From Confusion to Clarity: ProtoScore - A Framework for Evaluating Prototype-Based XAI

Helena Monke
helena.monke@ipa.fraunhofer.de
Fraunhofer Institute for Manufacturing Engineering and Automation IPA
Stuttgart, Germany,
Institute of Industrial Manufacturing and Management IFF, University of Stuttgart
Stuttgart, Germany

Benjamin Sae-Chew
benjamin.sae-chew@ipa.fraunhofer.de
Fraunhofer Institute for Manufacturing Engineering and Automation IPA
Stuttgart, Germany,
TUM School of Computation, Information and Technology, Technical University of Munich
Munich, Germany

Benjamin Fresz
benjamin.fresz@ipa.fraunhofer.de
Fraunhofer Institute for Manufacturing Engineering and Automation IPA
Stuttgart, Germany,
Institute of Industrial Manufacturing and Management IFF, University of Stuttgart
Stuttgart, Germany

Marco F. Huber
marco.huber@ieee.org
Fraunhofer Institute for Manufacturing Engineering and Automation IPA
Stuttgart, Germany,
Institute of Industrial Manufacturing and Management IFF, University of Stuttgart
Stuttgart, Germany

Abstract

The complexity and opacity of neural networks (NNs) pose significant challenges, particularly in high-stakes fields such as healthcare, finance, and law, where understanding decision-making processes is crucial. To address these issues, the field of eXplainable Artificial Intelligence (XAI) has developed various methods aimed at clarifying AI decision-making, thereby facilitating appropriate trust and validating the fairness of outcomes. Among these methods, prototype-based explanations offer a promising approach that uses representative examples to elucidate model behavior. However, a critical gap exists regarding standardized benchmarks to objectively compare prototype-based XAI methods, especially in the context of time series data. This lack of reliable benchmarks results in subjective evaluations, hindering progress in the field. We aim to establish a robust framework, ProtoScore, for assessing prototype-based XAI methods across different data types with a focus on time series data, facilitating fair and comprehensive evaluations. By integrating the Co-12 properties of Nauta et al., this framework allows for effectively comparing prototype methods against each other and against other XAI methods, ultimately assisting practitioners in selecting appropriate explanation methods while minimizing the costs associated with user studies. All code is publicly available via GitHub.

FAccT '25, Athens, Greece

ACM ISBN 979-8-4007-1482-5/2025/06
https://doi.org/10.1145/3715275.3732151

CCS Concepts

• **Computing methodologies** → **Machine learning**; • **General and reference** → **Evaluation**; **Metrics**; • **Human-centered computing** → *Human computer interaction (HCI)*.

Keywords

eXplainable AI, XAI, XAI properties, prototype explanation, explanation metrics, benchmark, technical evaluation, objective evaluation, automated evaluation

ACM Reference Format:
Helena Monke, Benjamin Sae-Chew, Benjamin Fresz, and Marco F. Huber. 2025. From Confusion to Clarity: ProtoScore - A Framework for Evaluating Prototype-Based XAI. In *The 2025 ACM Conference on Fairness, Accountability, and Transparency (FAccT '25), June 23–26, 2025, Athens, Greece.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3715275.3732151

1 Introduction

Neural networks (NNs) have become essential tools in machine learning, showing strong performance in applications such as image recognition, natural language processing, and many others. However, their high complexity and lack of transparency present challenges that limit their widespread use. The "black-box" nature of NNs complicates users ability to understand how decisions are made and the reasons behind the model's behaviors [36]. This is especially relevant in critical fields such as healthcare, finance, and law, where decisions can have significant consequences [8]. The importance of explanations for AI systems has also been acknowledged by policymakers and courts as shown by XAI methods being used in court cases such as in the Australian case of ACCC v Trivago [10], or rulings on the use of XAI by the Court of Justice of the European Union [6, 7], and the—now explicit—"Right to Explanation" within the GDPR [33] and AI Act Art. 86 [34]. To tackle the

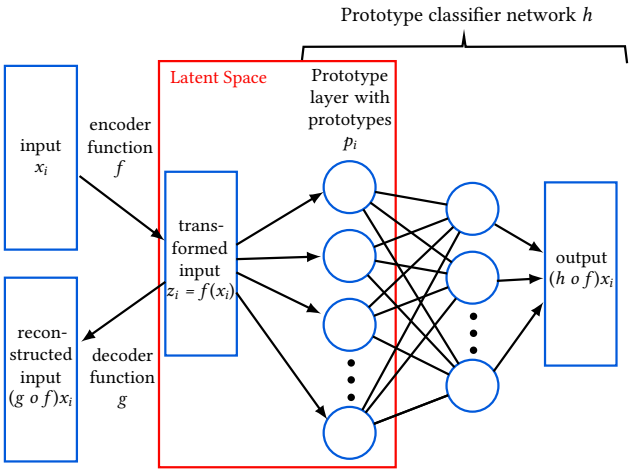


Figure 1: Architecture of a typical prototype network following [19]. It consists of an autoencoder and a classification network. In the prototype layer, the data samples are represented, e.g., as weighted distances of their latent representations to the prototypes. Other prototype methods may have slightly different structures. The benchmark focuses primarily on the latent space marked with the red rectangular.

aforementioned issues, the field of eXplainable Artificial Intelligence (XAI) has developed methods to clarify AI decision-making, enhancing human understanding of AI outputs and validating the fairness and reliability of those decisions. XAI techniques offer insights into a model's reasoning, aiding in identifying biases or flaws, and thus calibrate user trust [21]. In addition, there exist several approaches to guide practitioners in selecting the most appropriate XAI methods for the task at hand [40, 41].

One class of XAI methods uses "prototypes", which are representative examples that elucidate a model's behavior. A typical architecture of a prototype network is shown in Figure 1. The encoder maps the input data to a latent space, where prototypes are computed for data subgroups in the prototype layer, enabling classification based on proximity to these reference points. The decoder transforms prototypes back to the input space. Prototype-based methods are inherently interpretable and generate predictions and explanations by comparing input data to learned prototypes representing the training data [26]. These prototypes, which closely resemble the training observations, collectively represent the entire dataset [19]. They effectively map class examples to outputs, offering clarity through similarity reasoning and offering intuitive insights into the model's decision-making process. This approach emphasizes representative examples rather than relying on post-hoc explanations based on individual feature contributions (so-called "Saliency Maps"), where *post-hoc* describes explanations being created after model training and inference, in contrast to *ante-hoc* explanations, which are part of the model itself. Additionally, visualizing prototypes with input data helps identifying patterns and anomalies, enhancing comprehension of the underlying data distribution and model behavior.

