-values result in "tight" hulls that are non-convex. We are interested in the non-convex part. For an arbitrary negative real number α, the α-hull of S is defined as the intersection of all closed complements of discs with radius $-1/\alpha$ that contain all the points of S. The α-shape of S can be inferred by connecting neighboring points on the boundary of the α-hull, Edelsbrunner et al. [192] call them α-extreme points, with straight lines.

They further show that α-shapes can be computed efficiently from the Delaunay triangulation [191] of the point set S by excluding triangles that contain an edge whose length exceeds a given threshold ϵ. The filtered Delaunay graph $\mathcal{D}_\epsilon$ now has the vertex set $V = \{0, 1, ..., n-1\}$ and an edge set of $E = \{(u, v) \in V \times V | d(u, v) \leq \epsilon\}$, i.e., two vertices u and v are connected if and only if their distance is less than or equal to the selected distance threshold ϵ. For our application of embeddings in 2D space we use the Euclidean distance $d(\cdot, \cdot)$. Like the α value, the parameter ϵ also controls the "tightness" of the computed contour like, but is more intuitive. Large values include all edges of the Delaunay triangulation resulting in the convex hull [191] of the point set. Small values will result in a stronger fragmentation.

An example that illustrates the progression of the contours with increasing ϵ-values is shown in Figure 4.5 (b) - (e). Here, all points of the wine dataset are included in the contouring process and the resulting contours are shown for increasing ϵ-thresholds. Note how the set of contours changes from small fragments, over a smooth tight boundary, to the full convex hull. The choice of ϵ has strong effects on the readability and message of final augmentation for the embedding where multiple contours are drawn. An example

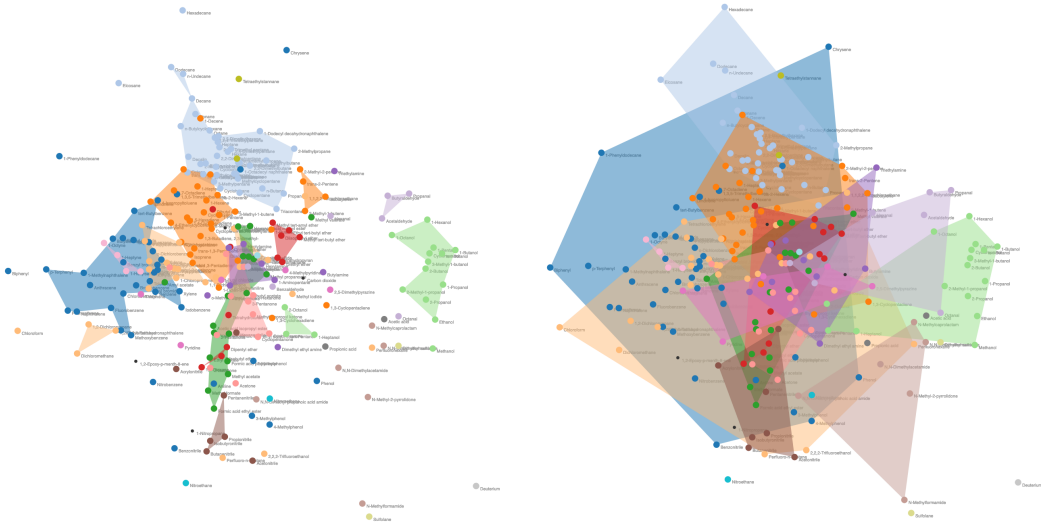


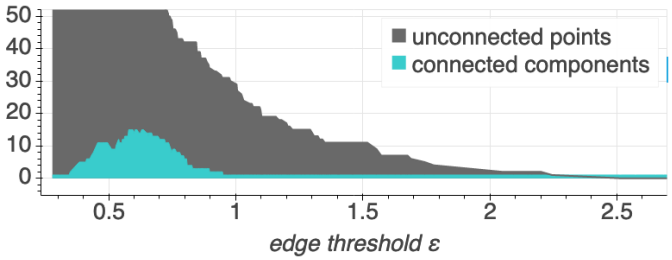
Figure 4.4: Effects of the distance parameter ϵ on final layout: The thermodynamics embedding is augmented with color for 20 chemical classes. (left) Small values for $\epsilon = 0.75$ result in highlighting the core regions of the classes. (right) Large values $\epsilon = 10 \geq \epsilon_{max}$ result in convex hulls that strongly overlap.

with multiple contours is given in Figure 4.4 for the thermodynamics dataset where 20 categorical classes are used to explain the embedding. For small ϵ-values, the contours for each class focus on dense core regions (left). For large values, the contours overlap strongly and are no longer helpful (right). Finding a good ϵ-value is an important aspect of the algorithm and will be discussed in the next section.

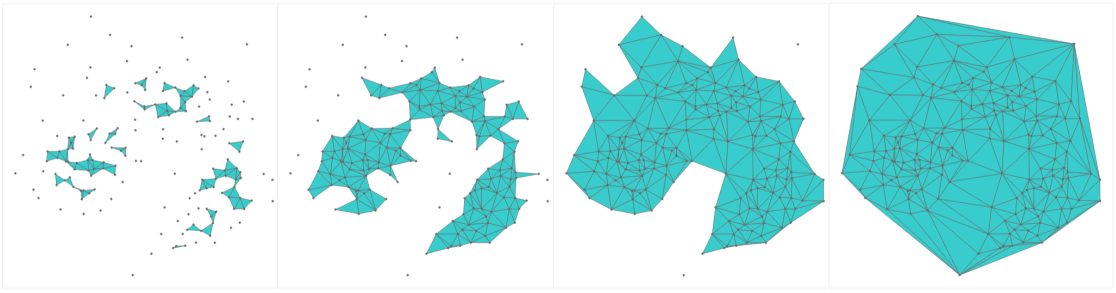
4.3.3 Topological Filtration

To better understand the progression of the size of the contour, we now look at it from the perspective of algebraic topology. The algorithm outlined above induces a filtration of the simplicial complex given by the Delaunay triangulation $\mathcal{D}$, i.e., the subcomplexes $\mathcal{D}_{\epsilon_1}$ and $\mathcal{D}_{\epsilon_2}$ form a nesting hierarchy with $\mathcal{D}_{\epsilon_1}$ being a subset of $\mathcal{D}_{\epsilon_2}$ if and only if $\epsilon_1 \leq \epsilon_2$. On the operational level, this results in the nice property that increasing ϵ can only add triangles to the non-convex hull, but never remove them. For $\epsilon = 0$, no contour is created and each data point is an outlier. For $\epsilon = \epsilon_{max}$, where ϵ_{max} is the longest edge in the Delaunay graph, we obtain the convex hull. For ϵ-values within this range, the number of outliers as well as the number of contours will change monotonically.

Algebraic topology is a research field that studies topological properties and changes of the simplicial complex under filtrations. Specifically, Betti-numbers describe the connec-



(a) Component count wrt to max edge length ϵ



(b) $\epsilon = 0.6$ (c) $\epsilon = 1.0$ (d) $\epsilon = 2.0$ (e) $\epsilon = \epsilon_{max}$

Figure 4.5: Parameter study for ϵ: The contour algorithm connects all points that have maximal distance ϵ. (b) Small values result in fragmented contours. (c) & (d) Increasing ϵ connects the components (e) Distances larger than the longest edge length result in the convex hull.

tivity of simplicial complexes. For our analysis, we consider the zeroth Betti-number b_0, which captures the number of connected components. In simply connected domains like ours, this information is often presented as a contour tree [199], which depicts the merging of the contours as ϵ changes. As the more critical information for our application is the number of outliers and connected components for any ϵ, we provide the topological summary in a histogram over ϵ. An example of this chart is given in Figure 4.5 (a).

The topological summary depicts a marginal of the contour tree differentiating between connected components containing exactly one point and those containing multiple points. Connected components with more than one point are depicted in blue and the gray area represents isolated points, which would be outliers in the algorithm. The plot is truncated at the top as in the beginning $\epsilon \in [0, 0.7]$ most data points are outliers. The graph presented here is typical for all datasets that we investigated. With $\epsilon = 0$, all data points are outliers. With increasing values $\epsilon \in [0.2, 0.7]$ many small contours form, which eventually merge to larger, more stable contours. Starting from $\epsilon = 1$, we only have one contour and the threshold parameter only controls the number of outliers. With $\epsilon = 2.25$ we reach the longest edge length of the Delaunay graph ϵ_{max}, where all points are included in the contour.

Note that the presented ϵ-values are not universal as they refer to edge length as computed by the embedding, which can vary or even be arbitrary depending on the embedding source. Scaling the embedding to a fixed size can help, but one still has to account for varying aspect ratios and point densities due to number of points. Wilkinson et al. [197] propose a default value for ϵ based on edge lengths in the minimum spanning tree (MST):

$$\epsilon = q_{75} + 1.5 \cdot (q_{75} - q_{25})$$

where q_{75} is the 75th percentile of the MST edge lengths and the expression in the parentheses is the interquartile range of the edge lengths. It is important to note that the minimum spanning tree of a set P of point sites (in any dimension) is a subgraph of the Delaunay triangulation. Hence, we use this criterion to provide a default value for the filter threshold ϵ.

4.3.4 Discussion of Triangle Filter Criteria

The algorithm described above relies on maximal edge length as the distance function for contour filtration. There are, however, alternatives. The three most widely used local criteria to control triangulation quality in general are edge length, triangle area, and inner angles. Optimizing inner angles is already ensured through the Delaunay triangulation, which creates an angle optimal triangulation, i.e., from all possible triangulations the one with largest smallest angle is chosen [191]. Hence, we disregarded this approach as we already ensured, that long spiky triangles are avoided as much as possible.

During the development of NoLiES we considered filtering based on the area of triangles, i.e., triangles and their bounding edges are included in $\mathcal{D}_\epsilon$ if they do not exceed the threshold size ϵ. The goal was to exclude large triangles that cover space without additional points inside the triangle. A comparison of the two distance metrics can be seen in Figure 4.6. This choice of metric, however, proved to be much harder to control and resulted in unexpected holes in the hull and long spiky triangles not being removed. Hence, we use the edge-length-based filtering as default criterion.

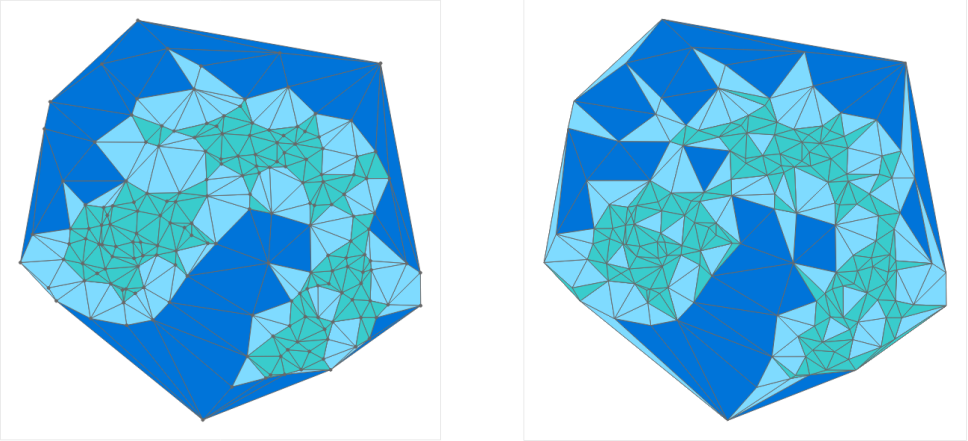
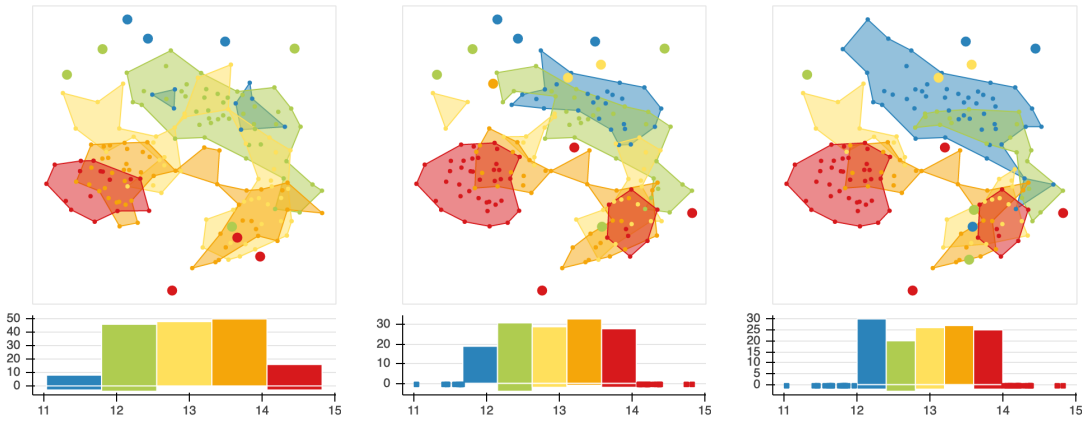


Figure 4.6: Comparison of filtration criteria on similar thresholds: (left) Edge-length-based filtration with thresholds of 0.9, 1.55, ϵ_{max}. (right) Area-based filtration with thresholds 0.2, 0.5, Δ_{max}, which features unwanted properties like spiky triangles and holes.

4.3.5 Attribute Range Discretization

The contour algorithm described so far computes the contour for a single bin of the attribute histogram and, hence, data binning plays an important role for the rangesets. Histogram computation has been intensively studied in the statistics community, and we can draw from excellent prior research [200]. A histogram depends on the following parameters: total range of the histogram, number of bins for the discretization, step size of the discretization (uniform vs. non-uniform), and handling of values outside the selected range. We provide default values for all attributes following the reasoning below that can be overwritten by the user in the Jupyter Notebook. A screenshot of the entire system including small-multiples with histograms is given in Figure 4.10.

Initially, the histogram range for each attribute is set to the full value range of the data. Adjustments may be necessary to obtain bins with easy to read values or to account for the range of possible values, e.g. setting to 0-100% when occurring values only range from 37-82%. In NoLiES, this can be done interactively via range-sliders in the attribute panel in Figure 4.10 (left). Upon dragging the slider, the contours update interactively giving instant feedback on the effects of the changes. An example of this interaction process is depicted in Figure 4.7. In our experiments, the contours were fairly stable and the exact choice of the absolute range was not too critical. Customized ranges can be stored in the notebook and are automatically used in the next iteration. Automatic outlier detection [201] and choosing thresholds with human-interpretable values [202] may be used to further automize the process.



(a) [min, max] (b) [11.7, 14.1] (c) [12, 14]

Figure 4.7: Effects of changing range boundaries interactively: Rangesets for alcohol content in the Wine dataset with various ranges. Values below and above the extrema are included in the extremal bins.

We discretize the selected range into five bins with equidistant boundaries in the default case. We found five bins sufficient to model the underlying distribution of our examples, as shown in Figure 4.7. Using more contours resulted in hard to read images which aligns with the findings by Kraus et al. [203], who report difficulties in precise readings of continuous maps of scalar fields. A comparison of a perceptually uniform continuous colormap and our discrete one is given in Figure 4.8. We are aware that "rainbow"-colormaps are a controversial choice. However, the colors in this colormap are easy to name and highly distinct, which simplifies the comparison in a small-multiples setting like ours and allows for unambiguous communication. Hence, we assign each of the five bins a fixed color and label [very low, low, medium, high, very high]. Again the colormap can be easily changed to personal preferences in the notebook as the Bokeh-library offers a rich set of default colormaps to choose from. We explored non-uniform discretization to better adapt to local structures in the point cloud, but found this to be misleading when interpreting the color distribution.

An additional augmentation of the histograms in NoLiES is the use of the negative y-axis. Data points outside the bin ranges are encoded as dot-glyphs below the y-axis. Data points that are outliers for the particular rangeset bin, i.e., their distance to the nearest point is larger than ϵ, are counted and encoded as bars extending in the negative y-direction. In this way, the user can directly see how parameter settings affect the topology of the rangeset chart. The encoding is well trackable in Figure 4.7.

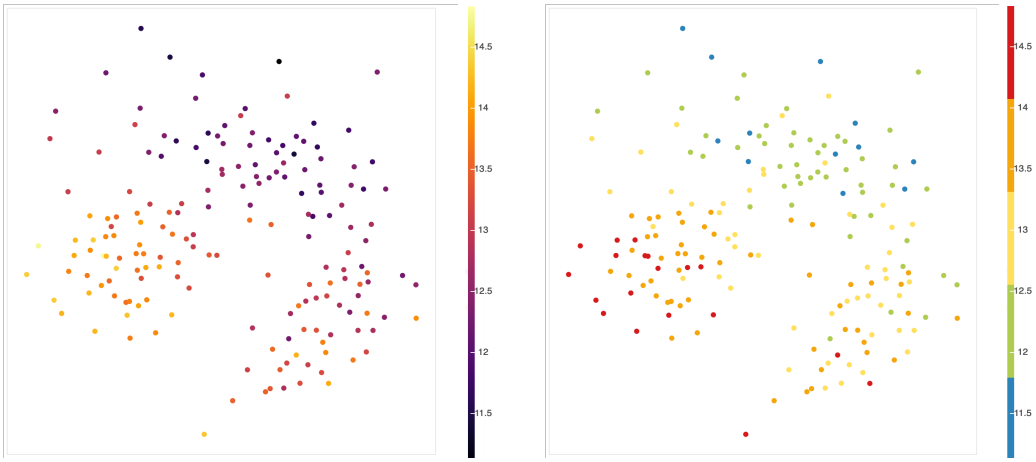


Figure 4.8: Comparison of continuous and discrete colormap: (left) Color-coded glyphs using Matplotlib's perceptually uniform *plasma* colormap vs (right) a discretized colormap based on Bokeh's *Spectral5*.

4.3.6 Visual Encoding

To visually represent the rangesets, we augment the embedding with polygons and point-halos. The contour for each bin is rendered as a filled semi-transparent polygon. The data points, within and outside the contours, are rendered as point-glyphs. Data points contained in a contour provide the user with a sense of data density. Data points outside a contour, outliers, are highlighted by increasing them in size. Additionally, they are drawn on top of the polygons to avoid them being hidden behind multiple semi-transparent layers.

4.4 System Design

The system, which we call NoLiES, is divided into three modules: the rangeset module, the notebooks, and the GUI. The rangeset computation is written as a stand-alone module. Thereby, the theoretical contribution of this chapter can be used outside the remaining software. The other two modules are steps in a workflow to support the visual interpretation of dimensionality reduction schemes. This workflow is illustrated in Figure 4.9. The notebooks provide the data scripting and static charts and the GUI provides an interactive web-application for detailed analysis. When talking about NoLiES, we primarily refer to the web-application.

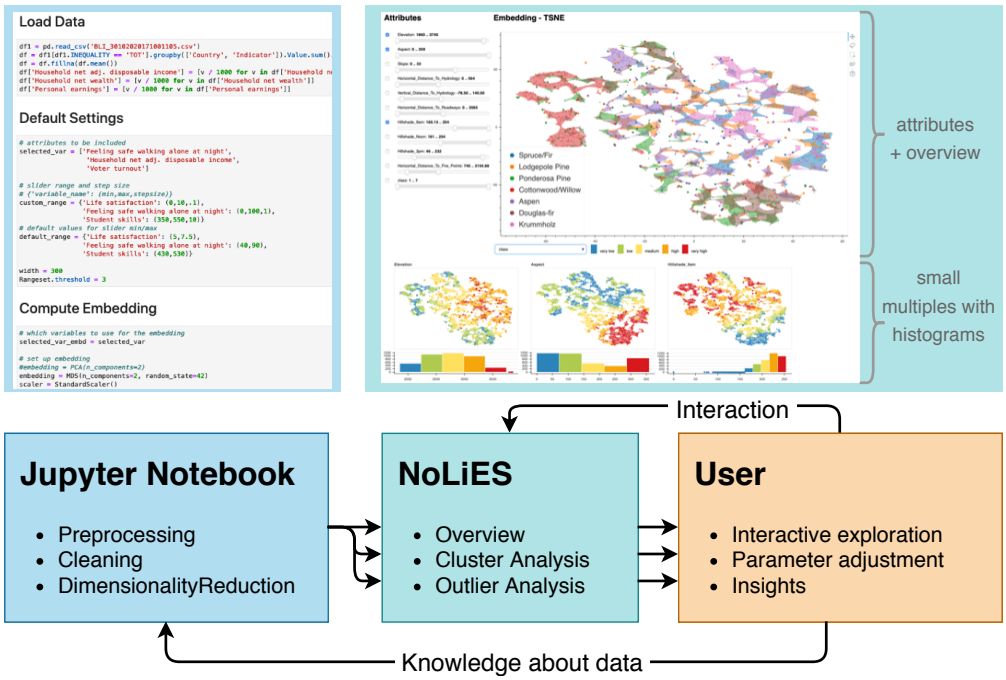


Figure 4.9: Analysis workflow: NoLiES is implemented in Jupyter Notebook, which is well suited for script-based data preprocessing and can be served as an interactive web-application. Knowledge that the user obtains during the analysis process, e.g., the best parameters or color codes, can be stored in the notebook.

Jupyter Notebook The notebook comprises data preprocessing and knowledge storage, cf. Figure 4.9 (left). It provides three sections that guide the user through the analysis process. The first section handles data loading, preprocessing, and cleaning. Additionally, we include placeholders for commonly used system parameters like custom slider ranges, attribute filters, and the rangeset threshold. The second section handles the multidimensional projection. Multiple widely used dimensionality reduction methods as implemented in the scikit-learn library [163] are included and can be selected by the user. We also include best practices for dimensionality reduction like attribute scaling and tests for correlation. The third section takes care of the visual design and interactions. We advise creating one notebook per dataset and using it as a storage location for knowledge that was derived during the analysis process.