For prototype methods, a major challenge remains: the lack of standardized benchmarks to objectively compare the performance of prototype-based XAI methods, especially for time series data. This absence hinders researchers and practitioners from determining the best prototype-based explainability method for specific applications. Without reliable benchmarks, evaluations tend to be subjective and inconsistent, leading to fragmented findings and slow progress in the field [45]. Researchers frequently use human evaluations and user studies, respectively, for explanation methods [23]. While this approach is a valuable complement to analysis [15], relying on human judgment to identify correct explanations poses challenges. The concept of an "accurate" explanation can vary significantly based on the task, and humans may not consistently assess the quality reliably [44]. Recent studies reveal that quantitative evaluations of explainable methods also have significant limitations. It is important to be cautious about the metrics used to evaluate explanation methods, as many existing metrics may not accurately reflect the user understanding or model behavior [13]. Existing frameworks and metrics provide insights for assessing various explanation types, but they often lack the specificity needed to evaluate prototype-based explanations, especially in time series analysis. For a review on existing methods, see Section 2 and Appendix A. The limited benchmarks targeting time series data highlight a critical gap in the current landscape of XAI evaluations. Prototype-based methods are promising for their interpretability and relevance for time series data. However, the absence of tailored evaluation frameworks makes it difficult to ascertain their effectiveness and reliability in practice.

Our contributions to the field of evaluating XAI methods are:

- A standardized and robust framework for automated, fair, comprehensive and reproducible evaluation of prototype-based explainability methods.
- Introduction of two additional conceptual attributes for prototypes that were not covered by Nauta et al. [28].
- Interpretation and adaptation of all conceptual attributes specifically for prototypes.
- Definition of corresponding metrics for all conceptual attributes for prototypes.
- Facilitation of comparisons across different data types, such as time series data and image data.
- Assistance in selecting explanations for user studies without replacing them, to minimize the overall financial and time costs associated with conducting large experiments.

We first provide a comprehensive background on existing evaluation methods for XAI, especially prototype-based methods, highlighting current challenges. In Section 3, we introduce the criteria and associated metrics forming the foundation of our benchmark. We then present a practical use case to illustrate the applicability of our proposed framework in Section 4 and demonstrate how different prototype methods can be evaluated with the established criteria to reveal the strengths and weaknesses of the prototype methods. Finally, we discuss the evaluation results and functionality of the benchmark tool in Section 5 and close the paper by summarizing our findings and suggesting directions for future research in prototype-based XAI evaluations.

2 Related Work

Various reviews, qualitative and quantitative evaluations, and benchmarks exist for XAI. Some reviews aid in selecting the appropriate explanation type, such as feature importance, counterfactuals, or prototype-based explanations. There exists a first comprehensive literature review on XAI for time series, distinct from image classification [40]. Other surveys focus on assessing XAI approaches. The *Co-12 properties* [28], a set of 12 conceptual attributes such as Compactness and Correctness, provide a foundation for evaluating explanation quality. Therefore, Chapter 3 provides more detailed information on the Co-12 properties, as they will be used to structure the proposed benchmark. Evaluation methods based on these Co-12 properties for part-prototypes have been proposed [27]. *Part-prototypes* cover a limited number of input dimensions, e.g., only show part of an image, and not entire input samples, as done with *non-part prototypes*. However, these suggestions for the usage of existing evaluation methods are specifically designed for part-prototypes in computer vision and may not assess all 12 properties for other data types and for non-part-prototypes, for example time series prototypes. Benchmarks for evaluating XAI methods on selected datasets also exist. Out of a couple of such studies, one evaluates XAI methods for time series data and introduces two novel approaches such as perturbation and sequence analysis [38]. But it focuses only on Saliency Map methods, such as LRP, LIME, DeepLift, and SHAP, excluding prototype methods. These evaluation methods are employed to validate XAI explanations, including LIME, SHAP, and Path Integrated Gradients (PIG), specifically adapted for multivariate time series classification in the maritime domain [42].

Several frameworks have emerged to compare and evaluate XAI methods, with *Quantus* [14] being the only one applicable to prototype-based XAI. A detailed description of further evaluation frameworks can be found in Appendix A. Quantus provides 37 evaluation metrics, covering a wide range of explanation types, including feature importance, heatmaps, localization, prototypes, and decision trees/rules methods. The metrics are categorized into faithfulness, robustness, localization, complexity, randomization, and axiomatic properties. However, some metrics represent different versions of the same concept, most are only applicable to attribution methods—not for prototypes—and the library lacks public benchmarks. The most common automated approaches for the quantitative evaluation of XAI methods were investigated by Le et al. [18], who identified and analyzed 17 toolkits regarding the coverage of the Co-12 properties of Nauta et al. [28] and whether they yield consistent results. While multiple toolkits exist for standard explanation tasks such as feature importance and heatmaps, benchmarks for assessing prototype methods are rarely found. The analysis noted that time series data is supported only by Quantus [14], which offers limited assessment for prototype explanations, focusing on image data with just three metrics. Despite a year and a half having passed since the publication of Le et al. [18], no new benchmarks for prototype methods have been proposed.

While significant progress in evaluating and benchmarking XAI methods, challenges remain, particularly for prototype-based approaches in time series data. To address these issues, we develop a dedicated benchmark framework *ProtoScore*, to evaluate prototype-based XAI methods across various data types, but especially for time series data. Our framework integrates the Co-12 properties to ensure a comprehensive assessment of explanation quality. The use of established properties allows for meaningful comparison of prototype methods to other XAI approaches evaluated along the same dimensions.