Interactive App After data preprocessing, the user changes to the interactive GUI view in the browser, cf. Figure 4.9 (right). In this view, only the chart elements from the notebook are visible and are now linked interactively. The GUI consists of three major components: (i) An attribute view that lists all included attributes from the raw data, their ranges, and the selected sub-ranges to be used for the binning. To help the user

assess truthfulness of the projection we include the projection quality [204] as an auxiliary attribute. (ii) The embedding renders the projected data as a point cloud with optional labels. The user can interactively alter the displayed rangesets in a dropdown menu. The title of the chart automatically includes the applied projection method as defined in the notebook. (iii) The small multiples view provides a quick overview over all selected attributes distributions and present the histograms for the binned attributes. Views can be interactively switched on and off in the attribute view with a checkbox. This is particularly useful if the dataset contains many attributes and only few of them need to be compared.

The attribute sliders are interactive and upon moving the sliders, the outlines and the histograms are updated interactively. This procedure helps to understand attribute value locations and the effect of the discretization. The user can also use this technique to manually filter for outliers and set tighter value bounds for the displayed contours. Once they found good default values, these can be stored in the notebook and will be set at defaults in the next run.

To ease the comparison between many attributes, we connect all data views with respect to selections. Often this is done by color or alpha-value manipulations, but these channels are already heavily used in our design and are no longer salient. We integrate a gray outline curve that encloses all data points selected with a Lasso tool. The selection curve is optimized with the rangeset algorithm and shared between all plots. See Figure 4.10 for an example. To not obscure the underlying information, the outline curve is offset by edge width [205]. Using this outline, the user can quickly compare selections across multiple views and directly find the region of interest.

Implementation NoLiES is implemented in Python in the *Jupyter* environment [206]. As the GUI is realized with the *panel* library [146], the application runs both within the notebook and as a stand-alone application showing only the GUI. Charts are created using *Bokeh* [145], Geometric operations with *shapely* [207]. Multidimensional projections and data preprocessing are provided through *scikit-learn* [163]. NoLiES is available on GitHub[1].

[1] https://github.com/Jan-To/nolies

4.5 Case Studies

In the following, we present three case studies with increasing complexity. The Better Life dataset is easy to comprehend and follow. The forest cover type dataset contains many data points that are not easily separable. A real-world study in thermodynamics targets explainable machine learning and large number of classes.

4.5.1 OECD Better Life

The OECD Better Life dataset [208] measures 25 attributes for 40 countries. The goal is to understand, and consequently predict, which factors promote a society's well-being. For illustration purposes we chose 11 attributes that cover a general mix of topics. Figure 4.10 shows the NoLiES GUI with all attributes enabled in the small multiples view.

The prime attribute of the dataset is the self-reported life satisfaction on a scale of 0 to 10. The central chart renders rangesets for this life satisfaction, annotated over an MDS embedding based on all attributes (health, wealth, labor, safety, ...). We observe that life satisfaction strongly correlates with the visual clusters in the 2D embedding. The ■ very high and ■ low happiness classes form big clusters that contain most data points. The histogram for life satisfaction can be found in the center-left of the small multiples.

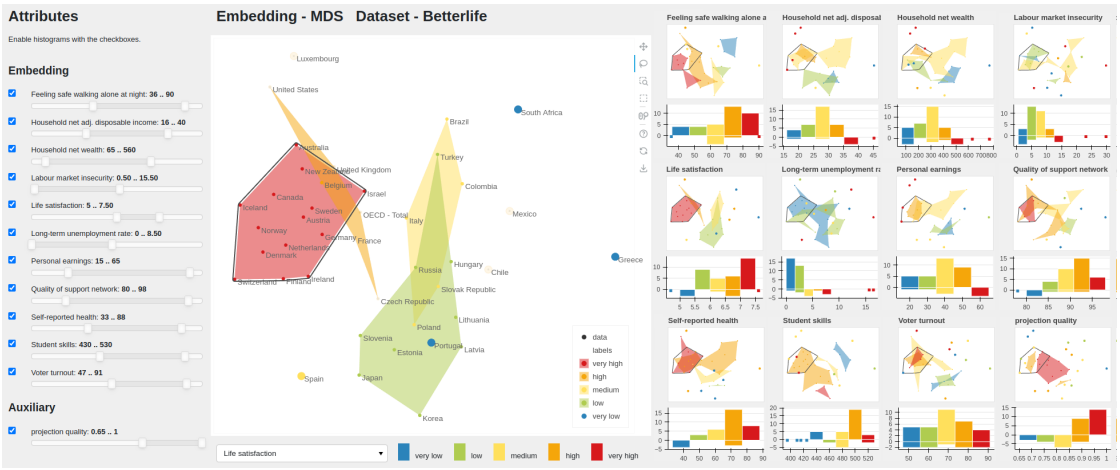


Figure 4.10: NoLiES on the OECD Better Life data [208] reveals multiple clusters that partially align with self-perceived life satisfaction (center). The gray selection transfers the ■ very happy countries to the small multiples of all attributes (right). Custom ranges are set with sliders (left) to achieve more expressive discretization. A higher-resolution version of this figure is available in Appendix.

We see that values range from 4.7 to 7.6. For better describable bins of size 0.5, we fix the range in the *life satisfaction* slider on the left to discretize between 5 and 7.5. The ■ very unhappy countries are spread across the plot. As such, they are not grouped in a rangeset contour, but form outliers which is reflected in the ■ negative bars in the histogram. We also observe outliers in the ■ high satisfaction class (Chile and Mexico) despite them being close to more unhappy countries. The projection quality chart on the bottom-right of the small multiples shows that the MDS projection can retain the neighborhood structure very well except for Luxembourg.

To obtain an overview over the distribution of assessed attributes, we look at the small multiples display in Figure 4.10 (right). All attributes are checked in the sliders, hence all are presented in the small multiples summary on the right. We observe that the attribute *Feeling safe walking alone* decreases from bottom-left to top-right. The histogram below the small multiple depicts that ■ red means $80-90\%$ agreement and ■ blue means less than 50% agreement to this statement. Other attributes like *Labor market insecurity* and *Student skills* feature a more complex and harder to describe distribution of attribute values, which is expected in an MDS embedding.

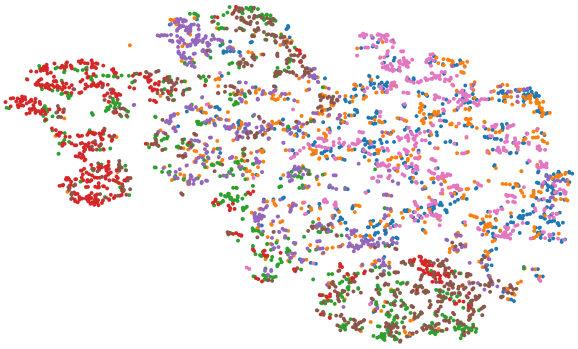
Next we focus on the gray outline, which was drawn by lasso selection in the large plot and contains all countries with ■ very high life satisfaction. Concentrating on the colors inside the gray outline for each small multiple, we can quickly observe that countries in the selection have diverse, but generally positive values in all attributes, except for *labor market insecurity* and *long-term unemployment rate* where lower values are better. In summary, we can state that countries with very high life satisfaction do well in all assessed categories. This also discriminates the ■ very happy cluster from countries in the ■ unhappy cluster, which have at least one problematic category with ■ low or ■ very low values; disposable income, self-reported health, and safety being the most prominent ones.

The US and Luxembourg form their own small cluster close to the very happy countries, but only rate themselves with ■ high life satisfaction. Comparing these two to the selection of very happy countries, we can identify attributes in which they diverge. Looking for color differences, we identify lower students skills and higher household net wealth than in the happier group. We conclude that this dataset suggests that wealth is not a significant factor for happiness and that consistently high values in all aspects of life is more important, leading to only high life satisfaction in these two countries.

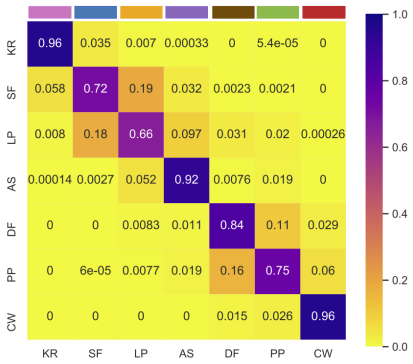
4.5.2 Forest Cover Type

The forest cover type dataset [209] covers 581k data points and 54 cartographic attributes like elevation, slope, and shade. The goal is to predict the 7 forest types using these attributes only. Blackard et al. [209] report 70% accuracy using a tiny neural network. The misclassification as depicted in the confusion matrix in Figure 4.11 (b) are extracted from their paper. Rows in the confusion matrix correspond to actual classes and columns to the predicted ones. We observe major misclassification between spruce/fir (SF) and lodgepole pine (LP) as well as douglas-fir (DF) and ponderosa pine (PP).

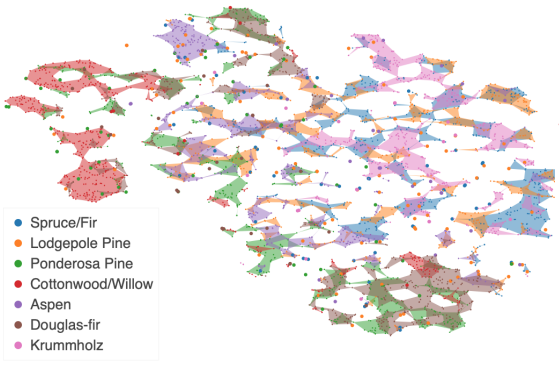
In order to make the machine learning predictions interpretable, an embedding view may help to explain the dataset. We create a balanced dimensionality reduction by sampling 600 data points from each of the 7 classes resulting in 4,200 samples. For



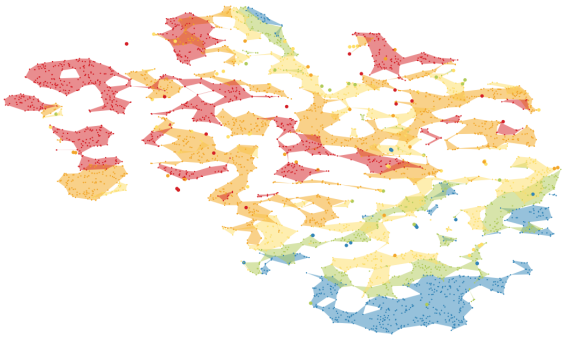
(a) Type of wood



(b) Confusion matrix

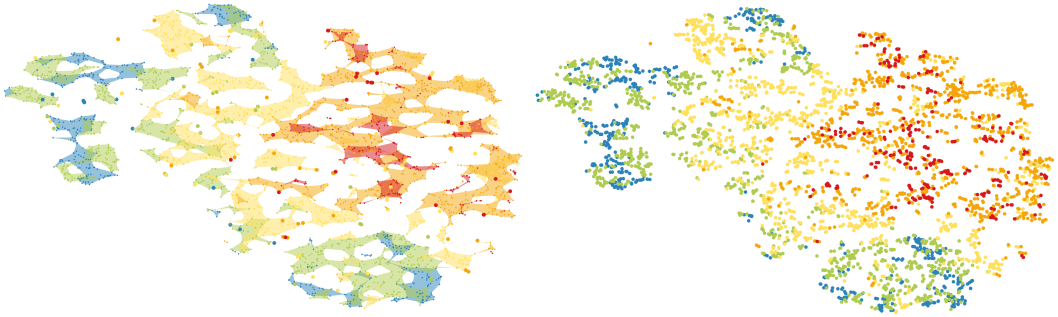


(c) Type of wood



(d) Shade at 9am

Figure 4.11: Forest cover type dataset with 4,200 data points and 9 attributes: 70% classification accuracy can be achieved for this dataset [209] with common confusion of spruce/fir and lodgepole pine (blue vs orange) as well as douglas-fir and ponderosa pine (brown vs green) (a) which is illustrated by overlapping contours in the embedding (b). (c) Original attributes help characterize classes and analyze the algorithm.



(a) Rangesets (b) Color-coding

Figure 4.12: Comparison of augmentation techniques on the attribute *elevation*: (a) Rangesets quickly outline regions of different parameter values. The size of outliers can be interactively increased. (b) Augmentation based on glyph-based color coding.

algorithmic simplicity, we compute the embedding from just the 10 numeric attributes. The discarded 44 categorical attributes one-hot encode the 4 wilderness areas and 40 soil types. Due to the higher number of data points and the $O(n^3)$ scaling of MDS, we chose the better scaling t-SNE-based embedding (perplexity: 30, early exaggeration: 30) in Figure 4.11 (a). The colors encode the ground truth class labels. To explore potential confusions, we render rangesets in Figure 4.11 (c). We observe that there is a lot of overlap between colors. On closer inspection, we find that only certain colors overlap, which is in agreement with the confusion of the model. We observe overlap, e.g., between ■ SF / ■ LP / ■ KR or ■ DF / ■ PP. Additionally, we observe that some colors only occur in particular regions, e.g. ■ red areas are located at top-left and in multiple small regions on the southern boundary of the ■ purple region.

It is important to note that structural analysis of t-SNE plots is challenging and may easily lead to misreadings [54]. The overlay with rangesets can help to counteract common misperceptions. As stated by Wattenberg, cluster sizes mean nothing in t-SNE [54]. With the overlaid attribute-based rangesets, the user can reconstruct the underlying distances between data points. Figure 4.11 (d) & Figure 4.12 augment the plot with two of the original attributes. Regions in the same color are close in high-dimensional space with respect to this attribute. While in Figure 4.11 (d) there is a continuous increase in shade at 9am from bottom-right to top-left, the ■ low elevation regions in Figure 4.12 are split into two groups by t-SNE; one at the top-left and one at the bottom-right of the embedding. We can thus deduce for the ■ Cottonwood/Willow cover type in the top-left of Figure 4.11 (c) that they mainly grow in areas with low elevation and high shade values at 9am. While these observations can also be made using classical glyph-based color coding, we found in our experiments that the trust in the observations was higher using rangesets – a detailed elucidation, however, is subject to future studies.

4.5.3 Matrix Completion in Thermodynamics

The prediction of fluid properties plays a central role in chemical engineering, e.g., for process design and optimization, since experimental measurements are usually cumbersome and expensive. Methods to predict the properties of binary mixtures are of particular importance, since the properties of multi-component mixtures can often be described based on information on the binary "submixtures" [210]. Matrix completion is a novel promising machine learning approach for this purpose [20, 21]. However, while data-driven matrix completion methods (MCM) yield great performance in predicting fluid properties, they are not intuitive and therefore difficult to understand from a physical perspective. This strongly reduces confidence among engineers and natural scientists and hampers their application. Hence, tools that enable a physical interpretation of MCM are paramount.

Since traditional approaches using glyph coloring and clustering failed in communicating any relationship between latent MCM features and domain knowledge, cf. Figure 4.2, we apply NoLiES for this purpose. Figure 4.13 shows MDS projections based on four learned MCM features of 240 solutes solved in 250 solvents. The model was trained on experimental data of activity coefficients at infinite dilution and a temperature of 298.15 K [20]. Hence, these MCM features constitute latent descriptors of the individual substances (solutes) that are derived only from mixture properties (activity coefficients).

The embedding features multiple clusters, which we overlay with rangesets of expert knowledge on the chemical classes of the solutes in Figure 4.13 (top). We observe that some chemical classes are very characteristic, e.g., alcohol, aldehyde, and nitrile, whereas the contours of others strongly overlap, especially in the center of the embedding. Also note that water and deuterium oxide (heavy water) appear as exceptional solutes on the bottom right edge, which fits well with their exceptional macroscopic properties. We learn that the chemical class of a solute is a suitable descriptor regarding the solute's MCM features and, hence, its activity coefficients. In addition, we find correlations of the MCM features with other physical descriptors, such as the solute's molar mass and polarity in Figure 4.13 (bottom).

Looking at the rangesets of the four MCM features in Figure 4.2 (bottom), we observe a clear spatial structure. A detailed analysis of the link between MCM features and physical properties of the solutes is done in the subsequent Chapter 5 and our application paper [4]. The results shown here already indicate that NoLiES offers exciting physical insights into latent MCM features, which can serve as basis for a targeted enhancement

of MCM, e.g., for selecting suitable physical descriptors to support the data-driven approach [211].

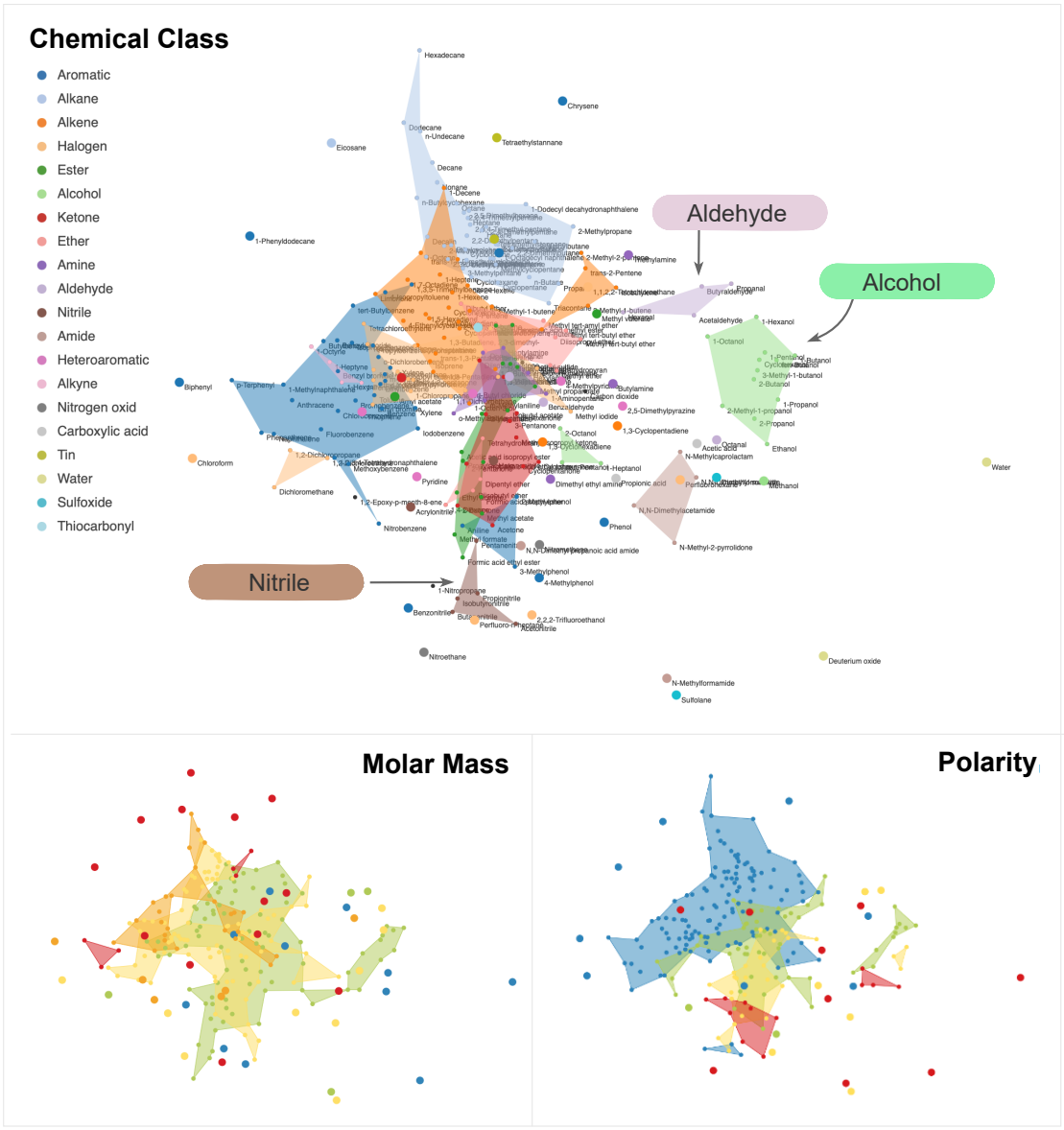


Figure 4.13: MDS-embedding of the four MCM features of 240 solutes trained to data for activity coefficients in binary mixtures [20]. The ground truth as captured by domain knowledge covers 20 chemical classes (color code, top-left) and information on the molar mass and polarity (bottom). Comparing domain knowledge and latent MCM features u_i (Figure 4.2) helps explain black-box machine learning techniques.

4.6 Discussion

The usage examples above demonstrate that the proposed approach can help to explore dimensions, evaluate clusters, and identify outliers in multidimensional projections. To complete the analysis, we further assess its applicability to practical scenarios in this section. Therefore, we conducted informal interviews with domain scientists to prove accessibility and tested the scaling behavior of our algorithm. We close with a discussion of limitations as well as potential improvements.

4.6.1 Application Study

The goals that we set out to accomplish were as follows: Design a technique and system (i) that is easy to use and comprehend, (ii) that is applicable to all types of embedding techniques, (iii) that can be directly integrated into existing analysis pipelines, and most importantly (iv) that enables the user to quickly and correctly understand attribute value distributions in the embedded data.

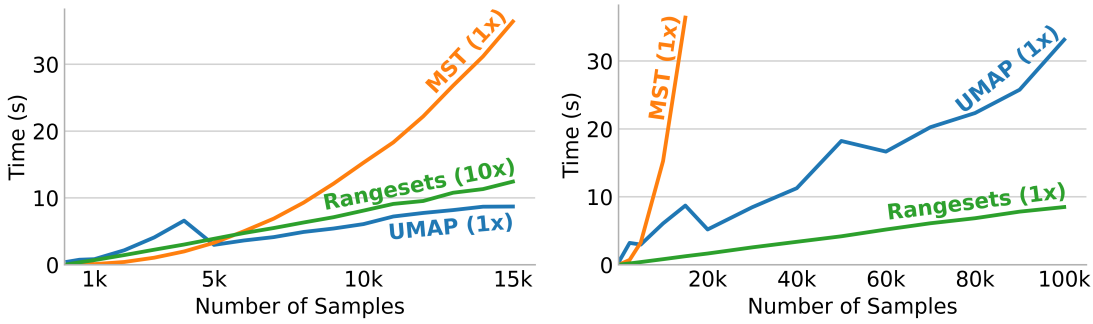
Goals (i + iv) are demonstrated in the use cases and were assessed in an informal expert user study with five domain scientists from various application fields that need to interpret high-dimensional data. We demonstrated NoLiES and gave them access to the notebooks. They all used data from their own work and explored structure and outliers therein (one application is reported in sect. 4.5.3). All experts commented directly that the visualization is visually appealing and easy to comprehend. One user commented that rangesets reminded him of cartography, which we deem an interesting analogy. The interactive GUI part proved directly accessible to all levels of computer literacy. The users with background in programming/python were additionally able to download NoLiES from GitHub and customize the notebook for the exploration of their own data.

Goals (ii + iii) are ensured by the implementation in Python and Jupyter Notebooks and the lack of coupling to the embedding. We demonstrate rangesets for MDS and t-SNE, which are often used to reveal inherent structure and clusters in the data. We also tested rangesets on PCA and SVM projections with similar results. Rangesets are computed in a post-processing step and the code can be easily used outside NoLiES. In the time since first publication of the tool, both the whole workflow and the stand-alone rangeset module have been successfully applied by various domain experts and students, validating the accessibility of the implementation.

4.6.2 Scalability

Algorithms from algebraic topology have great analytical power, but are known to be computationally expensive [212]. To assess the scaling for larger datasets, we conducted a systematic runtime analysis. We downsampled the cover type dataset [209], which consists of 500k data points, to set sizes between 100 and 100k. To account for variations in data and disk activity, we averaged the results of five measurement runs.

Our approach requires multiple runtime-relevant steps: the creation of the underlying embedding, a minimum spanning tree computation for assessing the default ϵ and the projection quality [204], and the rangeset computation itself. The embedding is exemplary created with the UMAP algorithm [36], which remained reasonably fast for all sample sizes at a maximum of 30 seconds as it is only run once, cf. Figure 4.14. Computing the minimum spanning tree with the python packages *scipy* and *scikit-learn* scaled near quadratically and was prohibitive beyond 15k data points, but can be avoided by manually setting ϵ and dropping the projection quality check. To analyze the rangesets, we constructed two use cases. We expect small datasets with up to 15k points to be analyzed in a small multiples setting with 10 attributes and 5 bins each, cf. Figure 4.14 (a). For larger datasets where plotting alone is already computation intensive, we expect the user to analyze one attribute at a time, cf. Figure 4.14 (b). Overall the rangeset computation scaled linearly with the number of data points. With maximum times of 12.5s and 8.5s in the respective scenarios, we found rangeset computation to be fast enough to work on data sizes below 100k.



(a) Small datasets

(b) Large datasets

Figure 4.14: Scalability study subsampled from the cover type dataset on the relevant algorithmic steps: embedding (UMAP), ϵ-estimation (minimum spanning tree MST), contours (rangesets). MST-Computation is infeasible after 15K samples. Rangesets scale linearly with the number of data points even on large data counts. (a) Small multiples analysis for 10 attributes with 5 bins each. (b) Single attribute analysis with 5 bins.

4.6.3 Limitations and Future Work

Limitations that we encountered relate mainly to scalability issues. The analysis in the previous section affirmed that with increasing number of data points interactivity slows down. Computing contours in the forest cover type dataset with 4.2k data points takes about the same time as the computation of the embedding on a regular desktop PC. For both routines, we use external libraries that we cannot easily improve. As rangesets are recomputed individually and only on slider changes, we found the latency acceptable. Regarding rangesets shown in one plot, discerning all of them became challenging with 20 classes in the thermodynamics example, as the current use a bokeh default colormap starts repeating hues. Colormaps optimized for perception that are aware of spatial overlap may further increase visibility [37]. Regarding the number of attributes, we show in Figure 4.10 that NoLiES can successfully be used for up to 11 attributes. With sufficient screen space the small multiples can be extended to show at least 16 attributes simultaneously. The geometrical nature of rangesets directly suggests several extensions like the support of Boolean operations on the sets. This concept could also be applied for a further automation of the analysis process of cluster properties, which we currently did fully manually.

4.7 Conclusion

In this chapter, we presented NoLiES, an interactive system for the interpretation of embeddings of multidimensional data projections. We introduced rangesets, an augmentation strategy for embeddings that outline data points with similar values in multiple non-convex contours. Rangesets have a dedicated handling of outliers and their only parameter is the maximal acceptable edge length between connected points. We discussed the relationship between rangesets and algebraic topology and demonstrated how the theory can be used to control the rangeset parameter. To work with rangesets of multiple data attributes, NoLiES integrates an interactive small multiples concept that is linked by selections and color coding. Important knowledge obtained during the analysis of a dataset can be stored in the notebook and used in future analysis.

Parts of this chapter have been previously published in:

J.-T. Sohns, M. Schmitt, F. Jirasek, H. Hasse, and H. Leitte. “Attribute-based Explanation of Non-Linear Embeddings of High-Dimensional Data”. *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 540–550. DOI: 10.1109/TVCG.2021.3114870

Chapter 5

Hierarchical Evaluation of Scalar Prediction Matrices

The analysis of matrix data is essential in domains like graph theory, biology, and engineering, i.e., disciplines that increasingly rely on machine learning [48]. Heatmaps are a central inspection tool in these domains as they directly visualize scalar matrix data without the need for abstractions. Patterns evident in the heatmaps offer data-driven understanding of the interrelation between row and column elements. As the observable patterns are inseparably dependent on the ordering, we describe a workflow for pattern-focused assessment of scalar matrix orderings on the example of model predictions for thermodynamic mixture properties.

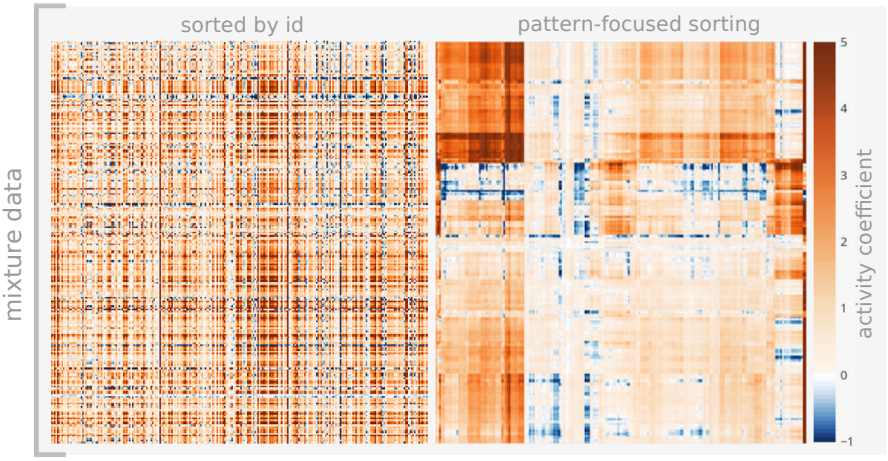
Real-world matrix data is usually not constrained to only the matrix elements, but each row or column can further be described by features differing in data type or scale from the matrix elements. Contrasting the matrix elements with these additional feature values is called enrichment analysis and can provide crucial insights about the quality or relationships within or between matrix patterns. While extensive support for general heatmap visualization exists through domain specific tools [41] and common plotting libraries, the support for enrichment analysis is often limited to the application case. The enrichment is either too domain-specific [39] or supports only minimal data types [41]. As the sole purpose of enrichment is to compare it to patterns in the matrix, the focus should be on matching the two datasets to judge the bidirectional quality of identified patterns. We therefore present an interactive software that links matrix patterns to common enrichment data types as well as propose validation measures to guide the analysis.

5.1 Application Background

Modeling and predicting thermodynamic properties of mixtures is of paramount importance in chemical engineering. Their knowledge is the basis for design and optimization of processes in the chemical, pharmaceutical, and biotechnological industry. The gold standard for obtaining these properties are experiments, which are, however, very

expensive and time-consuming; it is impossible to study all mixtures of interest experimentally. Therefore, prediction methods for thermodynamic properties of mixtures are of paramount importance. The vast majority of prediction methods rely on features of the pure substances that make up the mixtures [213], e.g., their composition with regard to structural groups as in so-called group-contribution methods [214]. A better understanding of the relationship between properties of pure substances and mixture properties holds the potential of significantly improving present and future prediction methods, with a direct impact on process design and optimization in chemical engineering. With our software, we address this challenge by analyzing the activity coefficient of binary mixtures, a measure that describes the deviation from ideal mixture behavior. While the software is currently visually slightly specialized, it is applicable to any other domain with similar data types.

The data considered for the analysis falls into two categories: a scalar property of substance *mixtures*, i.e. the heatmap, and properties of *pure* substances, the enrichment.



substance data

id	Name	Dipole Mom.	Mol.Weight	...	Tm [K]	Tc [K]	Single-Label	Multi-Label
1	Acetaldehyde	2.8068	44.0530	...	150.20	461.00	AL_ws	npol_short,al,keto
3	Acetonitrile	4.0013	41.0526	...	229.30	548.00	CN_ws	npol_short,pol_mono,cn
4	Acetone	2.9948	58.0800	...	178.35	508.10	ON_ws	npol_short,on,keto
7	Ethyl bromide	2.2150	108.9660	...	154.55	503.80	XALK_br	X_alk,Br_alk,X,Br,npol_short,pol_mono
9	Ethyl iodide	2.0665	155.9660	...	164.65	565.00	XALK_i	X_alk,I_alk,X,I,npol_short,pol_mono
...	...	...	...	...	...	...	...	...
1595	Chrysene	0.0005	228.2930	...	528.15	994.20	C_ARO_cond	aro,c_aro,cyc,six,cond
1700	Hexanenitrile	4.3811	97.1601	...	193.30	622.05	CN_wis	npol_mid,pol_mono,cn
1878	1-Phenyldodecane	0.4831	246.4360	...	266.15	779.10	AN_long	_CHn,npol_long,aro,c_aro,cyc,six
1917	n-Butylcyclohexane	0.0685	140.2690	...	198.45	667.00	AN_cyc	_CHn,an,npol_long,cyc,six
1995	trans-2-Pentene	0.0470	70.1340	...	132.95	475.00	EN_lin	en,npol_mid

Figure 5.1: Input data: Each row and column in the heatmap represents the mixture properties of a substance that has additional properties shown in the bottom table. (top left) A heatmap sorted by substance ID does not permit pattern analysis. (top right) The same data sorted by row and column similarity reveals patterns in mixtures that shall be related to pure substance properties.

The data for binary *mixtures* can conveniently be arranged in matrices. The heatmaps in Figure 5.1 (top) represent one such thermodynamic property of binary mixtures, namely the activity coefficient of a solute (row) at infinite dilution in a solvent (column). The left heatmap is sorted by substance IDs—for comparison, the right one is sorted to show visible patterns. Uniform regions in the matrix indicate that the respective solutes (solvents) are similar with regard to the activity coefficients.

Properties of *pure* substances as represented in the table in Figure 5.1 (bottom) enrich each row/column of the heatmap. We call them *substance features* and distinguish into two types here: (1) rigorous properties, which are measurable or unambiguously deducible from the molecular structure of the substance, and (2) more indistinct descriptors, which are defined based on experience or chemical intuition. As properties of type (1), we consider dipole moment, molar weight, polarizability, anisotropy, normalized anisotropy, relative number of H-bond acceptors, relative number of H-bond donors, normal melting temperature (Tm), and critical temperature (Tc), which are all of numerical type. As (rather subjective) descriptors of type (2), we consider the affiliation of a substance to a chemical class, like *branched alkanes*, *cyclic alkanes*, or *heteroaromatics*, which are single-label classes, as well as attributes defined based on the molecular structure, like *cyclic*, *aromatic*, or *long-chained*, which are multi-label classes.

The activity coefficient is essential for chemical process engineering in practice, but was, thus far, hard to predict precisely. Due to recent advances of successfully applying matrix completion methods [21], the data was expanded to more than 50k binary mixtures, 234 solutes and 214 solvents. This novel approach generates mixture data from other mixtures rather than relying on substance-driven methods [214]. Hence, it enables unbiased comparison between mixture and enriched substance properties. Correlation with measurable properties like dipole moment may provide guidance for future prediction methods. Moreover, aligning mixture groups with established class labels like heteroaromatics can assess their suitability for prediction techniques.

5.2 Requirement Analysis

This chapter stems from a collaboration with the Laboratory of Engineering Thermodynamics at the University of Kaiserslautern-Landau. The starting point of the project was the heatmap presented in Figure 5.1 (top-left), which represents the activity coefficients of 234 solutes at infinite dilution in 214 solvents at a temperature of 298 K, i.e., for more than 50k different binary mixtures. 8% (4k) of the data are experimental values taken

from the Dortmund Data Bank [215], while the remaining 92% are predictions of one of the best available methods for this purpose [21]. The activity coefficient is a central thermodynamic property that is commonly used for describing the non-ideality of liquid mixtures [20]. Modeling activity coefficients is at the core of many simulation tasks in chemical engineering, which in general requires understanding the relation to descriptors of the respective pure components. The goal of chemists and chemical engineers is to understand the structure in this data and relate it to pure component descriptors.

We loosely followed the design study methodology proposed by Sedlmair et al. [216]. In the *discovery phase*, the groups exchanged necessary domain knowledge and collected the data. A central pillar of the *design phase* were continually improved prototypes that were build using accessible visualization tools like seaborn [217] and holoviews [218], which include necessary visualization and data aggregation tools like clustermap [217], i.e., heatmap visualization with associated cluster trees, and statistical aggregates like grouped histograms. These early prototypes built a common ground to communicate, explore shortcomings, and refine the demands of the domain experts.

During the joint discussions of these prototypes, we made the following design observations: Central features are *access to raw data and use of established chart types*. Our work centers around pattern mining in complex data. Using chart types that are familiar to the user and represent the raw data makes it easier to judge the effect of the automatic analysis routine. Therefore, we choose a clustermap as the central plot. *Flexibility of the software* is another important aspect. During the progressive data analysis, we realized that the data basis is not fixed. New insights may require the integration of new substance properties. Hence, a flexible design is necessary that can adapt to arbitrary numbers of descriptors and new data types. *Linked interactive filtering* in multiple/all directions discloses relationships between views. In each prototype iteration, the users intuitively tried to experiment with linked selections first. Thus, we made interaction and linking central components.

The main challenge in mixture prediction is that most methods rely on implicit *knowledge of experts*. With today's rising availability of data, problem-driven visualizations should move the information location towards *data*- and therefore *computer*-driven approaches [216]. Thus, the central goal of our work is finding data-driven patterns in mixture data and linking them to properties of pure substances. Likewise, we are also interested in breaks in expected patterns, i.e., if substances that belong, according to chemical intuition, to the same chemical class show few similarities. In a nutshell, we aim to understand what similarity among substances actually means, but with regard to

their behavior in mixtures. Up to now, no software is available in this field to answer these questions and current research in thermodynamics basically depends on manual work of experienced physical chemists.

5.3 Related Work

Research directions on visual pattern analysis in matrices are threefold: Patterns are either defined within the matrix, on the tree that constitutes an ordering, or the distribution of annotated attributes in the ordered matrix.

Matrix Patterns For observable patterns in symmetric binary matrices, Behrisch et al. provide a comprehensive analysis in Magnostics [100]. However, they state that the defined patterns are specific to symmetric matrices, which are commonly sparse and binary, and do not generalize to data tables [49]. Wilkinson [219] describes canonical data patterns observable in general heatmaps – asymmetric scalar matrices. For patterns found in real applications, we argue in Section 5.4.1 that Lekschas et al. [220] present a more suitable description of patterns, although their approach is focusing on small recurring motifs in symmetric scalar matrices. Their approach finds patterns in large matrices, focusing on small recurring motifs, which is suitable for sparse networks, not dense table data like our case. Here, Chen [221] argues that aggregating heatmaps of correlation matrices into rectangular partitions is sufficient for interpretation. He later extended this idea to general heatmaps only showing the mean of each block [222].

Tree Patterns To visualize patterns in a tree, Parthl et al. [223] classify four options to visualize attributes: directly on the node; small-multiples of the graph; linked views of graph and attributes; or adaption of graph layout. Small-multiples work for comparing attributes in graphs [224], but do not match with the strict order of a matrix. Degree-of-Interest trees aggregate the tree-layout depending on a function of interest [225]. While initially interest was defined over interaction with a node, Lineage [40] extends this idea to attributes in genealogy. They propose several strategies for a binary Degree-of-Interest, which we extend to a continuous measure of node variation. Chen et al. [226] show that the interactive exploration of matrices over a dendrogram provides insight. Combining the benefits of Lineage [40] for on-node mapping and the linked views of GAP [58] and GUIRO [56], we derive our design for annotating external attributes in Section 5.5.

Enrichment Patterns Heatmap literature typically allocates minimal effort to enriching heatmaps with external attributes. Notable exceptions include VIS-STAMP [227], incorporating linked parallel coordinates and a map for context and filtering, and Lex et al.'s genealogy-specific system [39], which links domain-specific views to a sorted 2.5D heatmap. Clustergrammer [41] stands out as a recent, well-implemented paradigm for heatmap plotting, adding color-coded columns for categorical features. Additional data is accessible via hyperlinks to open databases. In HiPiler [220] individual matrix snippets are enriched via border colors, though this does not transfer well to many categories or continuous distributions. Since our annotated data drastically exceeds the number of distinguishable class colors or is continuous, we opt for aggregating label variation and visualizing variable distributions rather than directly relying on color-coded attribute values.

5.4 Method

To answer the questions described in Section 5.2, we propose a three-step workflow. We start with an overview of practically observable patterns in scalar matrices in Section 5.4.1 and continue with a block-focused assessment of ordering techniques in Section 5.4.2. We close with augmentation strategies for validating these patterns with external data in Section 5.4.3. Subsequently, we discuss the design of suitable supporting plots to identify relationships with enrichment data in Section 5.5.

5.4.1 Matrix Patterns in Scalar Asymmetric Matrices

The solute-to-solvent ratio in mixtures is generally not interchangeable, e.g., you would not solve water in salt. Hence, there is a one-way relationship between each pair of solute and solvent and the corresponding matrix is inherently asymmetric. From the established matrix patterns summarized in Figure 5.2, we therefore deduce the ones that are applicable to our asymmetric practical data.

We start with established patterns that do not transfer to our use case. Simple and Equi patterns [219] are assuming a uniform global correlation unlikely to be found in real data. Similarly, the Band/Circumplex and Bandwidth patterns [219, 49] are only relevant if the diagonal holds meaningful information, i.e. for symmetric matrices. Loops [220] are only sensible in adjacency matrices.

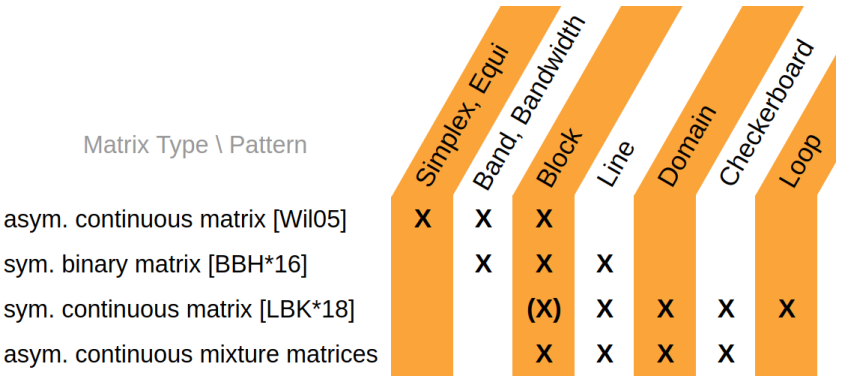


Figure 5.2: The observable patterns depend on the type of matrix (rows). The columns mark which patterns have been (indirectly) described for each type.

Block The most common pattern throughout applied literature is the block pattern, where coherent rectangular areas appear in the sorted matrix. A block can be described by a range of row- and column-IDs and a corresponding scalar value. Due to noise in real data, the block area is commonly not as uniform as in the synthetic example in Figure 5.3 (top-left). A block denotes that a number of entities (set of rows) share similar values in a number of features (set of columns). For binary mixtures, this pattern denotes that the related set of solutes exhibits similar activity coefficients in any solvent from the respective set, i.e. they form a common class.

Line The line pattern is a special variant of the block pattern. Here, a single row or column features extremal values that set the data points apart. Multiple similar entities/features can exist. If ordered accordingly, they will form a long narrow block spanning multiple rows/columns. Lines are entities that are highly dissimilar from any other data point therefore indicating outliers. In binary mixtures, water plays such a special role as it leads to extreme activity coefficients if mixed with different components as can be seen in Figure 5.3 (right).