3 Methodology

Our benchmark, *ProtoScore*, is designed to evaluate trained prototype models, focusing on their explainability properties. To define specific metrics and assess the quality of prototypes adequately, certain assumptions are necessary. These assumptions include that a prototype closely resembles or is identical to a data instance, and that the set of prototypes represents the entire dataset [24]. Consequently, prototype methods are excluded, which optimize prototypes for proximity to the decision boundary, as discussed by Wang et al. [43], as they do not represent the dataset. The focus of the benchmark is strictly on the technical aspects of prototypes, using the latent space of the prototype model. Therefore, only models that incorporate a latent space can be evaluated with our benchmark tool. To begin, we define essential quantities that are necessary to evaluate the prototypes in Section 3.1. The development of ProtoScore is primarily based on the properties outlined by [27, 28] for comprehensive and comparable evaluation. These properties are described in Section 3.2, accompanied by two additional properties we deem necessary for prototype methods. We provide metrics evaluating each property, which are specially adapted for prototypes, yielding individual scores that are then averaged to produce a total score. The implementation details are described in Section 3.3.

3.1 Preliminaries

We begin by defining key concepts that will be used to establish metric later. Given a set of input samples

$$\mathcal{D} = \{x_1, x_2, \ldots, x_N\} \quad \text{with } x_i \in \mathbb{R}^d \text{ and } |\mathcal{D}| = N\,,$$

we define an encoder function that maps each data point x_i to its latent representation z_i in a lower-dimensional space

$$f : \mathbb{R}^d \to \mathbb{R}^n \quad x_i \mapsto z_i \quad \text{with } d \gg n\,, \tag{1}$$

where d is the input space dimension and n is the latent space dimension. The latent representation of $\mathcal{D}$ is given by

$$\mathcal{D}_{\text{enc}} = \{f(x_1), f(x_2), \ldots, f(x_N)\} = \{z_1, z_2, \ldots, z_N\}$$
$$\text{with } x_i \in D \text{ and } z_i \in \mathcal{D}_{\text{enc}}\,.$$

In this latent space we identify the set of prototypes, where in some prototype methods $\mathcal{P} \subseteq \mathcal{D}_{\text{enc}}$, by

$$\mathcal{P} = \{p_1, p_2, \ldots, p_M\}\,, \quad \text{with } p_i \in \mathbb{R}^n \text{ and } |\mathcal{P}| = M\,.$$

The prototype classifier network for predicting y_i from z_i is defined as

$$h : \mathbb{R}^n \to \mathbb{R} \quad z_i \mapsto y_i\,, \tag{2}$$

resulting in the overall output given by $y_i = (h \circ f)(x_i)$. The decoder function is used to transform each data point or associated prototype from the latent representation back to the input space

$$g : \mathbb{R}^n \to \mathbb{R}^d \quad z_i \mapsto x_i\,, \quad p_j \mapsto x_{p_j}\,. \tag{3}$$

In the latent space, the dataset $\mathcal{D}_{\text{enc}}$ can be grouped into clusters C_i, resulting in the set of clusters

$$C = \{C_1, C_2, \ldots, C_K\} \quad \text{with} \bigcup_i C_i = D_{\text{enc}} \quad \text{and } |C| = K \, .$$

The average distance from a point z_i to a cluster C_j is defined as

$$a(z_i, C_j) = \frac{1}{|C_j \setminus \{z_i\}|} \sum_{z \in C_j \setminus \{z_i\}} \|z_i - z\|_2 \, ,$$

where $\|z_i - z\|_2$ indicates the Euclidean distance. In this calculation z_i is excluded to avoid considering the distance from a data point to itself. The motivation to use the Euclidean distance instead of other metrics like the ones suggested in Hanawa et al. [12] is its straightforward interpretation. This is particularly relevant in a clustering context, where the aim is to minimize the geometric distance between data points. The Euclidean distance is robust and less sensitive to outliers compared to other distance metrics. Moreover, metrics like cosine similarity focus only on the direction and ignore the magnitude of the distance between points. Thus, two points can be far apart in space yet share the same direction. The Silhouette score [35] is calculated using the mean intra-cluster distance $a(z_i, C_k)$, where $z_i \in C_k$, and the mean nearest-cluster distance $a(z_i, C_m)$, where

$$C_m \in \underset{l \neq k}{\operatorname{argmin}} \{a(z_i, C_l)\}$$

indicates the nearest Cluster to z_i. The Silhouette score is then defined as

$$s(z_i) = \frac{a(z_i, C_m) - a(z_i, C_k)}{\max\{a(z_i, C_m), a(z_i, C_k)\}} \, . \tag{4}$$

The Silhouette score is advantageous compared to other metrics as it quantifies the separation of clusters by assessing both cohesion (how close instances within the same cluster are) and separation (how far apart instances in different clusters are). Many other metrics focus exclusively on one aspect, resulting in an incomplete assessment of clustering quality. The Silhouette score is similar to the MMD-Critic Score [16] when using a kernel function such as the Euclidean distance in the MMD-Critic. Both assess the two mentioned aspects, though they have different scaling effects. The clustering of data points in the latent space is optimized by minimizing the average Silhouette score over all data points z_i using k-means clustering. This clustering will be used as baseline for assessing the prototypes. The centroid of cluster C_i is calculated via

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \, , \quad \text{where } C_i \in C \, . \tag{5}$$

The set of centroids can then be expressed as

$$\mathcal{M} = \{\mu_1, \mu_2, \ldots, \mu_K\} \quad \text{with } |\mathcal{M}| = |C| = K \, .$$

These clusters and centroids are used to assess the prototype quality as described in the following.

3.2 Metrics

The Co-12 properties by Nauta et al. [28] are categorized into three main dimensions: Content, Presentation, and User. However, the properties lack practical usage for accessing prototype explanations. Therefore, our framework extends these conceptual attributes by providing applicable metrics to assess prototype quality as described in the following subsections. In our ProtoScore benchmark, a constrained set of those properties is used, as listed in Table 1. Since our benchmark focuses on testing the technical properties of prototypes, we excluded criteria that are not assessed automatically and are better suited for user studies. In total, seven of the Co-12 properties are evaluated. Two of the five properties not considered are inherently satisfied by design or identical across the prototype methods (Completeness, Composition), and one is not applicable (Controllability) for prototype models, eliminating the need for their evaluation. The remaining two properties (Context, Coherence) are not automatically assessable and thus, user studies are needed. This strategic exclusion of certain properties is grounded in their specific nature and context, thus the excluded properties and their exclusion are described in further detail in the following.