Domain In a domain, a larger block region contains additional sub-blocks where the color is darker or lighter, as represented in Figure 5.3. Like blocks, domains can be characterized by their respective row and column IDs. The nesting structure requires a hierarchical description model. We found cluster trees suitable to describe the nesting structure and the distinctness of sub-blocks. From a chemical-engineering perspective, we can interpret this type of pattern as a rather large group of solutes and solvents that mix similarly with another group, while there are subgroups that are even more similar.

Checkerboard Checkerboard patterns are an arrangement of globally alternating blocks of high and low values. While they are a common phenomenon in gene expression

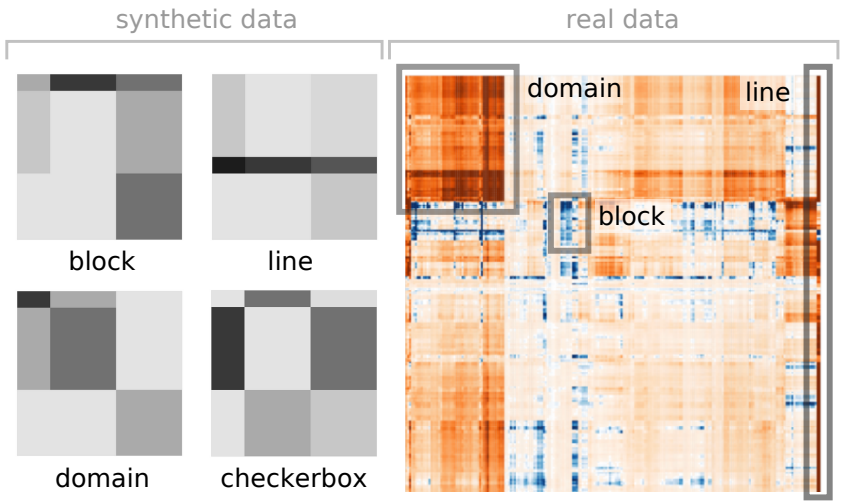


Figure 5.3: Patterns in asymmetric matrices: (left) Four patterns are identified in asymmetric matrices. (right) In real data, the patterns feature various degrees of expressiveness.

data, they are barely visible in mixture data, indicating groups of high intra- but low inter-activity.

We conclude that the occurring patterns can all be described as (hierarchical) blocks, which have been shown to be sufficient for heatmap interpretation [221, 222] and are the most common in practical data [41, 228]. For this approximation to apply, Chen [221] outlines three pre-conditions: (1) appropriately ordered samples; (2) carefully derived partitions; and (3) representative summary statistics. We adhere to these principles by evaluating the ordering and partitions in Section 5.4.2 and validating patterns with domain knowledge statistics in Section 5.4.3.

Matrix Reordering Algorithms As manual reordering is too tedious for real datasets, we resort to a choice of reordering algorithms. In Behrisch et al.'s [49] extensive evaluation of available algorithms focused on block-diagonal patterns, they conclude that hierarchical clustering, specifically optimal leaf-ordering [229], excels at producing local patterns. That is even though hierarchical clustering is intended to cluster, not to induce a global linear order on matrix rows [48]. Recent studies on continuous matrices also suggest that Robinsonian techniques (i.e., those based on similarity matrix ordering [49]) and machine learning techniques are best suited for detecting block patterns [230]. Since we further aim to capture hierarchical domain patterns, we choose hierarchical clustering with optimal leaf-ordering – a Robinsonian technique that is long-established in biological contexts [231, 232, 233, 58].

5.4.2 Evaluation of Blocks in Matrices

Hierarchical clustering has two parameters: a metric that defines the distance between two rows/columns, and a linkage type that defines the distance between two clusters. The choice of these parameters is crucial for the clustering result and therefore the matrix ordering. While all common combinations can be explored in our tool, we describe a generally applicable workflow to compare hierarchical clusterings regarding their ability to uncover block patterns.

The strength of a block pattern can be quantified over its uniformity. As a well-established and therefore readily understood measure of uniformity in a dataset, we use the standard deviation of values within a block. Alternative choices with similar results are mean square error or mean absolute error. The two dendrograms defined by hierarchically clustering rows and columns can be pruned at any point, partitioning the matrix into blocks. We express the quality of such a partitioning by the average uniformity of each individual block. To account for the relative importance of each block, we weight the score of each block with the number of contained elements, then average over all blocks. Since we perform separate clusterings on each of the axis, separating into 1 to n clusters per axis leads to n^2 numbers of possibilities for block layouts. We assume that a matrix ordering that forms more distinct block patterns with the same number of blocks is preferable. The appropriate number of clusters is, however, typically not known beforehand.

To find the default parameter setting for our tool, we analyzed the induced matrix orderings for all combinations of common distance measures (Euclidean, Manhattan, cosine) and linkage parameters. Single, average, and complete linkage minimize the shortest, average, and longest distances between any two points in a cluster, respectively. In Figure 5.4 (top) we present three charts for Euclidean distance with different linkage strategies. The x- and y-axis denote the number of clusters, and the color-value indicates the quality score for this partitioning. The standard deviations for single linkage are consistently higher than those for complete- and average linkage.

We note that the standard deviations decrease along a path from bottom-left (single block) to top-right (finest granularity). To ease comparability, we reduce the heatmap to a line in Figure 5.4 (center). Therefore, we choose a suitable path through the heatmap that captures the fastest decline in standard deviation. In our experiments, a suitable path occurred along the diagonal, though the path could be shifted or bent for more asymmetric data. We observe that complete linkage results in consistently the lowest

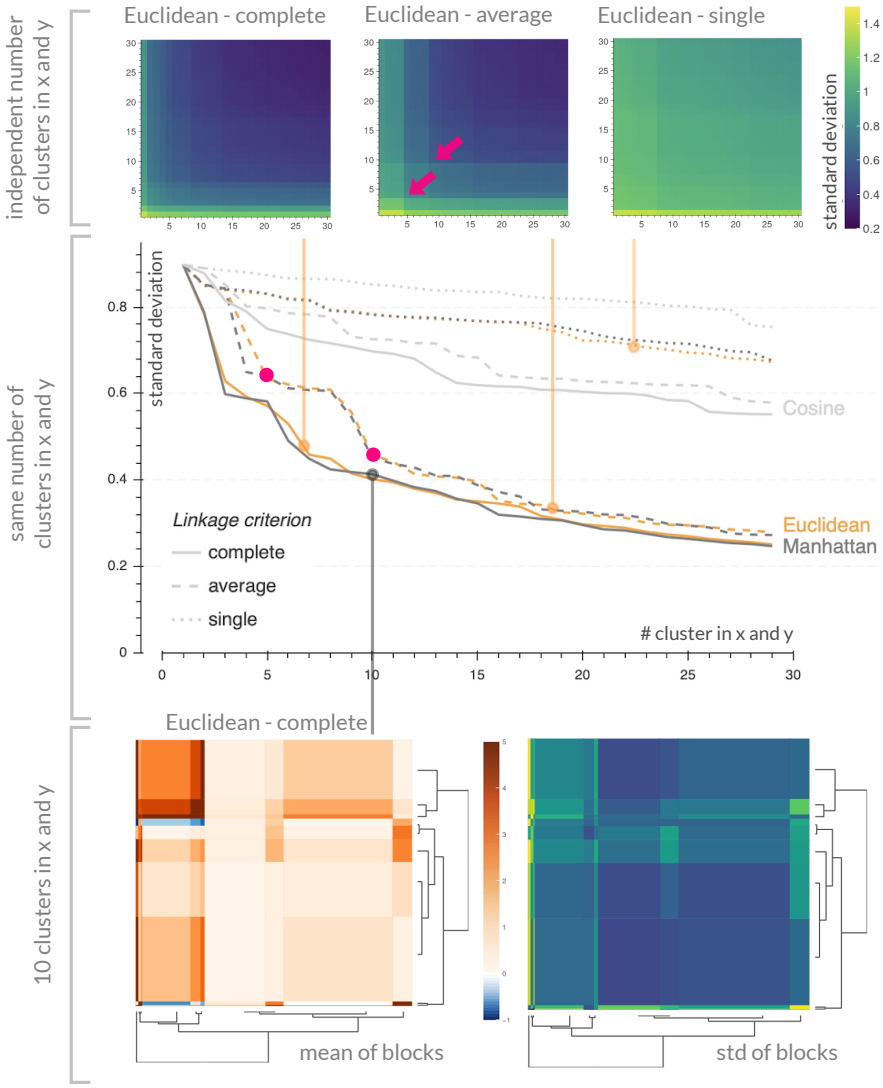


Figure 5.4: Effects of clustering parameters: (top) Comparison of linkage criteria for arbitrary combinations of cluster granularity in x- and y-direction. (center) Comparison of clustering parameters for same granularity in x and y. (bottom) Effects for division into 10 clusters in x- and y-direction.

standard deviations for all three metrics. Average linkage contains characteristic drops (marked in pink) that are also visible in the 2D plot, which indicate strong local improvements in block quality. Euclidean and Manhattan distance showed equally low scores in our tests. We chose Euclidean(-complete) as our default, as it is the most common. To verify our result, we search for the "elbow" in the line for Euclidean-complete (solid orange), which gives us the Pareto optimum, the best trade-off between minimized number of clusters and low error rates. We find it at approx. 7–10 clusters. The partitioned matrix with 10 clusters each is shown in Figure 5.4 (bottom). Coloring the blocks by their

mean (left) partitions the heatmap into regions of predominantly high or low activity coefficients. The distribution of the standard deviation (right) provides insight into the error rates within each block.

Note that it is crucial to keep the variation comparable across rows and columns, i.e., the matrix data needs to be standardized. In our application case, we already had the same measure for all data points and scaled it logarithmically to achieve perceptually equidistant changes across the full value range.

5.4.3 Validating Patterns using Domain Knowledge Variation

Finally, we want to guide the interpretation of patterns based on enriched substance properties; among others, we thereby want to identify the subset of properties that characterizes the substances (rows and columns) the best with regard to their mixture behavior (matrix entries). Clusters where domain knowledge matches with the similarity in the matrix signal a correlation, which is usually considered interesting [41, 234]. On the other hand, clusters that do not match with domain knowledge are even more interesting, since they can spark new ideas for undiscovered relationships. A clustering is considered to be matching the domain knowledge, if both approaches group the same annotated elements together. Hence, we consider the variation of associated domain knowledge within a cluster as its validation score. Clusters that have low variation in associated domain knowledge are considered pure, i.e., they get a low score. Clusters with high variability in domain knowledge receive a high score. This concept is illustrated in Figure 5.5, where the inner nodes of the dendrograms are colored based on the validation score of user-selected substance descriptors. Hover and selection interaction then provide detailed information and will be discussed in detail in Section 5.5.

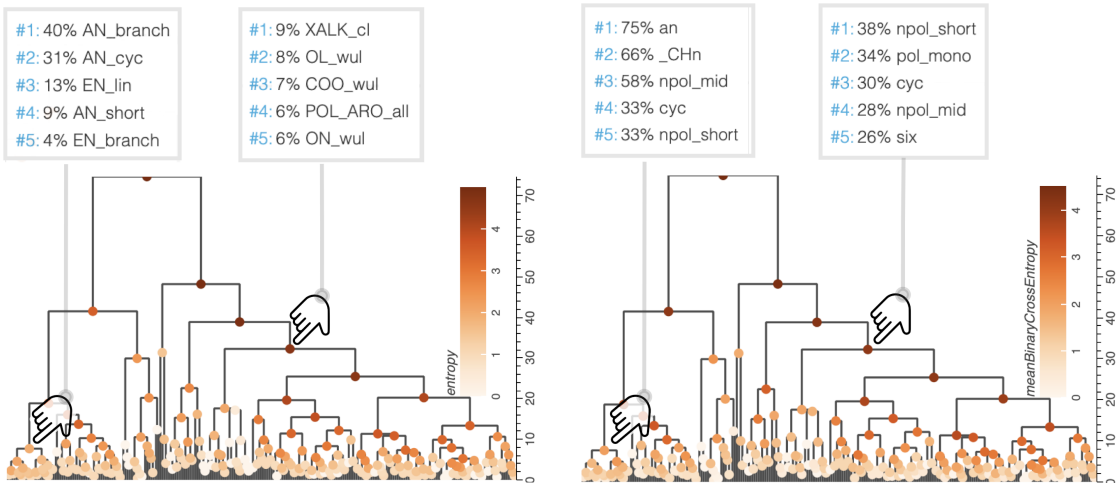
As we have seen in Section 5.2, substance descriptors come in three data types: *single-label* (chemical class), *multi-label* (composition with regard to functional groups), and *numerical* (measurable and deducible quantities). This also covers most of the data-types potentially occurring in other application domains. The direct color-coding of descriptors is limited by available colors and screen space [41, 234]. Hence, we recommend dedicated scalar measures for these three types of data. While the measures itself are well-established in information theory, their application to defining agreement between clustering and domain knowledge has not been proposed before to the best of our knowledge.

For the **single-label** case, we suggest *entropy*:

$$e_{\text{single}} = -\sum_{i=1}^{n} p_i * \log_2(p_i) \tag{5.1}$$

where p_i is the probability of label i in a given cluster. Entropy is maximal for uniformly distributed data and increases with increasing numbers of elements, which is what we expect for our application. Alternative information theoretic measures [235] would work, but e.g. purity considers only the biggest class and not the full distribution within a cluster. Other measures, such as pair-counting and set-matching [235], rely on ground truth labels, which are typically not available in exploration workflows. Figure 5.5 (a) shows the entropy for the single-class labels, which were manually assigned based on chemical intuition (a total of 41 labels, e.g., *cyclic alkanes*, *water-soluble alcohols*, etc. were considered). Hovering over the dendrogram nodes shows a tooltip with the most frequent labels. For the left tooltip, two labels make up 71% of the classes, which results in low entropy. The right tooltip shows a cluster with many different substances and, accordingly, high entropy.

For **multi-label** assignments, the assumption of entropy – each entity has exactly one label, hence the probabilities sum to 1 – does no longer hold. Further, the measure should evaluate equally for uniform distributions of different sizes. We use a measure from machine learning, namely *binary cross entropy*, which is commonly used as a loss function



(a) Single-label descriptor (b) Multi-label descriptor

Figure 5.5: Interactive dendrograms for cluster analysis and refinement: Each dendrogram is color-coded with a dedicated measure for a different data type. Dark colors indicate nodes with high variation. A dedicated interaction tool provides detailed information about the respective descriptors.

to quantify how well a predicted multi-labeling approximates the ground truth. In our scenario, we estimate how far our clusters are from optimal uniform multi-labelling:

$$\begin{aligned} e_{\text{multi}} &= \frac{1}{N}\sum_{i=1}^{N}((1-y_i)*\log_2(1-p_i)) - (y_i*\log_2 p_i) \\ &\overset{y=1}{=} -\frac{1}{N}\sum_{i=1}^{N}\log_2(p_i) \end{aligned} \tag{5.2}$$

For all labels y in a cluster, $y_i = 1$, if the label is correctly predicted to be in the cluster and $y_i = 0$, if it is falsely predicted. p_i is the fraction of elements with this label in the cluster. Since we again lack the ground truth necessary for cluster validation, we assume the ideal case is that clusters are pure, i.e., all labels N occurring in a cluster are present in all elements of a cluster $y_n = 1, \forall n \in N$. With this assumption, the formula simplifies significantly. The metric then indicates the degree of deviation from the ideal case of a pure cluster. Figure 5.5 (b) shows the binary cross entropy for a set of structural attributes characterizing the substances (e.g., *cyclic*, *aromatic*, *long-chained*); the set of structural attributes was defined manually here, but any categorical multi-labeling, like the well-established group-contribution method UNIFAC [236], could be used. The tooltip again shows the most frequent labels and denotes how many substances share a label.

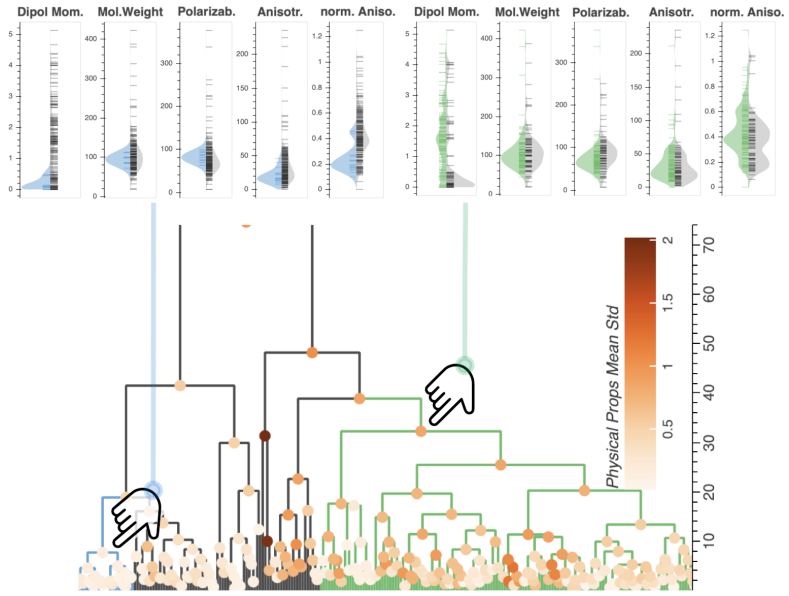


Figure 5.6: Interactive dendrogram color-coded by variation within continuous numerical descriptors. Selection contrasts the selected cluster from the rest of the data.

For **numerical data**, we chose the *mean standard deviation of standardized values* σ_{mean}, which means we compute an average standard deviation over all included continuous measures:

$$\sigma_{\text{mean}} = \frac{1}{|V|}\sum_{v=1}^{|V|}\sigma_v \quad \text{with}$$
$$\sigma_v = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(z_{v,i} - \overline{z_v})^2} \quad \text{and} \quad z_{v,i} = \sum_{i=1}^{m}\frac{x_{v,i} - \overline{x_v}}{\sigma_v} \tag{5.3}$$

In a first step, we make variables comparable by standardizing each variable individually based on their distribution in the full dataset m. We then determine the variation within a cluster by computing the individual standard deviation σ_v for each variable restricted to the values of the cluster n. To make the measure independent of the number of variables, we output the mean of the standard deviations over all variables V. We chose this formula, since standardization of features will be necessary in almost all application scenarios, and it indicates directly how the standard deviation within the cluster compares to the global one. Figure 5.6 shows σ_{mean} for five continuous descriptors. Selecting a node in the tree shows violin plots for the numerical descriptors, contrasting the selected cluster (colored-coded in the tree and the violin) with the rest of the data.

5.5 System Design

Using the substance feature variation, the user can now manually traverse the tree and search for clusters with a semantic meaning. For their interpretation, they need detailed information about the substance features, which we provide in interactively linked widgets. The interface is deduced from the previously compiled design goals: Hover and explicit plots offer *access to raw data*; enrichment plots scalable with regard to number of features and data type give *flexibility*; and *linked interaction* provides intuitive exploration.

Figure 5.7 shows the entire GUI. The collapsible parameter sidebar on the left covers algorithmic and style settings that can be controlled by the user. The visualization section on the right contains multiple linked views on the data with interaction capabilities. We have already discussed the design of the clustermap. For the enrichment substance features, we provide four additional visualizations. (1) A table containing the names and molecular formulas of the substances, (2) a 2D projection of matrix rows or columns to reveal relative distances, (3) an extension matrix plot showing multi-labels, e.g. the

Figure 5.7: Exploring Mixture Data: A sorted heatmap visually groups blocks of similar chemical substances (rows and columns). Pattern strength is analyzed by variation in internal and external data. Linked widgets connect the discovered groups to additional domain knowledge. A higher-resolution version of this figure is available in the Appendix.

composition of the molecules with regard to structural groups, and (4) violin plots to analyze distributions of numerical features. We thereby extend previous enrichment support [41] to more and potentially continuous variables, independent of application domain.

All widgets are interactively linked. Selecting a substance or substance group in one of the widgets triggers a highlighting operation in all the other ones, supporting the user in finding relevant features. Individual data point selections are drawn with bigger lines and bright orange color. In Figure 5.7 two high-level nodes were selected subsequently. The respective subtrees are color-coded blue and red. The other views adjust to the selection supporting the user in finding relevant descriptors. The table only shows currently selected substances. As kernel density estimates have been proven to work for cluster attribute comparison [237], the violin plots contrast the current selection against the remaining data or a previously saved selections. We combine violin plots with hoverable rug plots to ease the reading of outliers and to provide interaction capabilities, e.g. selection of individual substances. In case a user is uncomfortable with violin plots, they can change to equally arranged histograms. Even if carefully chosen, the matrix imposes a linear ordering that cannot capture the potentially complex neighborhood relation-

ships between rows/columns. We include a non-linear projection that mirrors the colors of user selections. We chose MDS over other non-linear projections for its simplicity to explain to domain scientists. The hover tool provides the exact values and substance names of the glyphs in every view. A demonstration of all interactions is given in the accompanying video. The implementation is built in Python with Bokeh and Panel [146] as the charting and interaction libraries. The tool is available on GitHub[1].

5.6 Case Studies

From our analysis in Section 5.4.2, we know that we work on a suitable ordering. Therefore, we demonstrate how the software can be used to first find pattern-correlating numerical features and then confirm or question reference classifications.

5.6.1 Matrix Pattern Correlation with Continuous Features

Finding informative substance properties is crucial for the development of prediction methods for thermodynamic properties. To date, this task is based on intuition of human experts and, generally, by considering properties of *pure* substances. The software developed in this work facilitates an unbiased analysis based on *mixture* data, namely by studying which substance properties are particularly homogeneous in matrix clusters.

We notice two prevalent groups in the matrix, which correspond to the highest ranked nodes in the dendrogram. We select and save them with different colors in Figure 5.7. In an initial review of the substance table, we already observe a high degree of homogeneity among the substances. The blue cluster mainly contains non-polar hydrocarbons, whereas the red cluster includes highly polar compounds. The distributions of the substance properties in the violin plots, confirm our observation: we find rather small dipole moments and relatively high polarizabilities, a characteristic of non-polar molecules, in the blue cluster compared to the red cluster, while the red cluster exhibits greater heterogeneity. Hence, we conclude that the polarity of the molecules is one of the most important properties with regard to activity coefficients. While this agrees well with chemical intuition, polarity is usually integrated in the input of previous hand-crafted mixture prediction, so it is interesting to quantitatively find this in our unbiased *mixture* data. Thus, we can interactively analyze data-driven relationships between matrix patterns and multiple external feature distributions.

[1] https://github.com/Jan-To/EnrichMatrix

5.6.2 Transparent Evaluation of Reference Classifications

Classifying substances is fundamental for the development of predictive thermodynamic models, but a non-trivial task that is usually done by a human expert in a subjective manner. The software developed in this work enables a data-based evaluation of such classifications.

We change the color coding of the tree nodes to the occurrence of a specific class label as defined by an expert, e.g., *water-soluble nitriles* (CN_wl) in Figure 5.8 (center). In this mode, we notice that the *water-soluble nitriles* are part of the green and the purple clusters based on the mixture data. Taking into account that the two clusters contain distinct subsets of other classes, we conclude that *water-soluble nitriles* is not a very characteristic group label regarding the behavior of substances in mixtures. It probably should not be used for mixture prediction purposes. A reverse procedure is also conceivable: if we look for a cluster that is very homogeneous in its feature values, i.e., has a light node color, we quickly find the cluster in Figure 5.8 (right). Surprisingly, the tooltip reveals that it is quite heterogeneous according to its expert group labels. We find *branched*, *cyclic*, and *short alkanes* together with *linear* and *branched alkenes*, which apparently all show a very similar mixture behavior. Based on this observation, we can deduce that a separate consideration of these tagged groups is not pertinent and that

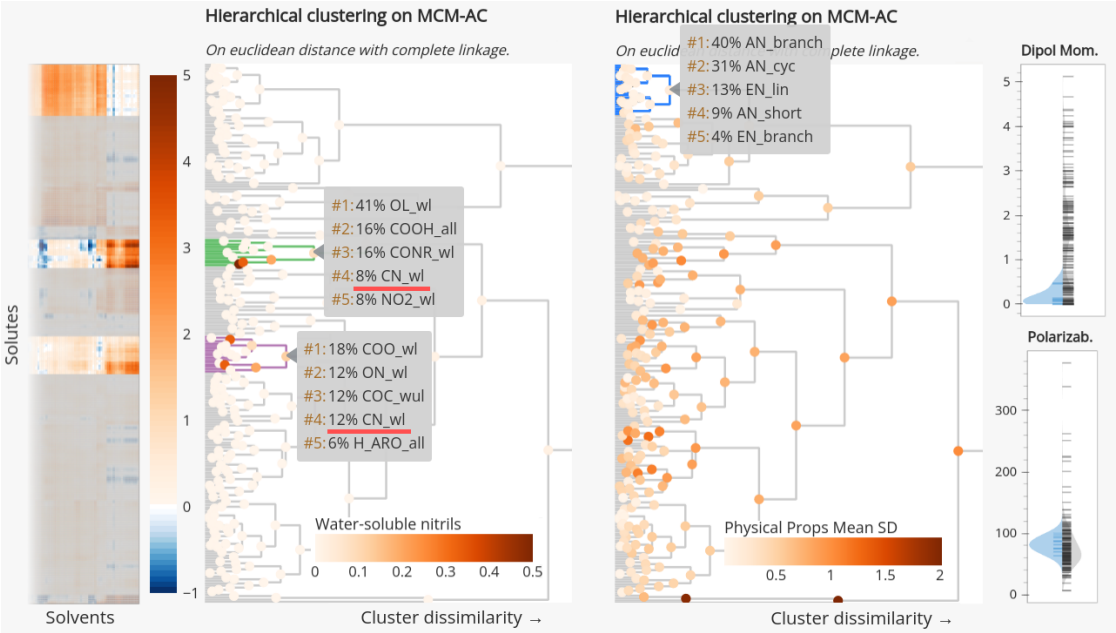


Figure 5.8: (Center): Water-soluble nitriles (CN_wl) can be found in two clusters, which indicates different mixture behaviors. (Right): Example for a cluster with high homogeneity regarding single-labels, which, however, includes multiple expert-labeled classes.

a classification should rather be a more general class or based on substance properties. Some properties, namely, dipole moment and polarizability, are very homogeneous in this cluster and therefore promising candidates. Through the multidirectional analysis between matrix patterns, class labels and feature values we enable domain scientists to validate, invalidate and suggest annotated matrix classifications.

5.6.3 Identification of Inconsistencies in Matrix Data

Binary mixtures whose components are similar to each other have similar activity coefficients at infinite dilution. As a consequence, components that are found close to each other within the dendrogram should have similar pure substance properties.

During the analysis of the dendrogram, we found a rather high-level cluster that contains all *perfluorinated alkanes* (FF_all) of our dataset, but besides also includes strongly polar substances with the property of forming hydrogen bonds, cf. Figure 5.9. This is surprising since *perfluorinated alkanes* exhibit unique properties, which massively reduce their

Figure 5.9: (Top) The blue cluster contains all perfluorinated alkanes (FF_all) and some other components. (Bottom) The other components within the cluster differ strongly from FF_all, i.e., they are highly polar and capable of forming strong hydrogen bonds, whereas FF_all are non-polar and do not form hydrogen bonds (pink marks).

miscibility with components of both polar and non-polar nature and make them distinct from other classes; this also becomes apparent when inspecting their properties in Figure 5.9 (bottom). Finding the *perfluorinated alkanes* in a cluster together with other classes of components is therefore strongly in conflict with the domain knowledge and should serve as an indicator for critically assessing the respective mixture data.

5.7 Discussion

In this section, we summarize the findings of our case studies and present a user evaluation. We reflect on the feedback received from users during the evaluation process, highlight the strengths and weaknesses of our approach and how they can inform future improvements to our tool.

5.7.1 Case Studies

The case studies demonstrate the effectiveness of our software in analyzing and interpreting mixture data. The first case study showed that the software can be used to identify informative descriptors for modeling mixture properties based on a cluster analysis of the mixture data. The second case study highlighted the ability of the software to validate and improve existing classifications of substances, providing valuable insights into the relationships between substance properties and mixture behavior. The third case study illustrated how the software can be used to identify inconsistencies matrix clusters, which can help researchers critically assess experimental mixture data and improve thermodynamic model development. In conclusion, our software gave hints towards descriptive or non-descriptive chemical classes and properties and therefore can serve as a first step to increase the quality of thermodynamic prediction models.

5.7.2 User Evaluation

In addition to the case studies by our domain scientist authors, we conducted a qualitative user study with six domain PhD students to evaluate the accessibility and effectiveness of our tool in practical use. The PhD students were not involved in the previous study and design process. All six are pursuing a degree in chemical engineering, albeit most do not specialize in the analysis of mixture data. Only one of them was familiar with the matrix data before the study, but has only seen numerical representations without annotations.

After a brief introduction to the data, the tool, and its functionality, we asked three introductory questions, after which they could explore the data on their own. We encouraged the participants to think aloud, took notes of their comments, and additionally recorded the audio during each session. In the end of the session, we conducted a short interview to capture a summary feedback on usability and productiveness. Each session took between 30 and 60 minutes. All participants started their analysis on the screen ordered as determined in Section 5.4.2 and were given the same tasks: *Can you find patterns in the matrix and characterize them with domain knowledge? Can you determine substance properties that significantly influence the mixture behavior? Which patterns coincide with domain knowledge?*

Most participants were able to immediately identify, explore and correlate patterns in the matrix. Selections in all plots were made abundantly. One person was overwhelmed with the abundance of simultaneous views and another felt unfamiliar with the concept of dendrograms, though both reservations resolved quickly and without intervention. The filtered table was equally used as the violin plots and checked against each other for reliability. The implications of the various node colorings in the dendrogram were extensively explored. The participants used them to recognize both consistent and inconsistent clusters and confirmed them within the violins.

The users particularly praised the wide range of consistent interaction and selection possibilities (4 users), the visual clarity (3 users), and the accessibility of complementary information (3 users). The participants commented on the beginner-friendly design through abstractions to color and violins, enabling analysis without knowledge of statistical methods (3 users), though the person familiar to the dataset liked that raw data is accessible by hovering at all times. Improvement suggestions were aimed to enhance the user experience. Participants were missing a “reset” button (3 users), images of the structural formula of substances (2 users), and resizing violin plots (2 users).

Overall, the positive feedback from our user study showed that application domains can greatly benefit from visual analytic interfaces compared to current workflows. Interactive linked views as well as visual abstractions are quick to learn and use as long as design and interactions are intuitive. With regard to our specific application, users were able to successfully check various levels of cluster validation (data, visible pattern, dendrogram, violin) against each other. The participants were so interested in the interaction possibilities that, even though most of them had no relationship with the data, everybody continued exploring after the official session ended. The participants were excited about the tool’s interaction possibilities and wanted to apply their own data, indicating

(1) that relating attributes to matrices is a common problem and (2) that the need for visual analytics software in application domains is still huge.

5.7.3 Limitations and Future Work

By handling multiple data types, the software is designed to be generic and lends itself directly to the integration with asymmetric scalar matrices from other domains. However, there are some limitations and areas for future work. Firstly, we rely on hierarchical clustering for its interpretability and inherent hierarchy for domain patterns, though alternative ordering techniques may be viable. Secondly, displaying all data points simultaneously limits the matrix size to $\sim$500x500 due to pixel resolution, but solutions like scrolling or agglomeration have been successfully employed before [226]. Nonetheless, our interface is currently designed to analyze (small groups of) identifiable elements. For massive, anonymous data, the fine-grained analysis of individual comparisons probably exceeds the necessary complexity. Thirdly, enrichment data must be available; however, in our experience with other engineering sciences, this data is widely accessible, and engineers are often eager to consolidate it. Hence, an apparent direction for future research is the application to new domains. Fourthly, the software is not yet fully integrated into the workflow of chemical engineers. The current version is a standalone tool that requires users to export data from their existing software and import it into our tool. Future work could focus on integrating our tool into existing software packages to streamline the workflow for users.

5.8 Conclusion

In this chapter, we introduced an analysis software for asymmetric scalar matrices that are complemented with meta-data for row and column entities. Central building blocks are a pattern-focused sorting of the heatmap and the guided variation analysis of the meta-data across multiple linked views. While the workflow and most parts of the software are independent of the application domain, we focus on machine-learning-assisted chemical engineering, particularly, the exploration of the predicted relationship between activity coefficients in binary mixtures and properties of the pure substances. We demonstrate that the software can be used in practice to find informative descriptors for modeling mixture properties based on a cluster analysis of the mixture data, and contrast it with existing domain knowledge. The analysis with our software provided data-driven

directions towards suitable substance descriptors and classification schemes for binary mixtures, advancing the field from manual analysis by physicochemical intuition. The user study proved that the software is easy to learn and gives interesting insights into the data. Additionally, we received valuable hints for future research.

Parts of this chapter have been previously published in:

J.-T. Sohns, D. Gond, F. Jirasek, H. Hasse, G.H. Weber, and H. Leitte. "Embedding Space Explanations of Learned Mixture Behavior". *Proc. 3rd Conf. on Phys. Modeling for Virtual Manufacturing Systems and Processes*. 2023, pp. 32–50. DOI: 10.1007/978-3-031-35779-4_3

Chapter 6

Additional Contributions

Beyond the core contributions presented in the main chapters of this dissertation, I have also worked actively on several additional projects during my PhD. These projects either apply the techniques developed in this dissertation to practical domains or reflect parallel investigations in the field of multivariate data visualization. Together, they extend and reinforce the overarching research goals of this work and provide a broader context for its impact.

6.1 Practical Applications

Application to Thermodynamic Modeling [4] This work demonstrates how the techniques presented in Chapter 3 and Chapter 4 can be applied in the field of thermodynamic modeling, including the extension to regression models. Specifically, it showcases a deeper analysis of a machine learning model predicting activity coefficients in binary mixtures, which are fundamental descriptors in chemical engineering. I was responsible for adapting the methods to the domain, conducting the analysis, and writing the manuscript.

Improved ML Model Informed by This Dissertation [6] Building on the hierarchical insights gained through the work presented in Chapter 5, this work introduces a hierarchical matrix completion approach for predicting properties of binary mixtures. The proposed method advances the state of the art in this application domain. I contributed to data curation, analysis, visualization, and provided input to the initial drafting of the manuscript.

6.2 Methods in Multivariate Visualization

Interactive Exploration of Data Partition Sequences [7] This work introduces novel augmentations of alluvial diagrams [238] for the interactive exploration of data partition sequences; structures commonly resulting from clustering multivariate data. The work

highlights new perspectives on the visual analysis of multivariate ML data, as demonstrated on dimension reduction, clustering and ML training supervision. I contributed to the design of the measures and visualizations, curated the datasets, created the supplementary video, and supported parts of the writing process.

Visual Anomaly Detection in Temporal Knowledge Graphs [8] This work introduces a visual analytics system for detecting anomalies in temporal knowledge graphs, combining temporal and structural encodings in a coordinated visual interface. The system was awarded at the IEEE VAST Challenge 2024 for its *effective composition of visual encodings*. Although temporal knowledge graphs are outside the direct scope of this dissertation, they represent a particularly challenging case of multivariate data. I supervised the research project, particularly the visualization design, the creation of the supplementary video, and the writing of the manuscript.

Publications mentioned in this chapter:

[4] **J.-T. Sohns**, D. Gond, F. Jirasek, H. Hasse, G.H. Weber, and H. Leitte. "Embedding Space Explanations of Learned Mixture Behavior". *Proc. 3rd Conf. on Phys. Modeling for Virtual Manufacturing Systems and Processes*. 2023, pp. 32–50. DOI: 10.1007/978-3-031-35779-4_3

[6] D. Gond, **J.-T. Sohns**, H. Leitte, H. Hasse, and F. Jirasek. "Hierarchical Matrix Completion for the Prediction of Properties of Binary Mixtures". *Computers & Chemical Engineering* 199 (2025), p. 109122. DOI: 10.1016/j.compchemeng.2025.109122

[7] M. Poddar, **J.-T. Sohns**, and F. Beck. "Not Just Alluvial: Towards a More Comprehensive Visual Analysis of Data Partition Sequences". *Vision, Modeling, and Visualization*. 2024. DOI: 10.2312/vmv.20241202

[8] K. Iselborn, M. Allmann, **J.-T. Sohns**, and H. Leitte. "Visual Anomaly Detection in Temporal Knowledge Graphs". *2024 IEEE Visual Analytics Science and Technology VAST Challenge*. **Honorable Mention**. 2024, pp. 21–23. DOI: 10.1109/VASTChallenge64683.2024.00015

Chapter 7

Conclusions and Future Work

We conclude this dissertation with a brief summary of the results presented in the previous chapters and outline potential directions for future research.

7.1 Summary

The ever-increasing capability of machine learning models opens up both the possibility and the necessity to interpret the structures they learn. After surveying established techniques in multivariate visualization and explainable machine learning, we identified a demand for visualization tools for machine learning predictions that explicitly account for multivariate feature spaces. In response, this dissertation introduced three complementary approaches to visualize and analyze machine learning predictions, each addressing a distinct facet of multivariate interpretation: input-based, relationship-based, and knowledge-based analysis.

Input-based analysis We determined that visualizing decision boundaries in a way that preserves the original input distances is a promising approach to enhance multivariate counterfactual explanations. This dissertation presented CoFFi, an interactive tool to visually explore decision boundaries and counterfactual scenarios. CoFFi constructs local linear maps – 2D slices of the decision space around a specific instance – that integrate feature importance and mutability, prioritizing the most relevant features for analysis. Visual variants of established model inspection techniques complement the interactive exploration of the boundary shape and the customization of counterfactual examples. Case studies demonstrated that the tool can convey simple, complex, and malformed decision boundaries better than prior methods, while explicit probing revealed personalized multivariate counterfactuals with context.

Relationship-based analysis We demonstrated that non-linear dimensionality reductions, a fundamental machine learning technique, lacked the visualization means to comprehensively convey relationships between the original data attributes and the reduced projections, which is crucial for accurate interpretation. This dissertation introduced NoLiES, a visualization system that provides attribute-based explanations for non-linear projections. The centerpiece of NoLiES is the rangeset visualization method. Inspired by

algebraic topology, rangesets create non-convex hulls around data points binned by attribute values, effectively linking attribute distributions to specific regions of the embedding. Unlike previous color-coding techniques, rangesets simultaneously handle clusters, outliers and projection ambiguities. The NoLiES interface integrates small multiples to explore multiple attributes in parallel to enable a structural overview. NoLiES has proven effective in case studies of various data characteristics, including the analysis of learned features in thermodynamics.

Knowledge-based analysis We proposed that learned structures in modern models capture meaningful insights which should be analyzed in conjunction with existing domain knowledge. We identified that for scalar matrices, a common representation of relational predictions, there is limited support for enrichment analysis, which is essential for a comprehensive assessment against external attributes. This dissertation presented a hierarchical evaluation framework for visualizing and analyzing cluster patterns in scalar matrices using the example of binary mixture predictions. To this end, we revisited canonical matrix patterns, explored their translation to this application, and described a workflow to fit the matrix ordering. Our interactive software combines heatmaps, dendrograms, and linked detail views to explore hierarchical aggregation levels and validate them quantitatively against domain-specific data properties of various types. A case study demonstrated how this framework uncovers insights within predicted mixture data, contributing to the improvement of state-of-the-art prediction methods for thermodynamic properties. A user study affirmed the relevance and demand of domain-specific tools that integrate domain knowledge into (machine learning) matrix visualization to address real-world challenges in scientific research.

Subsequently, we reported on a successful **application case** that advanced the field of thermodynamic modeling, demonstrating the practical impact of the proposed approaches. The visualization part of the dissertation was further extended through two **related studies** proposing practical solutions for data partition sequences and temporal knowledge graphs.

In summary, this dissertation presented interactive visualization methods that enhance the explainability of machine learning predictions in multivariate spaces. By providing tools for input-based, relationship-based, and knowledge-based analysis, we addressed challenges and utilized opportunities occurring at this intersection of disciplines in a comprehensive manner. Together, these methods advance the field towards more interpretable multivariate spaces, enabling a more transparent, trustworthy, and insightful machine learning interpretation.

7.2 Future Work

The approaches presented in this dissertation open up several directions for future research.

More Domains The applied work demonstrated both the strong demand and significant practical potential of the proposed tools. They were systematically evaluated through case studies and expert user studies in line with established visualization research guidelines, providing clear evidence of their effectiveness and relevance. Building on this foundation, future large-scale studies across different domains could further assess generalizability, yield deeper insights into practical requirements, and guide future development.

More Dimensions All three approaches currently handle up to about 20 attribute dimensions in practice. Although sufficient for many engineering scenarios, real-world models often involve far more features. Scaling to more dimensions is therefore inevitable for broader applicability. As the visualization channels are already quite dense, this challenge will likely require adaptions of the underlying visualization algorithms, for example through more specialized dimensionality reduction, hierarchical techniques, or more guidance towards relevant dimensions.

More Observations The presented tools are demonstrated to be effective on datasets from a few hundred to a few thousand observations, while in practical machine learning applications this number is often significantly higher. This limitation arises both from the algorithmic and, more critically, the visual scalability. Large parts of the algorithmic runtimes could be improved with more efficient implementations or approximation techniques. However, the visual scaling remains constrained by fundamental visualization problems such as overplotting and clutter. Developing aggregation and abstraction techniques tailored to explanation tasks appears particularly promising.

More Guidance Current methods rely on manual exploration, which enables flexible and open-ended analysis, but is time-consuming and requires expertise. Introducing more automated guidance, such as recommendation systems, could accelerate the process and make it more accessible to a non-technical audience. This is particularly important in applied domains, where users are experts in their domain but not in visualization or machine learning. Our studies indicated substantial interest in such support even against the current rise of fully automated artificial intelligence tools, which are not yet reliable enough to replace transparent, data-driven exploration.

In summary, opportunities for improving scalability, applicability and guidance remain promising. Combined with the rapid increase in complexity of modern machine learning models, these opportunities suggest that the effective explanation of multivariate machine learning predictions is far from solved and will continue to be an urgent and fruitful area of research.

List of Publications

The following publications were authored or co-authored by me during the course of the PhD. Publications [2, 3, 4, 5, 6] are direct results of the research presented in this dissertation, while others reflect collaborations and parallel investigations conducted during the same period.

[1] **J.-T. Sohns**, G.H. Weber, and C. Garth. "Distributed Task-Parallel Topology-Controlled Volume Rendering". *Topological Methods in Data Analysis and Visualization VI* (2021). (Based on the author's master thesis). DOI: 10.1007/978-3-030-83500-2_4.

[2] **J.-T. Sohns**, M. Schmitt, F. Jirasek, H. Hasse, and H. Leitte. "Attribute-based Explanation of Non-Linear Embeddings of High-Dimensional Data". *IEEE Transactions on Visualization and Computer Graphics* 28.1 (2022), pp. 540–550. DOI: 10.1109/TVCG.2021.3114870.