Completeness measures how well the explanation captures the model's decision-making process. It consists of Reasoning-Completeness and Output-Completeness. Reasoning-Completeness indicates the degree to which the explanation describes the entire internal workings of the model. Reasoning-Completeness is inherently met by ante-hoc prototype models, as they provide an explanation that is equivalent to the model itself. Output-Completeness focuses on how much of the output is explained within the explanation, ensuring that it sufficiently explains the prediction made by the model. Output-Completeness is also inherently satisfied by design in non-part prototype models. Composition evaluates the presentation format and organization of explanations, focusing on how information is conveyed rather than the content itself. Since we only evaluate one explanation type—prototypes—and the design choices for displaying prototypes are independent of the prototype method, this aspect is not included in the benchmark. Controllability refers to users' ability to interact with and modify explanations. Since current prototype models are typically static and user-independent, no meaningful evaluation of this property is possible. Context considers the user's needs, expertise, and specific application, emphasizing the importance of tailoring explanations to different users. This property cannot be evaluated technically. Instead, user studies can provide qualitative feedback on the relevance and effectiveness of explanations in real-world scenarios. Coherence assesses the alignment of explanations with users' prior knowledge and beliefs, ensuring plausibility and reasonableness. This property cannot be evaluated automatically, but visualization of prototypes can help determine their alignment with common understanding.

We define two additional criteria to be incorporated into the benchmark, since the Co-12 insufficiently capture all aspects of prototypes, particularly concerning the latent space quality and representativeness of the prototypes, which are properties specific to prototype methods. Each criterion is evaluated using metrics that are adopted for prototypes. All metrics provide values from 0 to 1, where 0 indicates poor prototype quality and 1 represents the highest quality. The evaluated criteria, as listed in Table 1, with our defined metrics, will be presented in the following.

3.2.1 Correctness. Correctness measures how accurately explanations reflect the model's reasoning and predictions. In prototype models, Correctness is linked to the model's ability to explain decisions through visualizations of learned prototypes. Correctness is

Table 1: Criteria and metrics used in the benchmark tool.

Co-12 Property		Criterion	New Defined Metric
Co-12	Content	**Correctness**	Fidelity
		Consistency	Difference of prototypes between runs
		Continuity	Stability of explanations under input perturbations
		Contrastivity	Distance of prototypes
		Covariate Complexity	Silhouette score of prototypes
	Presentation	**Compactness**	Number of prototypes
		Confidence	Distance of sample to corresponding prototype
New		**Input Completeness**	Ratio of representative prototypes to centroids
		Cohesion of Latent Space	Silhouette score of clusters

often evaluated using synthetic datasets where the true prototypes are known in advance. Measures for the Correctness of prototypes then include the number of found prototypes and the distance between found and true prototypes. However, since this type of evaluation compares the explanation and the dataset, this evaluation can yield low Correctness scores if the model is not well-adapted to the data, despite the explanations aligning with the model's reasoning, i.e., being correct. An alternative evaluation method for Correctness, which is used here, measures the agreement between the output of the predictive model for an input sample and the output of the predictive model for the corresponding prototype, also known as *fidelity*. For ante-hoc prototype methods, the fidelity only depends on the autoencoder's reconstruction loss, otherwise the extracted prototype reflects exactly the models behavior. Thus, to rate the Correctness for the set of clusters C in the latent space, we extract all data points x_i from each cluster C_j along with the corresponding prototypes p_j, and transform them into the input space using the given decoder defined by Equation (3). The prediction for the data points and input space prototypes can then be computed using the encoder and classifier defined in Equation (1) and Equation (2). Correctness is defined as the ratio of all matching predictions between the data points y_i and their respective prototype y_{p_i} to the total number of data points N,

$$\mathrm{CR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}_{\{y_{p_i}\}}(y_i)\,,$$

where $\mathbb{I}_{\{y_{p_i}\}}(y_i)$ is the indicator function being one if the condition $y_{p_i} = y_i$ holds, otherwise it is zero.

3.2.2 Consistency. Consistency ensures that identical inputs produce the same explanation, promoting stability and reliability across instances and runs. It emphasizes implementation invariance, so that explanations should not vary with specific model details. Two models generating the same output for all inputs should provide the same explanations, regardless of their architectures. Since prototype models are typically deterministic, the same input should yield the same explanation. However, nondeterminism can arise from factors such as model initialization and random seeds. Evaluating latent distances of prototypes across different runs can indicate the stability of the model's explanations under varying conditions. Consequently, we employ additional models trained under the same conditions as the model being tested. Prototypes from these models are decoded into the input space using Equation (3) with their respective decoder functions and encoded into the latent space of the model under evaluation using Equation (1) to allow for comparison. The Euclidean distances between the prototypes of the new models p_{new} and the baseline model p_{base} are calculated and averaged over all prototypes M and all new models N_R. The Consistency score is then given by the inverted normalized mean distance

$$\mathrm{CS} = \exp\left(-\frac{1}{M \cdot N_R}\sum_{i=1}^{M}\sum_{j=1}^{N_R} \|p_{\text{base}_i} - p_{\text{new}_{i,j}}\|_2\right).$$

This score ranges from 0 to 1, with $p_{\text{new}_{i,j}}$ represented in the latent space of p_{base_i} and corresponding to the same cluster C_i.