[3] **J.-T. Sohns**, C. Garth, and H. Leitte. "Decision Boundary Visualization for Counterfactual Reasoning". *Computer Graphics Forum* 42.1 (2023), pp. 7–20. DOI: 10.1111/cgf.14650.

[4] **J.-T. Sohns**, D. Gond, F. Jirasek, H. Hasse, G.H. Weber, and H. Leitte. "Embedding Space Explanations of Learned Mixture Behavior". *Proc. 3rd Conf. on Phys. Modeling for Virtual Manufacturing Systems and Processes*. (2023), pp. 32–50. DOI: 10.1007/978-3-031-35779-4_3.

[5] **J.-T. Sohns**, D. Gond, F. Jirasek, H. Hasse, and H. Leitte. "Visual Scalar Matrix Evaluation: An Application to Thermodynamics". *VisGap - The Gap between Visualization Research and Visualization Software*. The Eurographics Association, (2024). DOI: 10.2312/visgap.20241121.

[6] D. Gond, **J.-T. Sohns**, H. Leitte, H. Hasse, and F. Jirasek. "Hierarchical Matrix Completion for the Prediction of Properties of Binary Mixtures". *Computers & Chemical Engineering* 199 (2025), p. 109122. DOI: 10.1016/j.compchemeng.2025.109122.

[7] M. Poddar, **J.-T. Sohns**, and F. Beck. "Not Just Alluvial: Towards a More Comprehensive Visual Analysis of Data Partition Sequences". *Vision, Modeling, and Visualization*. (2024). DOI: 10.2312/vmv.20241202.

[8] K. Iselborn, M. Allmann, **J.-T. Sohns**, and H. Leitte. "Visual Anomaly Detection in Temporal Knowledge Graphs". *2024 IEEE Visual Analytics Science and Technology VAST Challenge*. **Honorable Mention**. (2024), pp. 21–23. DOI: 10.1109/VASTChallenge64683.2024.00015.

References

[9] J.M. Durán and G. Pozzi. "Trust and Trustworthiness in AI". *Philosophy & Technology* 38.1 (2025), p. 16. DOI: 10.1007/s13347-025-00843-2.

[10] S. Liu et al. "Towards better analysis of machine learning models: A visual analytics perspective". *Visual Informatics* 1.1 (2017), pp. 48–56. DOI: 10.1016/j.visinf.2017.01.006.

[11] S. Lapuschkin et al. "Unmasking Clever Hans predictors and assessing what machines really learn". *Nature communications* 10.1 (2019), p. 1096. DOI: 10.1038/s41467-019-08987-4.

[12] P. Izmailov et al. "On feature learning in the presence of spurious correlations". *Proceedings of the 36th International Conference on Neural Information Processing Systems*. Curran Associates Inc., (2022). DOI: 10.48550/arXiv.2210.11369.

[13] W.L. Hamilton et al. "Diachronic word embeddings reveal statistical laws of semantic change". *Proceedings of the Annual Meeting of the Association for Computational Linguistics* 1 (2016), pp. 1489–1501. DOI: 10.18653/v1/P16-1141.

[14] G.P. Way and C.S. Greene. "Extracting a Biologically Relevant Latent Space from Cancer Transcriptomes with Variational Autoencoders". *Proceedings of Pacific Symposium on Biocomputing* 23 (2017), pp. 80–91. DOI: 10.1101/174474.

[15] Y. Liu et al. "Latent Space Cartography: Visual Analysis of Vector Space Embeddings". *Computer Graphics Forum* 38.3 (2019), pp. 67–78. DOI: 10.1111/cgf.13672.

[16] Z. Wu et al. "Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking". *Nature Communications* 14 (2023), p. 2585. DOI: 10.1038/s41467-023-38192-3.

[17] S.K.R. Homberg et al. "Interpreting Graph Neural Networks with Myerson Values for Cheminformatics Approaches". *ChemRxiv* (2024). DOI: 10.26434/chemrxiv-2023-1hxxc-v2.

[18] J. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". *Nature* 596.7873 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[19] Z. Yang et al. "AlphaFold2 and its applications in the fields of biology and medicine". *Signal Transduction and Targeted Therapy* 8.1 (2023), p. 115. DOI: 10.1038/s41392-023-01381-z.

[20] F. Jirasek et al. "Machine Learning in Thermodynamics: Prediction of Activity Coefficients by Matrix Completion". *The Journal of Physical Chemistry Letters* 11.3 (2020), pp. 981–985. DOI: 10.1021/acs.jpclett.9b03657.

[21] F. Jirasek et al. “Hybridizing physical and data-driven prediction methods for physicochemical properties”. *Chemical Communications* 56 (2020), pp. 12407–12410. DOI: 10.1039/d0cc05258b.

[22] G. Montavon et al. “Methods for interpreting and understanding deep neural networks”. *Digital Signal Processing* 73 (2018), pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011.

[23] C. Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. *Nature Machine Intelligence* 1.5 (2019), pp. 206–215. DOI: 10.1038/s42256-019-0048-x.

[24] G. Alicioglu and B. Sun. “A survey of visual analytics for Explainable Artificial Intelligence methods”. *Computers & Graphics* 102 (2022), pp. 502–520. DOI: 10.1016/j.cag.2021.09.002.

[25] M.T. Ribeiro et al. ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, (2016), pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[26] S.M. Lundberg and S.-I. Lee. “A Unified Approach to Interpreting Model Predictions”. *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4765–4774.

[27] J.H. Friedman. “Greedy Function Approximation: A Gradient Boosting Machine”. *The Annals of Statistics* 29.5 (2001), pp. 1189–1232. DOI: 10.1214/aos/1013203451.

[28] Y. Ma and R. Maciejewski. “Visual Analysis of Class Separations With Locally Linear Segments”. *IEEE Transactions on Visualization and Computer Graphics* 27.1 (2021), pp. 241–253. DOI: 10.1109/TVCG.2020.3011155.

[29] S. Wachter et al. “Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR”. *Harvard Journal of Law & Technology* 31 (2017), pp. 842–887. DOI: 10.48550/arXiv.1711.00399.

[30] Y. Ma et al. “A Visual Analytics Framework for Explaining and Diagnosing Transfer Learning Processes”. *IEEE Transactions on Visualization and Computer Graphics* 27.02 (2021), pp. 1385–1395. DOI: 10.1109/TVCG.2020.3028888.

[31] X. Yuan et al. “Dimension Projection Matrix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data”. *IEEE Transactions on Visualization and Computer Graphics* 19 (2013), pp. 2625–33. DOI: 10.1109/TVCG.2013.150.

[32] M. Espadoto et al. “UnProjection: Leveraging Inverse-Projections for Visual Analytics of High-Dimensional Data”. *IEEE Transactions on Visualization and Computer Graphics* 01 (2021). DOI: 10.1109/TVCG.2021.3125576.

[33] F.C.M. Rodrigues et al. “Constructing and Visualizing High-Quality Classifier Decision Boundary Maps”. *Information* 10.9 (2019). DOI: 10.3390/info10090280.

[34] K. Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720.

[35] L.v.d. Maaten and G. Hinton. “Visualizing data using t-SNE”. *Journal of machine learning research* 9 (2008), pp. 2579–2605.

[36] L. McInnes et al. “UMAP: Uniform Manifold Approximation and Projection”. *Journal of Open Source Software* 3.29 (2018), p. 861. DOI: 10.21105/joss.00861.

[37] A. Mayorga and M. Gleicher. “Splatterplots: Overcoming Overdraw in Scatter Plots”. *IEEE Transactions on Visualization and Computer Graphics* 19 (2013), pp. 1526–1538. DOI: 10.1109/TVCG.2013.65.

[38] L. Nonato and M. Aupetit. “Multidimensional Projection for Visual Analytics: Linking Techniques with Distortions, Tasks, and Layout Enrichment”. *IEEE Transactions on Visualization and Computer Graphics* 25 (2019), pp. 2650–2673.

[39] A. Lex et al. “Caleydo: Design and evaluation of a visual analysis framework for gene expression data in its biological context”. *IEEE Pacific Visualization Symposium (PacificVis)*. (2010), pp. 57–64. DOI: 10.1109/PACIFICVIS.2010.5429609.

[40] C. Nobre et al. “Lineage: Visualizing Multivariate Clinical Data in Genealogy Graphs”. *IEEE Transactions on Visualization and Computer Graphics* 25.3 (2019), pp. 1543–1558. DOI: 10.1109/TVCG.2018.2811488.

[41] N.F. Fernandez et al. “Clustergrammer, a web-based heatmap visualization and analysis tool for high-dimensional biological data”. *Scientific Data* 4.1 (2017), p. 170151. DOI: 10.1038/sdata.2017.151.

[42] R.A. Fisher. “The Use of Multiple Measurements in Taxonomic Problems”. *Annals of Eugenics* 7.2 (1936), pp. 179–188. DOI: 10.1111/j.1469-1809.1936.tb02137.x.

[43] S. Liu et al. “Visualizing High-Dimensional Data: Advances in the Past Decade”. *IEEE Transactions on Visualization and Computer Graphics* 23.3 (2017), pp. 1249–1268. DOI: 10.1109/TVCG.2016.2640960.

[44] A. Inselberg. “The plane with parallel coordinates”. *The Visual Computer* 1.2 (1985), pp. 69–91. DOI: 10.1007/BF01898350.

[45] J.M. Chambers. *Graphical Methods for Data Analysis*. 1st. Chapman and Hall/CRC, 1983. DOI: 10.1201/9781351072304.

[46] E.R. Tufte. “Envisioning Information”. *Optometry and Vision Science* 68.4 (1991), pp. 322–324. DOI: https://doi.org/10.1007/BF02618477.

[47] N. Elmqvist et al. “Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation”. *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1539–1148.

[48] I. Liiv. “Seriation and matrix reordering methods: An historical overview”. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 3.2 (2010), pp. 70–91. DOI: 10.1002/sam.10071.

[49] M. Behrisch et al. “Matrix Reordering Methods for Table and Network Visualization”. *Comput. Graph. Forum* 35.3 (2016), pp. 693–716. DOI: 10.1111/cgf.12935.

[50] S.T. Roweis and L.K. Saul. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. *Science* 290.5500 (2000), pp. 2323–2326. DOI: 10.1126/science.290.5500.2323.

[51] L. Cayton et al. “Algorithms for manifold learning”. *Univ. of California at San Diego Tech. Rep* 12.1-17 (2005), p. 1.

[52] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005. DOI: 10.1007/978-1-4757-2711-1.

[53] T. Sainburg et al. “Parametric UMAP Embeddings for Representation and Semisupervised Learning”. *Neural Computation* 33.11 (2021), pp. 2881–2907. DOI: 10.1162/neco_a_01434.

[54] M. Wattenberg et al. “How to Use t-SNE Effectively”. *Distill* (2016). DOI: 10.23915/distill.00002.

[55] J. Stahnke et al. “Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions”. *IEEE Transactions on Visualization and Computer Graphics* 22 (2016), pp. 629–638.

[56] M. Behrisch et al. “GUIRO: User-Guided Matrix Reordering”. English. *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 184–194. DOI: 10.1109/TVCG.2019.2934300.

[57] S. L’Yi et al. “XCluSim: A visual analytics tool for interactively comparing multiple clustering results of bioinformatics data”. *BMC bioinformatics* 16 (2015), S5. DOI: 10.1186/1471-2105-16-S11-S5.

[58] H.-M. Wu et al. “GAP: A Graphical Environment for Matrix Visualization and Cluster Analysis”. *Computational Statistics & Data Analysis* 54 (2010), pp. 767–778. DOI: 10.1016/j.csda.2008.09.029.

[59] M.I. Jordan and T.M. Mitchell. “Machine learning: Trends, perspectives, and prospects”. *Science* 349.6245 (2015), pp. 255–260. DOI: 10.1126/science.aaa8415.

[60] M.N. Hoque and K. Mueller. “Outcome-explorer: A causality guided interactive visual interface for interpretable algorithmic decision making”. *IEEE Transactions on Visualization and Computer Graphics* 28.12 (2021), pp. 4728–4740.

[61] P. Schramowski et al. “Making deep neural networks right for the right scientific reasons by interacting with their explanations”. *Nature Machine Intelligence* 2.8 (2020), pp. 476–486.

[62] D. Munechika et al. “Visual auditor: Interactive visualization for detection and summarization of model biases”. *2022 IEEE Visualization and Visual Analytics (VIS)*. (2022), pp. 45–49.

[63] Á.A. Cabrera et al. “FAIRVIS: Visual Analytics for Discovering Intersectional Bias in Machine Learning”. *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. (2019), pp. 46–56. DOI: 10.1109/VAST47406.2019.8986948.

[64] S. Van Den Elzen and J.J. Van Wijk. “Baobabview: Interactive construction and analysis of decision trees”. *IEEE Conference on Visual Analytics Science and Technology (VAST)*. (2011), pp. 151–160. DOI: 10.1109/VAST.2011.6102453.

[65] D. Smilkov and S. Carter. *TensorFlow Playground*. https://playground.tensorflow.org. Accessed: 2025-07-15. 2016.

[66] A.W. Harley. “An Interactive Node-Link Visualization of Convolutional Neural Networks”. *Advances in Visual Computing ISVC*. (2015), pp. 867–877. DOI: 10.1007/978-3-319-27857-5_77.

[67] F. Hohman et al. “Gamut: A Design Probe to Understand How Data Scientists Understand Machine Learning Models”. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, (2019), pp. 1–13. DOI: 10.1145/3290605.3300809.

[68] S. Ali et al. “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence”. *Information Fusion* 99 (2023), p. 101805. DOI: 10.1016/j.inffus.2023.101805.

[69] F. Hohman et al. “Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers”. *IEEE Transactions on Visualization and Computer Graphics* 25.8 (2019), pp. 2674–2693. DOI: 10.1109/TVCG.2018.2843369.

[70] A. Chatzimparmpas et al. “A survey of surveys on the use of visualization for interpreting machine learning models”. *Information Visualization* 19.3 (2020), pp. 207–233. DOI: 10.1177/147387162090467.

[71] R. Guidotti et al. “A Survey of Methods for Explaining Black Box Models”. *ACM Comput. Surv.* 51.5 (2019). DOI: 10.1145/3236009.

[72] European Parliament and Council of the European Union. *Regulation (EU) 2016/679*. of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). OJ L 119, pp. 1–88, May 4, 2016. URL: https://data.europa.eu/eli/reg/2016/679/oj.

[73] C. Rudin et al. “Interpretable machine learning: Fundamental principles and 10 grand challenges”. *Statistic Surveys* 16 (2022). DOI: 10.1214/21-SS133.

[74] T. Hastie and R. Tibshirani. “Generalized additive models”. *Statistical science* 1.3 (1986), pp. 297–310. DOI: 10.1214/ss/1177013604.

[75] D. Bhati et al. “A Survey of Post-Hoc XAI Methods From a Visualization Perspective: Challenges and Opportunities”. *IEEE Access* 13 (2025), pp. 120785–120806. DOI: 10.1109/ACCESS.2025.3581136.

[76] X. Zhu et al. “Fuzzy Rule-Based Local Surrogate Models for Black-Box Model Explanation”. *IEEE Transactions on Fuzzy Systems* 31.6 (2023), pp. 2056–2064. DOI: 10.1109/TFUZZ.2022.3218426.

[77] L. Breiman. “Random Forests”. *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[78] D.W. Apley and J. Zhu. “Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models”. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 82 (2016), pp. 1059–1086. DOI: 10.1111/rssb.12377.

[79] A. Goldstein et al. “Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation”. *Journal of Computational and Graphical Statistics* 24 (2013). DOI: 10.1080/10618600.2014.907095.

[80] M. Al-Shedivat et al. “Contextual Explanation Networks”. *Journal of Machine Learning Research* 21.194 (2020), pp. 1–44. URL: http://jmlr.org/papers/v21/18-856.html.

[81] J. Lei et al. “Distribution-Free Predictive Inference for Regression”. *Journal of the American Statistical Association* 113.523 (2018), pp. 1094–1111. DOI: 10.1080/01621459.2017.1307116.

[82] M.T. Ribeiro et al. “Anchors: High-Precision Model-Agnostic Explanations”. *AAAI Conference on Artificial Intelligence*. (2018), pp. 1527–1535. DOI: 10.5555/3504035.3504222.

[83] D. Alvarez Melis and T. Jaakkola. “Towards Robust Interpretability with Self-Explaining Neural Networks”. *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 7775–7784. DOI: 10.5555/3327757.3327875.

[84] Z.J. Wang et al. “CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization”. *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1396–1406. DOI: 10.1109/TVCG.2020.3030418.

[85] M. Atzmueller et al. “Explainable and interpretable machine learning and data mining”. *Data Mining and Knowledge Discovery* 38.5 (2024), pp. 2571–2595. DOI: 10.1007/s10618-024-01041-y.

[86] W. Saeed and C. Omlin. “Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities”. *Knowledge-Based Systems* 263 (2023), p. 110273. DOI: 10.1016/j.knosys.2023.110273.

[87] T. Miller. “Explanation in artificial intelligence: Insights from the social sciences”. *Artificial Intelligence* 267 (2019), pp. 1–38. DOI: 10.1016/j.artint.2018.07.007.

[88] B. Kovalerchuk et al. “Survey of Explainable Machine Learning with Visual and Granular Methods Beyond Quasi-Explanations”. *Interpretable Artificial Intelligence: A Perspective of Granular Computing*. Springer International Publishing, 2021, pp. 217–267. DOI: 10.1007/978-3-030-64949-4_8.

[89] U.M. Fayyad. “From Stochastic Parrots to Intelligent Assistants—The Secrets of Data and Human Interventions”. *IEEE Intelligent Systems* 38.3 (2023), pp. 63–67. DOI: 10.1109/MIS.2023.3268723.

[90] E.M. Bender et al. “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, (2021), pp. 610–623. DOI: 10.1145/3442188.3445922.

[91] Z. Ji et al. “Survey of Hallucination in Natural Language Generation”. *ACM Computing Surveys* 55.12 (2023). DOI: 10.1145/3571730.

[92] U. Ehsan et al. “The Who in XAI: How AI Background Shapes Perceptions of AI Explanations”. *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, (2024). DOI: 10.1145/3613904.3642474.

[93] S. Anjomshoae et al. “Context-based image explanations for deep neural networks”. *Image and Vision Computing* 116 (2021), p. 104310. DOI: 10.1016/j.imavis.2021.104310.

[94] D. Keim et al. “Visual Analytics: Definition, Process, and Challenges”. *Information Visualization: Human-Centered Issues and Perspectives*. Springer Berlin Heidelberg, 2008, pp. 154–175. DOI: 10.1007/978-3-540-70956-5_7.

[95] A. Chatzimparmpas et al. “FeatureEnVi: Visual Analytics for Feature Engineering Using Stepwise Selection and Semi-Automatic Extraction Approaches”. *IEEE Transactions on Visualization and Computer Graphics* 28.4 (2022), pp. 1773–1791. DOI: 10.1109/TVCG.2022.3141040.

[96] H. Strobelt et al. “Seq2seq-Vis: A Visual Debugging Tool for Sequence-to-Sequence Models”. *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2019), pp. 353–363. DOI: 10.1109/TVCG.2018.2865044.

[97] A. Oliveira. et al. “SDBM: Supervised Decision Boundary Maps for Machine Learning Classifiers”. *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - IVAPP,* SciTePress, (2022), pp. 77–87. DOI: 10.5220/0010896200003124.

[98] M. Espadoto et al. “Visual Analytics of Multidimensional Projections for Constructing Classifier Decision Boundary Maps”. *International Conference on Information Visualization Theory and Applications 10*. (2019), pp. 28–38. DOI: 10.5220/0007260800280038.

[99] D. Sacha et al. “Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis”. *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 241–250. DOI: 10.1109/tvcg.2016.2598495.

[100] M. Behrisch et al. “Magnostics: Image-Based Search of Interesting Matrix Views for Guided Network Exploration”. *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017), pp. 31–40. DOI: 10.1109/TVCG.2016.2598467.

[101] M.P. Neto and F.V. Paulovich. “Explainable Matrix - Visualization for Global and Local Interpretability of Random Forest Classification Ensembles”. *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2021), pp. 1427–1437. DOI: 10.1109/tvcg.2020.3030354.

[102] M. Espadoto et al. “Deep Learning Inverse Multidimensional Projections”. *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, (2019). DOI: 10.2312/EUROVA.20191118.

[103] P.E. Rauber et al. “Projections as visual aids for classification system design”. *Information Visualization* 17.4 (2018). PMID: 30263012, pp. 282–305. DOI: 10.1177/1473871617713337.

[104] F.C. M. Rodrigues et al. “Image-Based Visualization of Classifier Decision Boundaries”. *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. (2018), pp. 353–360. DOI: 10.1109/SIBGRAPI.2018.00052.

[105] A. Schulz et al. “Using Discriminative Dimensionality Reduction to Visualize Classifiers”. *Neural Processing Letters* 42 (2015), pp. 27–54. DOI: 10.1007/s11063-014-9394-1.

[106] A. Barbosa et al. “Visualizing and Interacting with Kernelized Data”. *IEEE Transactions on Visualization and Computer Graphics* 22 (2015), pp. 1–1. DOI: 10.1109/TVCG.2015.2464797.

[107] P. Lipton. “Contrastive Explanation”. *Royal Institute of Philosophy Supplement* 27 (1990), pp. 247–266. DOI: 10.1017/S1358246100005130.