3.2.3 Continuity. Continuity measures the stability and generalizability of the explanation function, emphasizing that similar inputs should produce similar explanations. This property focuses on the model's ability to maintain consistent explanations even with slight input perturbations. We evaluate Continuity by introducing small Gaussian noise to the input dataset $\mathcal{D}$, resulting in a noisy dataset

$$\mathcal{D}_{\text{noised}} = x'_1, x'_2, \ldots, x'_N$$
$$\text{where } x'_i = x_i + \epsilon_i \text{ with } x_i \in D, \epsilon_i \sim \mathcal{N}(0, \sigma^2 I)\,.$$

Here, σ is set to 5 % of the average range of sample values and I is the identity matrix. Both $\mathcal{D}$ and $\mathcal{D}_{\text{noised}}$ are encoded by the encoder function (1). For each data point x_i or noisy data point $x_i\prime$ in the original dataset, we identify its closest prototype of the prototype set $\mathcal{P}$ by

$$p_{x_i} = \arg\min_{p_j \in \mathcal{P}} \|f(x_i) - p_j\|_2\,. \tag{6}$$

We assess the model's prototype stability to input perturbation by computing the average distance between prototypes of the original and noised datasets, normalizing and inverting this distance to obtain the Continuity score

$$\mathrm{CN} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \|p_{x_i} - p_{x'_i}\|_2\right) \quad \text{with } x \in \mathcal{D},\ x' \in \mathcal{D}_{\text{noised}}\,.$$

3.2.4 Contrastivity. Contrastivity measures an explanation's ability to distinguish between different classes or targets. In ante-hoc prototype models, this feature is integrated into the design, where different classifications reflect distinct reasoning processes and corresponding explanations/prototypes. Thus, the distance between different prototype representations indicates their discrimination

strength. We calculate Contrastivity as the mean distances between prototypes in the latent space using the Euclidean distance according to

$$\mathrm{CT} = \frac{1}{M(M-1)} \sum_{p_i, p_j \in P, i \neq j} \|p_i - p_j\|_2 \ .$$

3.2.5 Covariate Complexity. Covariate Complexity refers to the difficulty users may have in understanding an explanation, i.e., the shown prototypes. The complexity or understandability of these visual features can be quantified through user studies that evaluate human interpretation. The evaluation of Covariate Complexity often involves measuring the "purity" of the clusters with association to the closest prototype. A high purity score indicates that the explanations highly resemble a particular concept. Thus, we can assess this property automatically by calculating the Silhouette score as defined in Equation (4), here calculated for the prototypes included as elements to their respective cluster set. The Silhouette score rates if the prototypes are close to the data points that encode a similar concept, but are far away from data points with different concepts. While automated purity metrics are valuable, they should be complemented by user studies to ensure that the prototypes accurately reflect human understanding.

3.2.6 Compactness. Compactness refers to the size of the explanation provided by a model, emphasizing that explanations should not overwhelm the user. It is evaluated by counting the number of prototypes. The compactness is calculated by taking the total number of prototypes M and normalizing it as

$$\mathrm{CP} = \exp\left((-M+1) \cdot a_{\text{normalize}}\right) ,$$

where $a_{\text{normalize}} = 0.08$ is a normalization hyperparameter that ensures the $0-1$ range for typical numbers of prototypes. The factor is chosen to balance the score, resulting in a mediocre score of 0.5 for ten prototypes, with fewer prototypes yielding better scores and more prototypes leading to lower scores.

3.2.7 Confidence. Confidence refers to the degree of certainty a model has regarding its predictions and explanations. It encompasses the availability of probability information related to the model's output and the reliability of those probability estimates. It is typically expressed as a probability score reflecting how certain the model is about a classification. Confidence scores derived from models trained with softmax and cross-entropy loss can be assessed for their reliability. We distinguish between different types of confidence: classification confidence (related to the predicted class) and explanation confidence (related to the reliability of the explanation generation). For prototype methods—given that the classification is generated via the class of the nearest prototype—classification confidence and explanation confidence are the same. Confidence is measured by calculating the average distance of the data points in the latent space z_i to the closest prototype $p(x_i)$, where the closest prototype $p(x)$ can be found by means of Equation (6). Then the Confidence score is given by

$$\mathrm{CF} = \exp\left(-\frac{1}{N} \sum_{i=1}^{N} \|z_i - p_{x_i}\|_2\right) .$$

3.2.8 Input Completeness. Input Completeness is an extension of Reasoning Completeness and Output Completeness, measuring how well prototypes represent the training data. It assesses the proportion of training data points adequately captured by the prototypes, indicating the model's ability to generalize to unseen instances. A high Input Completeness value suggests that prototypes are well-distributed and align closely with the data distribution, while a low value may indicate underrepresented regions, leading to poor generalization. Input Completeness is assessed by the ratio of representative prototypes to the number of clusters. Therefore, a prototype p_i is considered representative of C_j if its distance to the cluster's centroid μ_j is smaller than the average intra-cluster distance $a(\mu_j, C_j)$. We define the set of clusters that have at least one representative prototype as

$$C_{\text{rep}} = C_j \in C \mid \exists p_i \in \mathcal{P}, \text{ such that } \|p_i - \mu_j\|_2 < a(\mu_j, C_j) \ , \quad (7)$$

where the average intra-cluster distance is calculated by the average Euclidean distance from every point in C_j to its centroid μ_j according to

$$a(\mu_j, C_j) = \frac{1}{|C_j|} \sum_{z \in C_j} \|\mu_j - z\|_2 \ .$$

Then, the Input Completeness score is given by

$$\mathrm{IC} = \frac{|C_{\text{rep}}|}{|C|} \ ,$$

which is the relative frequency of clusters that have a representative prototype, i.e., these clusters satisfy the acceptance criterion (7) and thus, comprise at least one prototype.

3.2.9 Cohesion of Latent Space. Not only the prototypes are evaluated but also the latent space is assessed because it is essential as the foundation for calculating the prototypes. The quality of the latent space is crucial as it directly impacts the representation of the data and the effectiveness of the prototypes derived from it. This criterion evaluates how well the latent space organizes the training data into distinct clusters. A well-structured latent space produces clusters that are easily separable from one another, allowing for clear differentiation between various categories or classes within the data. Effective clustering ensures that similar data points are grouped together, enhancing the model's ability to generalize and make accurate predictions on new data.

To quantify the clustering ability of the latent space, we use the Silhouette score defined in Equation (4). This metric evaluates how similar an object is to its own cluster compared to other clusters. A higher Silhouette score indicates better-defined clusters. The Cohesion of Latent Space (CLS) Score is defined as the average Silhouette score across all clusters C_i with centroid μ_i

$$\mathrm{CLS} = \frac{1}{|C|} \sum_{i=1}^{n} s(\mu_i) \quad \text{for } C_i \in C \text{ and } \mu_i \in M \ .$$

3.3 Implementation of the Benchmark Framework

The benchmark framework (cf. Algorithm 1) requires an encoder (1), a decoder (3), a trained classifier, a dataset with labels on which the autoencoder and prototype classifier network have been trained, and a list of learned prototypes. Prior to calculating

Algorithm 1 Benchmark Framework

Require: Encoder f, Decoder g, Prototype Classifier h, Dataset $\mathcal{D} = \{x_1, \cdots, x_N\}$, Labels, Prototypes $\mathcal{P} = \{p_1, p_2, \ldots, p_M\}$
Ensure: Prototype Quality Scores
1: Transform input data into latent space: $z_i = f(x_i)$ (see Equation (1))
2: Classify latent representations z_i by means of ground-truth labels
3: Cluster the data for each class using k-means by optimizing Silhouette score as defined in Equation (4)
4: Compute cluster centroids according to Equation (5)
5: Assign prototypes to each cluster resp. data point using Equation (6)
6: Calculate all prototype quality metrics listed in Table 1
7: Compute a total score by averaging the individual scores with equal weighting

the metrics for the specified criteria, several computations are required, as detailed in Section 3.1. The input data is transformed into a representation within a latent space using the provided encoder. Subsequently, the data points are categorized according to their ground-truth labels. For each class, clustering is performed through minimizing the Silhouette score defined in Equation (4) and described in Section 3.1 to identify the most appropriate number of clusters and the clusters themselves, respectively, using k-means. K-means is tested for $2-15$ clusters with the best performing number used, based on the Silhouette score. For each cluster, a centroid is computed as shown in Equation (5), and prototypes are mapped according to Equation (6). Finally, the metrics outlined in Section 3.2 are computed to evaluate each property individually, culminating in the calculation of an overall score that assesses the quality of the explanation. The complete implementation is available on GitHub[1].

4 Experiments

In this section we present an exemplary use case to demonstrate the usage of ProtoScore and additionally a more detailed investigation of prototype methods on multiple datasets. The description of the used datasets and prototype methods is included in Appendix B.

4.1 Exemplary Use Case

Users can rate the quality of the explanation of their prototype models using ProtoScore. For this, they can either use their own dataset or one already provided in ProtoScore. They can choose different prototype methods, train models with the selected dataset and run the benchmark. To use ProtoScore effectively, users should benchmark multiple models to obtain a diverse range of results. As an example, the results for two different time series prototype methods *MAP* (Model-Agnostic Prototype) [29] and *MSP* (Model-Specific Prototype) [11] are shown in Table 2. These methods, detailed in Appendix B.2, were trained on the ECG200 dataset [30], which is described in Section B along with other datasets that are used in the following sections. An exemplary illustration of the resulting prototypes for the MSP model with index 5 is given in Figure 2. Multiple models are trained for each method, as indicated by the

[1]https://github.com/HelenaM23/ProtoScore

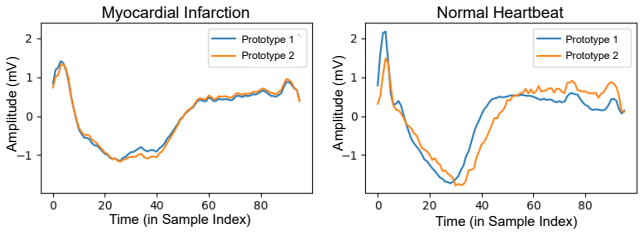


Figure 2: Exemplary prototypes for the MSP model with index 5. On the left, the prototype representing the abnormal class is displayed, while on the right, the prototype for the normal class, where specific conditions were detected, is shown.

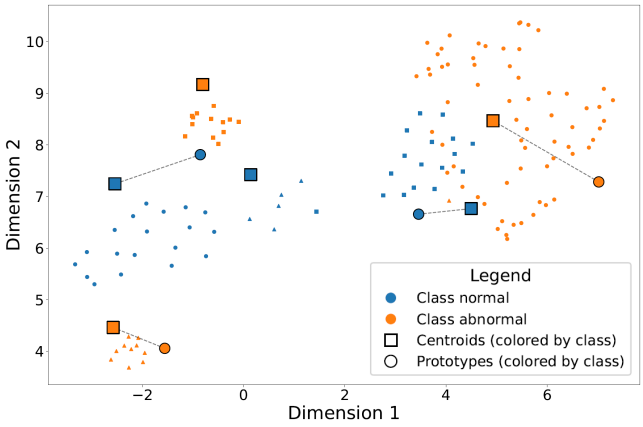


Figure 3: Dimension-reduced representation of the latent space for the MAP model with index 1. Note that the dimensionality reduction distorts distances, causing centroids and prototypes to appear outside their respective clusters. Small dots represent input data, with lines connecting each prototype (circle) to its corresponding cluster centroid (square). Points are color-coded by class.

index in the table, and evaluated against the prototype quality properties outlined in Section 3. The hyperparameters can be found in Appendix B.3. The total score averages the individual scores with equal weighting to summarize the prototype quality. Alongside, the mean squared error (MSE) validation loss for each model is given, which provides context for the model performance. The methods are sorted by descending validation loss. Note that a score of 0 indicates poor quality of the assessed prototypes, and a score of 1 represents the highest quality possible. The Compactness (CP) is consistent across all models, as four prototypes are always calculated. This number can be adjusted by the user. It is set to four in this case based on experience suggesting that this quantity strikes an optimal balance between compactness and representativeness of the prototypes for the dataset.

Figure 3 illustrates an exemplary latent space of the MAP model with index 1 (refer to Table 2). The latent space dimension is reduced using UMAP [22], a fast and scalable dimension reduction technique that preserves both local and global data structures for visualization, based on the assumption that the data lies on a locally

Table 2: Benchmark results for different models using the time series prototype methods MAP and MSP with the ECG200 dataset. The abbreviations in the column headings are the tested properties: (CR) Correctness, (CS) Consistency, (CN) Continuity, (CT) Contrastivity, (CC) Covariate Complexity, (CP) Compactness, (CF) Confidence, (IC) Input Completeness, (CLS) Cohesion of Latent Space.