[108] S. Barocas et al. “The Hidden Assumptions behind Counterfactual Explanations and Principal Reasons”. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. Association for Computing Machinery, (2020), pp. 80–89. DOI: 10.1145/3351095.3372830.

[109] Y. Wang et al. “A Perception-Driven Approach to Supervised Dimensionality Reduction for Visualization”. *IEEE Transactions on Visualization and Computer Graphics* 24.5 (2018), pp. 1828–1840. DOI: 10.1109/TVCG.2017.2701829.

[110] D.H. Jeong et al. “IPCA: An interactive system for PCA-based visual analytics”. *Comput. Graph. Forum* 28 (2009), pp. 767–774. DOI: 10.1111/j.1467-8659.2009.01475.x.

[111] J. Krause et al. “Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models”. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, (2016), pp. 5686–5697. DOI: 10.1145/2858036.2858529.

[112] A. Schulz et al. “DeepView: Visualizing Classification Boundaries of Deep Neural Networks as Scatter Plots Using Discriminative Dimensionality Reduction”. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*. (2020), pp. 2305–2311. DOI: 10.24963/ijcai.2020/319.

[113] P. Rheingans and M. DesJardins. “Visualizing high-dimensional predictive model quality”. *Proceedings Visualization 2000 (VIS)*. (2000), pp. 493–496. DOI: 10.1109/VISUAL.2000.885740.

[114] J. Xia et al. “LDSScanner: Exploratory Analysis of Low-Dimensional Structures in High-Dimensional Datasets”. *IEEE Transactions on Visualization and Computer Graphics*. Vol. 24(1). (2018), pp. 236–245. DOI: 10.1109/TVCG.2017.2744098.

[115] S. Ingram et al. “DimStiller: Workflows for dimensional analysis and reduction”. *2010 IEEE Symposium on Visual Analytics Science and Technology*. (2010), pp. 3–10. DOI: 10.1109/VAST.2010.5652392.

[116] A. Dhurandhar et al. “Model Agnostic Contrastive Explanations for Structured Data”. *arXiv* (2019). DOI: 10.48550/arXiv.1906.00117.

[117] A.V. Looveren and J. Klaise. “Interpretable Counterfactual Explanations Guided by Prototypes”. *Machine Learning and Knowledge Discovery in Databases. Research Track* (2021), pp. 650–665. DOI: 10.1007/978-3-030-86520-7_40.

[118] M. Chapman-Rounds et al. “EMAP: Explanation by Minimal Adversarial Perturbation”. *ArXiv* (2019). DOI: 10.48550/arXiv.1912.00872.

[119] S. Tsirtsis and M. Gomez-Rodriguez. “Decisions, Counterfactual Explanations and Strategic Behavior”. *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS)*. (2020), pp. 16749–16760. DOI: 10.5555/3495724.3497129.

[120] K. Sokol and P. Flach. “One Explanation Does Not Fit All: The Promise of Interactive Explanations for Machine Learning Transparency”. *KI - Künstliche Intelligenz* 34 (2020). DOI: 10.1007/s13218-020-00637-y.

[121] S. Dandl et al. “Multi-Objective Counterfactual Explanations”. *Lecture Notes in Computer Science* (2020), pp. 448–469. DOI: 10.1007/978-3-030-58112-1_31.

[122] R.K. Mothilal et al. “Explaining machine learning classifiers through diverse counterfactual explanations”. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (2020). DOI: 10.1145/3351095.3372850.

[123] O. Gomez et al. “ViCE: Visual Counterfactual Explanations for Machine Learning Models”. *Proceedings of the 25th International Conference on Intelligent User Interfaces*. Association for Computing Machinery, (2020), pp. 531–535. DOI: 10.1145/3377325.3377536.

[124] Y. Ming et al. “RuleMatrix: Visualizing and Understanding Classifiers with Rules”. *IEEE Transactions on Visualization and Computer Graphics* (2018), pp. 1–1. DOI: 10.1109/TVCG.2018.2864812.

[125] F. Cheng et al. “DECE: Decision Explorer with Counterfactual Explanations for Machine Learning Models”. *IEEE Transactions on Visualization and Computer Graphics* (2020), pp. 1–1. DOI: 10.1109/TVCG.2020.3030342.

[126] C. Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/. 2025.

[127] A. Dhurandhar et al. “Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives”. *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., (2018), pp. 592–603.

[128] J. Wexler et al. “The What-If Tool: Interactive Probing of Machine Learning Models”. *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019). DOI: 10.1109/TVCG.2019.2934619.

[129] J. Nam and K. Mueller. “TripAdvisor(N-D): A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail”. *IEEE Transactions on Visualization and Computer Graphics* 19 (2012). DOI: 10.1109/TVCG.2012.65.

[130] A. Tatu et al. “Subspace search and visualization to make sense of alternative clusterings in high-dimensional data”. *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*. (2012), pp. 63–72. DOI: 10.1109/VAST.2012.6400488.

[131] D. Mazumdar et al. “Random Forest Similarity Maps: A Scalable Visual Representation for Global and Local Interpretation”. *Electronics* 10 (2021), p. 2862. DOI: 10.3390/electronics10222862.

[132] E. Amorim et al. “iLAMP: Exploring high-dimensional spacing through backward multidimensional projection”. *Proceedings of IEEE Conference on Visual Analytics Science and Technology 2012*. (2012), pp. 53–62. DOI: 10.1109/VAST.2012.6400489.

[133] E. Amorim et al. “Facing the high-dimensions: Inverse projection with radial basis functions”. *Computers and Graphics* 48 (2015), pp. 35–47. DOI: 10.1016/j.cag.2015.02.009.

[134] D. Caragea et al. “Visual Methods for Examining SVM Classifiers”. *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics*. Springer Nature, 2008, pp. 136–153. DOI: 10.1007/978-3-540-71080-6_10.

[135] W.E. Lorensen and H.E. Cline. “Marching cubes: A high resolution 3D surface construction algorithm”. *ACM siggraph computer graphics* 21.4 (1987), pp. 163–169.

[136] J. van Wijk and R. van Liere. “HyperSlice: Visualization of Scalar Functions of Many Variables”. *Proceedings Visualization ’93* (1993), pp. 119–125. DOI: 10.1109/VISUAL.1993.398859.

[137] C.D. Hansen and C.R. Johnson. *Visualization handbook*. Elsevier, 2011. DOI: 10.5555/993936.

[138] R. Poyiadzi et al. “FACE: Feasible and Actionable Counterfactual Explanations”. *AIES ’20: AAAI/ACM Conference on AI, Ethics, and Society*. (2020), pp. 344–350. DOI: 10.1145/3375627.3375850.

[139] T. Laugel et al. “Comparison-Based Inverse Classification for Interpretability in Machine Learning”. *Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*. (2018). DOI: 10.1007/978-3-319-91473-2_9.

[140] M. Espadoto et al. “OptMap: Using Dense Maps for Visualizing Multidimensional Optimization Problems”. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (IVAPP)*. SciTePress, (2021), pp. 123–132. DOI: 10.5220/0010288501230132.

[141] M.T. Keane and B. Smyth. “Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)”. *Case-Based Reasoning Research and Development*. Springer International Publishing, (2020), pp. 163–178. DOI: 10.1007/978-3-030-58342-2_11.

[142] L. Chen et al. “Counterfactual Samples Synthesizing for Robust Visual Question Answering”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2020). DOI: 10.1109/TPAMI.2023.3290012.

[143] Y. Goyal et al. “Counterfactual Visual Explanations”. *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. PMLR, (2019), pp. 2376–2384. DOI: 10.48550/arXiv.1904.07451.

[144] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle).

[145] Bokeh Development Team. *Bokeh: Python library for interactive visualization*. 2020. URL: https://bokeh.org/.

[146] P. Rudiger. *Panel - A high-level app and dashboarding solution for Python*. https://panel.holoviz.org/index.html.

[147] R.A. Rossi and N.K. Ahmed. “The Network Data Repository with Interactive Graph Analytics and Visualization”. *AAAI*. (2015). URL: http://networkrepository.com.

[148] R. Caruana et al. “Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission”. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, (2015), pp. 1721–1730. DOI: 10.1145/2783258.2788613.

[149] S. Few. *Time on the Horizon*. 2008. URL: http://www.perceptualedge.com/articles/visual_business_intelligence/time_on_the_horizon.pdf.

[150] J. Heer et al. “Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations”. *Proceedings of the SIGCHI conference on human factors in computing systems*. (2009), pp. 1303–1312. DOI: 10.1145/1518701.1518897.

[151] K.R. Gabriel. “The Biplot Graphic Display of Matrices with Application to Principal Component Analysis”. *Biometrika* 58.3 (1971), pp. 453–467. DOI: 10.2307/2334381.

[152] A. Fisher et al. “All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously”. *Journal of Machine Learning Research* 20.177 (2019), pp. 1–81. URL: http://jmlr.org/papers/v20/18-760.html.

[153] L. McInnes et al. *UMAP: Performance Considerations*. https://umap-learn.readthedocs.io/en/latest/performance.html. Accessed: 2025-04-02. 2024.

[154] J. Klaise et al. “Alibi Explain: Algorithms for Explaining Machine Learning Models”. *Journal of Machine Learning Research* 22.181 (2021), pp. 1–7. URL: http://jmlr.org/papers/v22/21-0017.html.

[155] A. Freire et al. “Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study”. *2009 6th Latin American Robotics Symposium, LARS 2009*. (2009), pp. 1–6. DOI: 10.1109/LARS.2009.5418323.

[156] C. Ware. *Information Visualization: Perception for Design*. 3rd ed. Morgan Kaufmann, 2012. DOI: 10.1016/C2009-0-62432-6.

[157] A. Abid et al. “Exploring patterns enriched in a dataset with contrastive principal component analysis”. *Nature Communications* 9.2134 (2018). DOI: 10.1038/s41467-018-04608-8.

[158] T. Fujiwara et al. “Supporting Analysis of Dimensionality Reduction Results with Contrastive Learning”. *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2020), pp. 45–55. DOI: 10.1109/TVCG.2019.2934251.

[159] D.J. Lehmann and H. Theisel. “Orthographic Star Coordinates”. *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 2615–2624. DOI: 10.1109/TVCG.2013.182.

[160] J.W. Tukey. *Exploratory data analysis*. Vol. 2. Addison-Wesley Publishing Company Reading, Mass., 1977. DOI: 10.1002/bimj.4710230408.

[161] S. Aeberhard et al. “Comparative analysis of statistical pattern recognition methods in high dimensional settings”. *Pattern Recognition* 27.8 (1994), pp. 1065–1077. DOI: 10.1016/0031-3203(94)90145-7.

[162] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[163] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. DOI: 10.5555/1953048.2078195.

[164] N.S.-N. Lam. “Spatial Interpolation Methods: A Review”. *The American Cartographer* 10.2 (1983), pp. 129–150. DOI: 10.1559/152304083783914958.

[165] R.R. da Silva et al. “Attribute-based Visual Explanation of Multidimensional Projections”. *EuroVis Workshop on Visual Analytics (EuroVA)*. (2015), pp. 31–35. DOI: 10.2312/EUROVA.20151100.

[166] D. van Driel et al. “Enhanced Attribute-Based Explanations of Multidimensional Projections”. *EuroVis Workshop on Visual Analytics (EuroVA)* (2020). DOI: 10.2312/eurova.20201084.

[167] D.J. Lehmann and H. Theisel. “General projective maps for multidimensional data projection”. *Computer Graphics Forum*. Vol. 35. 2. Wiley Online Library. (2016), pp. 443–453. DOI: 10.1111/cgf.12845.

[168] M. Dowling et al. “SIRIUS: Dual, symmetric, interactive dimension reductions”. *IEEE Transactions on Visualization and Computer Graphics* 25.1 (2018), pp. 172–182. DOI: 10.1109/TVCG.2018.2865047.

[169] H. Lee et al. “iVisClustering: An Interactive Visual Document Clustering via Topic Modeling”. *Computer Graphics Forum* 31.3 (2012), pp. 1155–1164. DOI: 10.1111/j.1467-8659.2012.03108.x.

[170] L. Pagliosa et al. “Understanding attribute variability in multidimensional projections”. *29th Conference on Graphics, Patterns and Images (SIBGRAPI)*. (2016), pp. 297–304. DOI: 10.1109/SIBGRAPI.2016.048.

[171] D. Coimbra et al. “Explaining three-dimensional dimensionality reduction plots”. *Information Visualization* 15 (2015). DOI: 10.1177/1473871615600010.

[172] F.L. Dennig et al. “The Categorical Data Map: A Multidimensional Scaling-Based Approach”. *2024 IEEE Visualization in Data Science (VDS)* (2024), pp. 25–34. DOI: 10.1109/VDS63897.2024.00008.

[173] N. Cao et al. “FacetAtlas: Multifaceted Visualization for Rich Text Corpora”. *IEEE Transactions on Visualization and Computer Graphics* 16.6 (2010), pp. 1172–1181. DOI: 10.1109/TVCG.2010.154.

[174] B. Broeksema et al. “Visual Analysis of Multi-Dimensional Categorical Data Sets”. *Computer Graphics Forum* 32.8 (2013), pp. 158–169. DOI: 10.1111/cgf.12194.

[175] A. Skupin. “A cartographic approach to visualizing conference abstracts”. *IEEE Computer Graphics and Applications* 22.1 (2002), pp. 50–58. DOI: 10.1109/38.974518.

[176] M. Aupetit. “Visualizing Distortions and Recovering Topology in Continuous Projection Techniques”. *Neurocomputing* 70.7–9 (2007), pp. 1304–1330. DOI: 10.1016/j.neucom.2006.11.018.

[177] S. Cheng and K. Mueller. “The Data Context Map: Fusing Data and Attributes into a Unified Display”. *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 121–130. DOI: 10.1109/TVCG.2015.2467552.

[178] R. Faust et al. “DimReader: Axis lines that explain non-linear projections”. *IEEE Transactions on Visualization and Computer Graphics* 25 (2019), pp. 481–490. DOI: 10.1109/TVCG.2018.2865194.

[179] M. Cavallo and Ç. Demiralp. “A Visual Interaction Framework for Dimensionality Reduction Based Data Exploration”. *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, (2018), pp. 1–4. DOI: 10.1145/3170427.3186508.

[180] A. Chatzimparmpas et al. “t-viSNE: Interactive Assessment and Interpretation of t-SNE Projections”. *IEEE Transactions on Visualization and Computer Graphics* 26 (2020), pp. 2696–2714. DOI: 10.1109/TVCG.2020.2986996.

[181] S. Bachthaler and D. Weiskopf. “Continuous Scatterplots”. *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1428–1435. DOI: 10.1109/TVCG.2008.119.

[182] L. Micallef et al. “Towards Perceptual Optimization of the Visual Design of Scatterplots”. *IEEE Transactions on Visualization and Computer Graphics* 23 (2017), pp. 1588–1599. DOI: 10.1109/TVCG.2017.2674978.

[183] C. Collins et al. “Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations”. *IEEE Transactions on Visualization and Computer Graphics* 15.6 (2009), pp. 1009–1016. DOI: 10.1109/tvcg.2009.122.

[184] P. Simonetto et al. “Fully Automatic Visualisation of Overlapping Sets”. *Computer Graphics Forum* 28.3 (2009), pp. 967–974. DOI: 10.1111/j.1467-8659.2009.01452.x.

[185] T. Schreck et al. “Butterfly plots for visual analysis of large point cloud data”. *16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*. (2008). ISBN: 978-80-86943-15-2.

[186] S. Liu et al. “Distortion-Guided Structure-Driven Interactive Exploration of High-Dimensional Data”. *Computer Graphics Forum*. Vol. 33. 3. Wiley Online Library. (2014), pp. 101–110. DOI: 10.1111/cgf.12366.

[187] C. Seifert et al. “Stress Maps: Analysing Local Phenomena in Dimensionality Reduction Based Visualisations”. *EuroVis Workshop on Visual Analytics (EuroVA)*. (2010). DOI: 10.2312/PE/EuroVAST/EuroVAST10/013-018.

[188] B. Rieck et al. “Multivariate data analysis using persistence-based filtering and topological signatures”. *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2382–2391. DOI: 10.1109/TVCG.2012.248.

[189] B. Rieck and H. Leitte. “Persistent homology for the evaluation of dimensionality reduction schemes”. *Computer Graphics Forum*. Vol. 34. 3. Wiley Online Library. (2015), pp. 431–440. DOI: 10.1111/cgf.12655.

[190] H. Doraiswamy et al. “TopoMap: A 0-dimensional Homology Preserving Projection of High-Dimensional Data”. *IEEE Transactions on Visualization & Computer Graphics* 27.02 (2021), pp. 561–571. DOI: 10.1109/TVCG.2020.3030441.

[191] M. De Berg et al. “Computational Geometry”. *Computational geometry*. Springer, 1997, pp. 1–17. DOI: 10.1007/978-3-540-77974-2.

[192] H. Edelsbrunner et al. “On the shape of a set of points in the plane”. *IEEE Trans. Inf. Theory* 29 (1981), pp. 551–558. DOI: 10.1109/TIT.1983.1056714.

[193] J.P. May. *Simplicial objects in algebraic topology*. Vol. 11. University of Chicago Press, 1992. DOI: 10.2307/3620123.

[194] J.D. Hunter. “Matplotlib: A 2D graphics environment”. *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[195] P.V. Kerm. “Adaptive Kernel Density Estimation”. *The Stata Journal* 3.2 (2003), pp. 148–156. DOI: 10.1177/1536867x0300300204.

[196] E.M. Arkin et al. “On minimum-area hulls”. *Algorithmica* 21.1 (1998), pp. 119–136. DOI: 10.1007/PL00009204.

[197] L. Wilkinson et al. “Graph-theoretic scagnostics”. *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* (2005), pp. 157–164. DOI: 10.1109/INFVIS.2005.1532142.

[198] P. Joia et al. “Uncovering representative groups in multidimensional projections”. *Computer Graphics Forum*. Vol. 34. 3. (2015), pp. 281–290. DOI: 10.1111/cgf.12640.

[199] M. van Kreveld et al. “Contour trees and small seed sets for isosurface traversal”. *SCG ’97: Symposium on Computational Geometry*. Association for Computing Machinery, (1997), pp. 212–220. DOI: 10.1145/262839.269238.

[200] D. Freedman et al. *Statistics (Third ed.)* W. W. Norton, 1998. DOI: 10.2307/3314838.

[201] L. Wilkinson. “Visualizing Big Data Outliers Through Distributed Aggregation”. *IEEE Transactions on Visualization and Computer Graphics* 24.1 (2017), pp. 256–266. DOI: 10.1109/TVCG.2017.2744685.

[202] J. Talbot et al. “An extension of Wilkinson’s algorithm for positioning tick labels on axes”. *IEEE Transactions on visualization and computer graphics* 16.6 (2010), pp. 1036–1043. DOI: 10.1109/TVCG.2010.130.

[203] M. Kraus et al. “Assessing 2D and 3D Heatmaps for Comparative Analysis: An Empirical Study”. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (2020). DOI: 10.1145/3313831.3376675.

[204] R. Motta et al. “Graph-based measures to assist user assessment of multidimensional projections”. *Neurocomputing* 150 (2015), pp. 583–598. DOI: 10.1016/j.neucom.2014.09.063.

[205] R. Farouki and C. Neff. “Analytic properties of plane offset curves”. *Computer Aided Geometric Design* 7.1 (1990), pp. 83–99. DOI: 10.1016/0167-8396(90)90023-K.

[206] T. Kluyver et al. “Jupyter Notebooks – a publishing format for reproducible computational workflows”. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press. (2016), pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87.

[207] S. Gillies et al. *Shapely: manipulation and analysis of geometric objects*. toblerity.org, 2007. URL: https://github.com/Toblerity/Shapely.

[208] OECD. *Better Life*. URL: http://www.oecdbetterlifeindex.org.

[209] J.A. Blackard and D. Dean. “Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables”. *Computers and Electronics in Agriculture* 24 (1999), pp. 131–151. DOI: 10.1016/S0168-1699(99)00046-0.

[210] H.C. Carlson and A.P. Colburn. “Vapor-liquid equilibria of nonideal solutions”. *Industrial & Engineering Chemistry* 34.5 (1942), pp. 581–589. DOI: 10.1021/ie50389a013.

[211] F. Jirasek and H. Hasse. “Combining Machine Learning with Physical Knowledge in Thermodynamic Modeling of Fluid Mixtures”. *Annual Review of Chemical and Biomolecular Engineering* 14 (2023), pp. 31–51. DOI: 10.1146/annurev-chembioeng-092220-025342.

[212] U. Bauer. “Ripser: Efficient Computation of Vietoris–Rips Persistence Barcodes”. *Journal of Applied and Computational Topology* (2021), pp. 1–33. DOI: 10.1007/s41468-021-00071-5.

[213] F. Jirasek and H. Hasse. “Perspective: Machine Learning of Thermophysical Properties”. *Fluid Phase Equilibria* 549 (2021), p. 113206. DOI: 10.1016/j.fluid.2021.113206.

[214] J. Gmehling et al. “Group contribution methods for phase equilibrium calculations”. *Annual review of chemical and biomolecular engineering* 6 (2015), pp. 267–292. DOI: 10.1146/annurev-chembioeng-061114-123424.

[215] U. Onken et al. “The Dortmund Data Bank: A computerized system for retrieval, correlation, and prediction of thermodynamic properties of mixtures”. *International Journal of Thermophysics* 10.3 (1989), pp. 739–747. DOI: 10.1007/BF00507993.

[216] M. Sedlmair et al. “Design study methodology: Reflections from the trenches and the stacks”. *IEEE Transactions on Visualization and Computer Graphics* 18.12 (2012), pp. 2431–2440. DOI: 10.1109/TVCG.2012.213.