Index	Dataset	Method	Val Loss	CR	CS	CN	CT	CC	CP	CF	IC	CLS	Total
1	ECG200	MAP	MSE: 1.45	0.69	0.28	0.98	0.43	0.68	0.79	0.67	0.67	0.63	0.65
2	ECG200	MAP	MSE: 0.70	0.78	0.34	0.98	0.37	0.66	0.79	0.68	0.80	0.64	0.67
3	ECG200	MAP	MSE: 0.55	0.85	0.40	1.00	0.41	0.65	0.79	0.69	0.67	0.66	0.68
4	ECG200	MAP	MSE: 0.38	0.88	0.46	1.00	0.30	0.69	0.79	0.61	0.57	0.66	0.66
5	ECG200	MAP	MSE: 0.26	0.88	0.46	0.97	0.30	0.69	0.79	0.60	0.57	0.64	0.66
1	ECG200	MSP	MSE: 1.04	1.00	0.37	1.00	0.49	0.50	0.79	0.55	0.00	0.60	0.59
2	ECG200	MSP	MSE: 0.78	1.00	0.45	1.00	0.46	0.50	0.79	0.43	0.00	0.52	0.57
3	ECG200	MSP	MSE: 0.36	1.00	0.55	0.95	0.37	0.53	0.79	0.47	0.00	0.39	0.56
4	ECG200	MSP	MSE: 0.30	0.86	0.60	0.86	0.33	0.74	0.79	0.66	0.20	0.73	0.64
5	ECG200	MSP	MSE: 0.28	0.96	0.60	0.83	0.38	0.61	0.79	0.57	0.00	0.73	0.61

connected Riemannian manifold. Note that this reduction distorts distances, rendering them incomparable in the plot. Thus, cluster centroids and prototypes may appear to be outside their respective clusters, even though they are actually part of them. Analyzing the latent space reveals a strong correlation with the scores achieved by MAP's model 1 in Table 2. The overlap between the orange and blue clusters on the right side suggests that the challenges with Correctness (CR), Confidence (CF), and Cohesion of the Latent Space (CLS) may stem from this area. However, overall, the prototypes and clusters are well-separated, indicating strong performance in Contrastivity (CT), Covariate Complexity (CC), and Confidence (CF). Also the representativeness of the prototypes can be seen. Six clusters are identified, with four featuring representative prototypes, as indicated by the connecting lines. This finding agrees with the Input Completeness score (IC) in the table.

Comparing the results in the table reveals that MAP achieves higher total scores across different validation losses compared to MSP. MAP's total score lies in the range of $0.65 - 0.68$, whereas MSP's total score is is between 0.56 and 0.64. This indicates MAP's greater effectiveness in optimizing prototype quality, especially at lower validation loss values. Notably, MAP scores perform well in terms of properties such as Correctness (CR) ($0.69 - 0.88$) and Continuity (CN) ($0.98 - 1.00$), which are essential to ensure that the generated prototypes explain the model's behavior correctly and are continuous. As MAP models improve in predictive accuracy (lower validation loss), Correctness (CR) and Consistency (CS) improve but Contrastivity (CT) decreases. This could be due to the also decreasing reconstruction loss but closer inspection reveals that the prototypes tend to be located closer together. The high total scores for MAP across all models suggest that it is a reliable method for generating high-quality prototypes. Even at higher validation loss values, MAP maintains a relatively stable total score, indicating consistent performance in prototype quality across various validation loss levels. The MSP method shows a more varied performance profile. While its total scores are generally lower than MAP's, MSP excels in specific properties for certain models. For example, MSP achieves perfect scores of 1.00 in Correctness (CR) for model 1, model 2 and model 3. These high scores highlight MSP's ability to ensure that its prototypes align closely with the model's behavior. Moreover, MSP has two models with the highest Cohesion of Latent Space score (CLS) (0.73 for model 4 and model 5) and highest Consistency (CS) (0.60 for model 4 and model 5), indicating its ability to give a basis for calculating well separable prototypes and consistent prototypes over multiple runs. However, MSP exhibits weaknesses in other properties, such as Covariate Complexity (CC), Confidence (CF), Continuity (CN) and Input Completeness (IC), contributing to its overall lower total scores compared to MAP. The Input Completeness (IC) score of MSP is almost always 0.00 for the ECG200 dataset because the distances from the prototypes to the respective cluster centroids are larger than the cluster spreads, making the prototypes unrepresentative for their respective clusters.

Overall, the results underscore the importance of considering multiple evaluation criteria when benchmarking prototype methods. While MAP outperforms MSP in the total score and is superior in most of the individual properties across different classification performances, MSP demonstrates strengths in certain properties that might be critical for specific applications, such as those prioritizing high Correctness and Consistency. Additionally, the inclusion of the validation loss in the evaluation provides an important context to interpret the results. A method with high prototype quality but high validation loss might indicate that its prototypes are well-constructed, but fail to generalize effectively in predictive tasks. Conversely, a method with low validation loss and good prototype quality represents a more balanced approach suitable for applications. Users aiming to apply ProtoScore effectively should carefully analyze both the total scores and individual property scores in relation to their specific objectives. For example, if model interpretability by less complex prototypes and high confidence about the allocation of prototypes to data points are priorities, MAP's higher total scores and balanced property performance make it the preferable choice. Conversely, for tasks where Correctness and Consistency are paramount, MSP's higher scores in CR and CS may offer advantages.

4.2 Additional Experiments on More Datasets

In addition to choosing between different prototype methods for a specific use case, ProtoScore can be used as a general benchmark for prototype-based XAI methods across different datasets. Corresponding results are listed in detail in Appendix B.4 in Table 4. While our analysis focuses on univariate time series data, where x_i represents a time series, ProtoScore is also applicable to multivariate datasets (x_i represents all time series that belong to a data instance) and image data (x_i represents an image), since our defined metrics apply across different data types. Analyzing datasets like Wafer, SAWSINE, ECG200, FordA, FordB, and StarLightCurve highlights the applicability to both binary and multiclass classification. Evaluating MAP and MSP prototypes with our benchmark tool provides valuable insights. It reveals that MSP exhibits greater performance variability across different datasets, as evidenced in the ECG200 dataset. MSP achieves high property scores in specific instances but generally lower overall scores than MAP. MAP often achieves higher Correctness, Input Completeness and sometimes Continuity, while MSP excels in properties such as Cohesion of the Latent Space and sometimes in Confidence and Covariate Complexity. This emphasizes the importance of selecting methods based on the specific properties most relevant to the application.