[217] M.L. Waskom. “Seaborn: Statistical Data Visualization”. *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021.

[218] P. Rudiger. *Holoviews*. https://holoviews.org.

[219] L. Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, 2005. DOI: 10.1007/0-387-28695-0.

[220] F. Lekschas et al. “HiPiler: Visual Exploration Of Large Genome Interaction Matrices With Interactive Small Multiples”. *IEEE Transactions on Visualization and Computer Graphics* (2018). DOI: 10.1109/TVCG.2017.2745978.

[221] C.-h. Chen. “Generalized Association Plots: information visualization via iteratively generated correlation matrices”. *Statistica Sinica* 12 (2002), pp. 7–29. URL: https://www.jstor.org/stable/24307033.

[222] H.-M. Wu et al. “Matrix Visualization – Handbook of Data Visualization”. Springer Berlin, Heidelberg, 2008, pp. 681–708. DOI: 10.1007/978-3-540-33037-0.

[223] C. Partl et al. “enRoute: Dynamic Path Extraction from Biological Pathway Maps for Exploring Heterogeneous Experimental Datasets”. *BMC Bioinformatics* 14 (2013), S3. DOI: 10.1186/1471-2105-14-S19-S3.

[224] A. Barsky et al. “Cerebral: Visualizing Multiple Experimental Conditions on a Graph with Biological Context”. *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1253–1260. DOI: 10.1109/TVCG.2008.117.

[225] S.K. Card and D. Nation. “Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface”. *Proceedings of the Working Conference on Advanced Visual Interfaces*. Association for Computing Machinery, (2002), pp. 231–245. DOI: 10.1145/1556262.1556300.

[226] J. Chen et al. “Constructing Overview + Detail Dendrogram-Matrix Views”. *IEEE Transactions on Visualization and Computer Graphics* 15 (2010), pp. 889–96. DOI: 10.1109/TVCG.2009.130.

[227] D. Guo et al. “A visualization system for space-time and multivariate patterns (VIS-STAMP)”. *IEEE Transactions on Visualization and Computer Graphics* 12.6 (2006), pp. 1461–74. DOI: 10.1109/TVCG.2006.84.

[228] S. Engle et al. “Unboxing cluster heatmaps”. *BMC Bioinformatics* 18.2 (2017), p. 63. DOI: 10.1186/s12859-016-1442-6.

[229] U. Brandes. “Optimal leaf ordering of complete binary trees”. *Journal of Discrete Algorithms* 5.3 (2007). Selected papers from Ad Hoc Now 2005, pp. 546–552. DOI: 10.1016/j.jda.2006.09.003.

[230] M.P. Baroni and C.G. da Silva. “A comparative analysis of matrix reordering algorithms regarding canonical data patterns”. *Information Visualization* 21.3 (2022), pp. 321–332. DOI: 10.1177/14738716221091487.

[231] M.B. Eisen et al. “Cluster analysis and display of genome-wide expression patterns”. *Proceedings of the National Academy of Sciences* 95.25 (1998), pp. 14863–14868. URL: https://www.pnas.org/content/95/25/14863.

[232] J. Seo and B. Shneiderman. “Interactively Exploring Hierarchical Clustering Results”. *Computer* 35 (2002), pp. 80–86. DOI: 10.1109/MC.2002.1016905.

[233] G. Caraux and S. Pinloche. “PermutMatrix: a graphical environment to arrange gene expression profiles in optimal linear order”. *Bioinformatics* 21.7 (2004), pp. 1280–1281. DOI: 10.1093/bioinformatics/bti141.

[234] M.C. Ryan et al. “Interactive Clustered Heat Map Builder: An easy web-based tool for creating sophisticated clustered heat maps”. *F1000Research* 8 (2019), ISCB Comm J–1750. DOI: 10.12688/f1000research.20590.2.

[235] N. Arinik et al. “Characterizing and Comparing External Measures for the Assessment of Cluster Analysis and Community Detection”. *IEEE Access* 9 (2021), pp. 20255–20276. DOI: 10.1109/ACCESS.2021.3054621.

[236] A. Fredenslund et al. “Group-contribution estimation of activity coefficients in nonideal liquid mixtures”. *AIChE Journal* 21.6 (1975), pp. 1086–1099. DOI: 10.1002/aic.690210607.

[237] J. Stahnke et al. “Probing Projections: Interaction Techniques for Interpreting Arrangements and Errors of Dimensionality Reductions”. *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2016), pp. 629–638. DOI: 10.1109/TVCG.2015.2467717.

[238] M. Rosvall and C.T. Bergstrom. “Mapping Change in Large Networks”. *PLOS ONE* 5.1 (2010), pp. 1–7. DOI: 10.1371/journal.pone.0008694.

Full-Size Versions of Figures

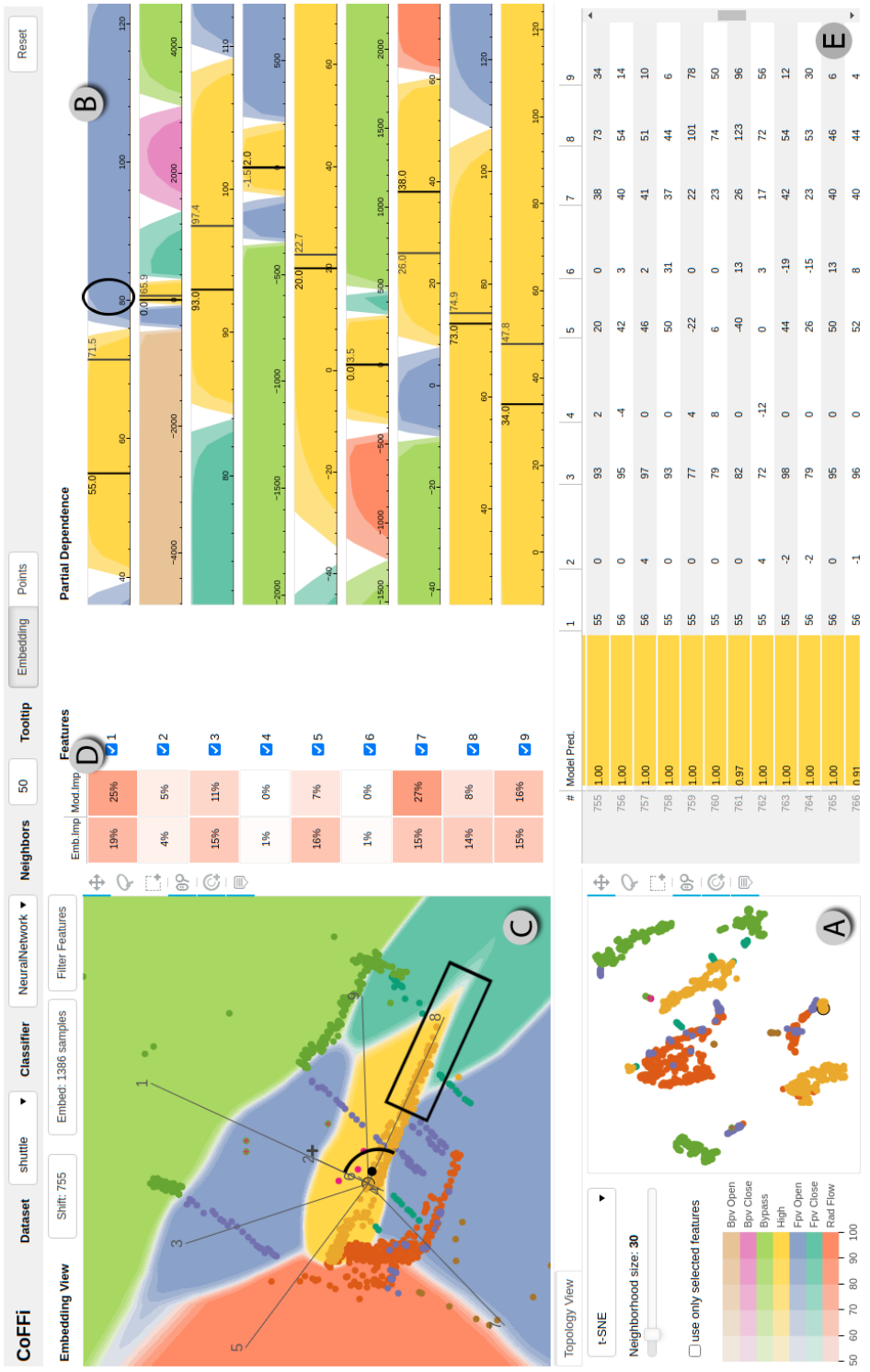


Figure 1: CoFFi interface on a space shuttle dataset [144], explaining decision boundaries of sample 755. The analysis comprises (A) cluster distribution, (B) univariate analysis, (C) multivariate analysis, (D) feature importance, and (E) a data table. Black annotations in (B) & (C) highlight conclusions of Section 3.4.3 & Section 3.4.4.

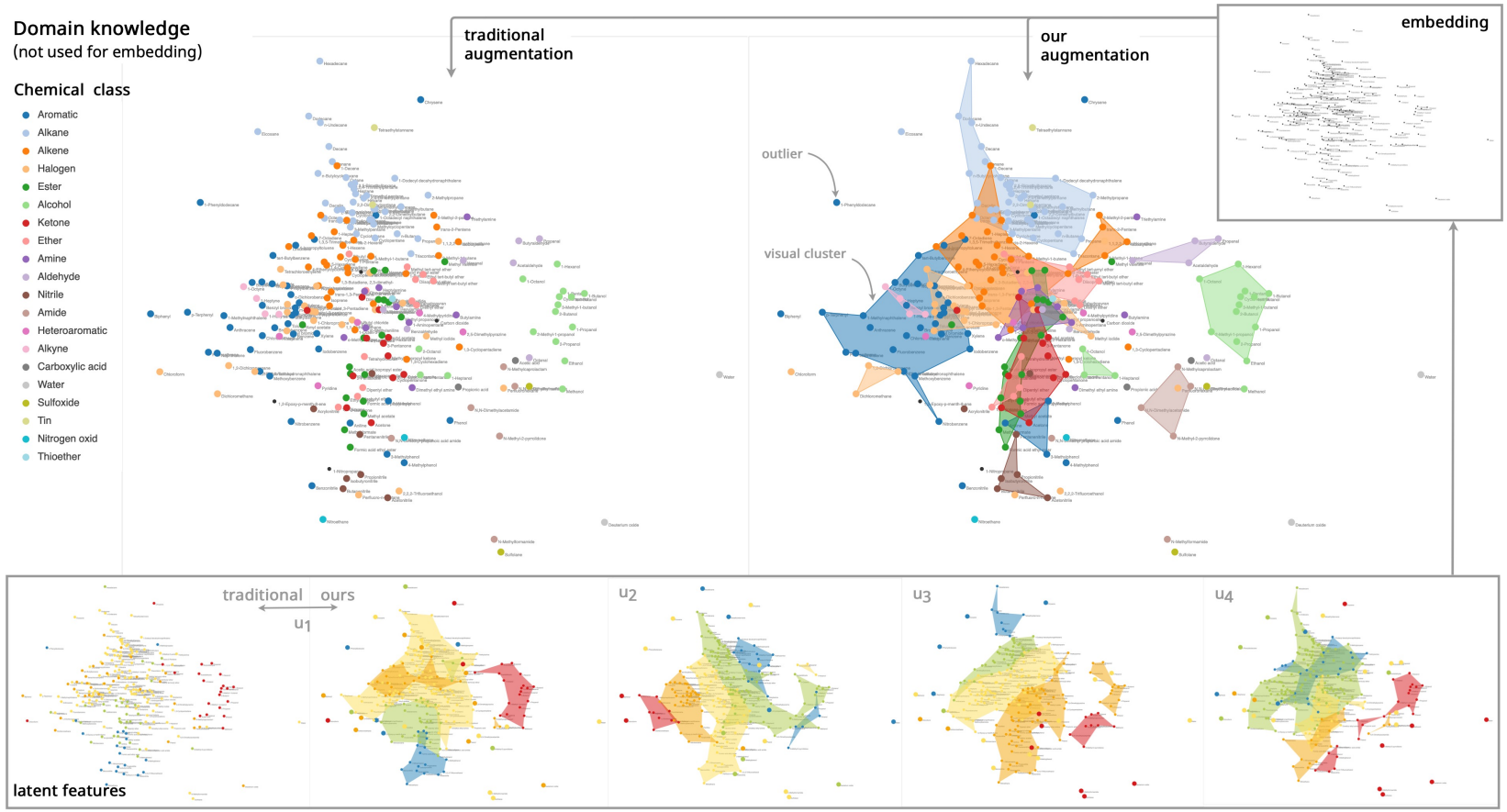


Figure 2: Explaining machine learning: The embedding (MDS, top right) shows a non-linear projection of the 4D latent feature space as computed by machine learning for 240 chemical compounds [20]. Color-based augmentation helps relate structures in the embedded point cloud to input attributes (bottom, continuous variables) and domain knowledge (top, categorical variables). Our new approach enabled domain scientists to detect previously unrecognizable patterns and outliers in their data.

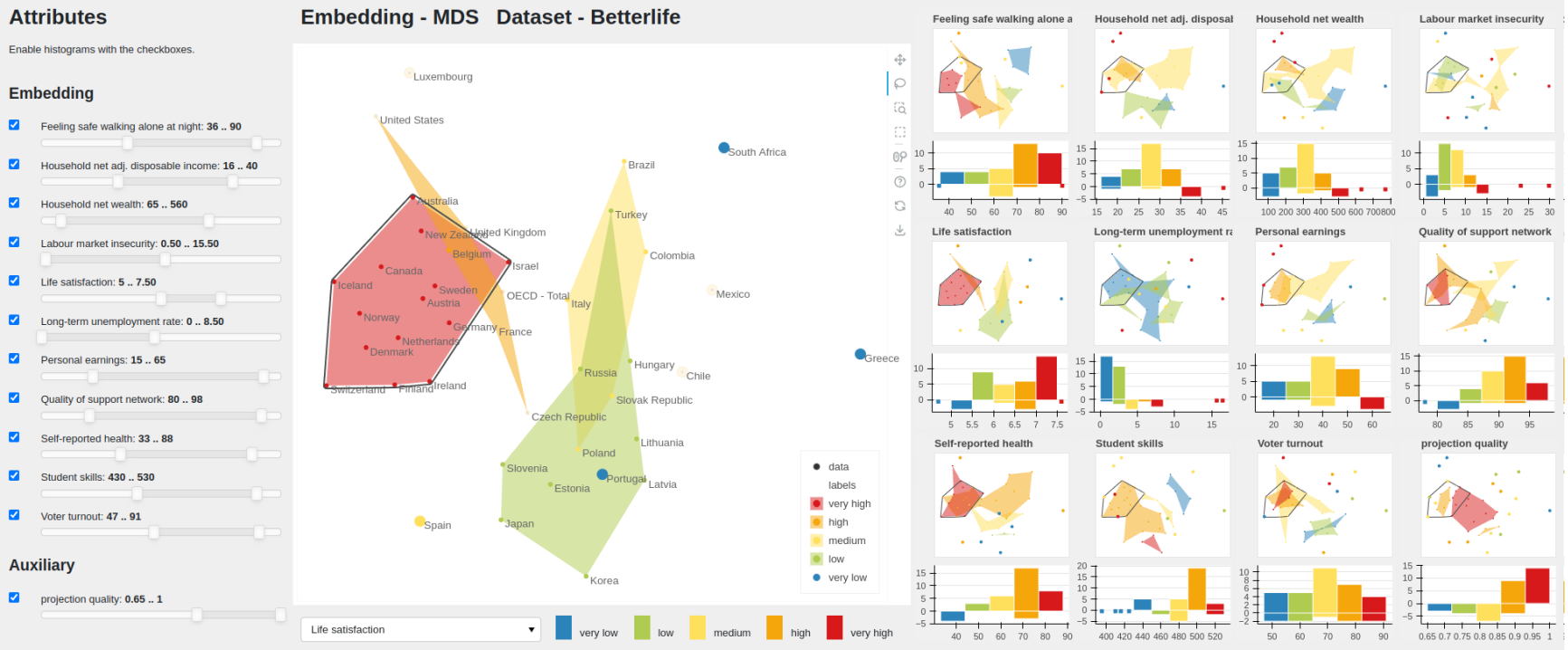


Figure 3: NoLiES on the OECD Better Life data [208] reveals multiple clusters that partially align with self-perceived life satisfaction (center). The gray selection transfers the ■ very happy countries to the small multiples of all attributes (right). Custom ranges are set with sliders (left) to achieve more expressive discretization.

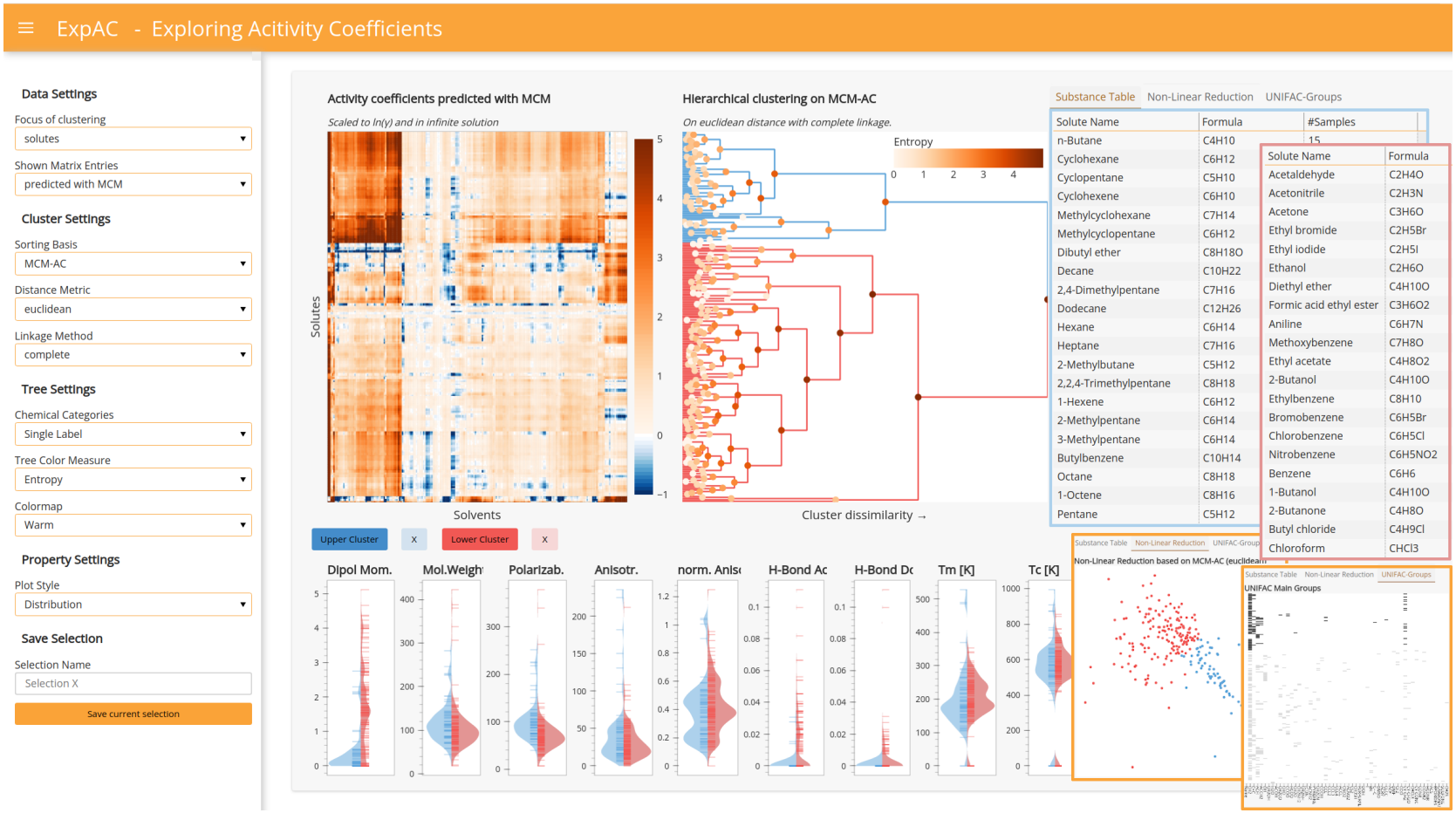


Figure 4: Exploring Mixture Data: A sorted heatmap visually groups blocks of similar chemical substances (rows and columns). Pattern strength is analyzed by variation in internal and external data. Linked widgets connect the discovered groups to additional domain knowledge. A higher-resolution version of this figure is available in appendix A.

Academic Curriculum Vitae

2019 – 2026 **PhD in Computer Science**
RPTU University Kaiserslautern-Landau, Germany
Supervisor: Prof. Dr. Heike Leitte
Interactive Exploration of Model Predictions for Multivariate Data

2018 **Research Stay**
Lawrence Berkeley National Laboratory, Berkeley, USA
Supervisor: Dr. Gunther H. Weber
Scientific Visualization on the Cori Supercomputer

2016 – 2019 **Master of Science in Computer Science**
University of Kaiserslautern, Germany
Supervisor: Prof. Dr. Christoph Garth
Distributed Task-Parallel Topology-Controlled Volume Rendering

2012 – 2016 **Bachelor of Science in Computer Science**
University of Kaiserslautern, Germany
Supervisor: Prof. Dr. Christoph Garth
Evaluating a Task-Parallel Implementation of Sort-Last Rendering in HPX