Analyzing trends in loss values and their relationship to explanation quality reveals valuable insights. Generally, one might expect that as the validation loss decreases, the quality of explanations improves across the criteria. However, our results demonstrate that this relationship is not always consistent. For instance, while the validation loss for the MAP method on the ECG200 dataset in Table 2 increases, certain properties such as Correctness and Consistency improve, while Contrastivity decreases. Moreover, datasets such as FordB for MSP show differences in prototype scores for similar loss values, highlighting that explanation quality can differ along different training runs even if it results in the same loss. This indicates that improvements in model performance are not always associated with better prototype explanations. In contrast, some models can optimize specific aspects of explanation quality such as Continuity and Covariate Complexity without necessarily achieving the lowest loss. These observations suggest that ProtoScore provides more detailed insights than relying solely on loss metrics alone. While loss values are useful for evaluating overall reconstruction or prediction performance, they fail to capture the nuances of explanation quality. ProtoScore provides enhanced insights by examining multiple criteria that reflect different dimensions of interpretability. This enables a more comprehensive evaluation, revealing balances and inconsistencies that loss metrics alone may overlook.

Notably, trade-offs exist between certain properties as models are optimized, which can be seen in Appendix B.4 Table 4. For example, properties such as Continuity and Contrastivity tend to correlate. High Contrastivity, which implies significant distances between prototypes, reduces the likelihood of shifts in prototype assignments when input noise is introduced. Additionally, Input-Completness is often accompanied by Confidence. This is evidenced by shorter distances between data points and their respective prototypes when the prototypes are situated near their clusters. On the other hand, Cohesion of Latent Space can influence Input Completeness negatively (as shown in the Wafer dataset using MAP). A dense latent space hinders a prototype's ability to represent clusters due to stricter acceptance thresholds based on intra-cluster distances. These observations underscore the nuanced interplay between metrics, highlighting trade-offs and complementarities that can guide explanation refinement efforts.

4.3 Outlier Analysis

This outlier analysis explores the impact of introducing outliers to the SAWSINE dataset, where 3% of the data points are modified by adding noise. The focus is on understanding how these alterations affect the metrics. The results are shown in Table 5 in Appendix B.4 for the unmodified SAWSINE dataset, the evaluation of the modified dataset with original models, and new models trained on the modified dataset. Plot 4 in Appendix B.4 shows the latent space of the MAP model with index 3, evaluated on the dataset both with and without outliers. Through dimension reduction, the plot with outliers appears distorted compared to the original one. However, the four core clusters (two orange and two blue) remain visible, with several points outside the clusters and overlapping due to the altered perspective. Upon validating the original models with the outlier dataset, Correctness decreases by up to 3 percentage points, due to increased prediction difficulty from noise disrupting natural data patterns. Metrics like Consistency, Contrastivity, and Compactness remain unchanged due to the lack of different training scenarios. Continuity decreases by up to 3 percentage points, reflecting challenges in maintaining smooth prediction transitions with outliers. Confidence worsens by a maximum of 3 percentage points, and Input Completeness declines for one MSP model. Covariate Complexity remains stable for MAP models but varies for MSP models, with some improving and others declining. This corresponds to variation in the Cohesion of Latent Space for MSP models. For MAP models, the Cohesion of Latent Space declines due to data points lying outside clusters.

When validating newly trained models using the outlier dataset, while hyperparameters were selected as before for models without outliers, Correctness generally decreased for MAP models but remained stable for MSP models. Consistency improved slightly compared to mixed validation, though MSP showed significant variability across different models, as seen in other datasets. Continuity worsened for MAP models but was similar to mixed models, with a slight improvement for MSP models. Contrastivity generally improved, while Covariate Complexity and Confidence declined. Input Completeness mostly remained unchanged across models, and latent space cohesion deteriorated for MAP models but varied for MSP models.

Overall, the variability in model performance, influenced by training conditions, is evident, with MSP models exhibiting more variability than MAP models. This is consistent with previous findings across various datasets, emphasizing the diverse training influences on MSP models and the observed variations in multiple models for the same dataset. The outlier analysis results supports the benchmark tool's correct and effective functioning, as it responds in general predictably when outliers are introduced into the dataset. The changes in scores are expected when certain points in the dataset are more distant from distinct clusters.

5 Discussion

5.1 Reproducibility

A key advantage of ProtoScore is its reproducibility, as most metrics are deterministic, ensuring consistent benchmark results across runs. Only the Continuity metric fluctuates due to the introduction of stochastic noise. In theory, with an infinite number of samples, Continuity converges. However, with the sample number used in this benchmark, only minor fluctuations are observed, leading to scores that lie within a range of minor uncertainties.

5.2 Challenges for Interpretation

ProtoScore can be challenging to interpret for underfitted models due to clustering issues that affect the calculated metrics. Underfitted models struggle to accurately capture underlying patterns and to correctly display the data in the latent space in distinct clusters. This is reflected in a low Cohesion of Latent Space score but is not always displayed in the other criteria or even leads to misinterpretable improvements in these criteria. For example, the dependence on clustering can obscure insights into Input Completeness, as the threshold for a representative prototype shrinks. For underfitted models, some scores lead to confusion about the model's overall performance, and thus each criterion must be interpreted in the context of all other criteria to avoid misinterpretations.

Total scores provide a useful summary, but can obscure weaknesses in individual properties. For example, a high total score might mask deficiencies in critical properties such as Correctness that are critical for certain applications. Certain properties affect the behavior of other properties, as described in Section 4, leading to potential misinterpretations. Understanding these correlations and their underlying principles is vital, as each metric provides complementary insights into explanation quality. Careful consideration must be taken to ensure that these metrics are interpreted in context, particularly when designing or refining models to optimize multiple criteria simultaneously. Moreover, the importance of each property varies across use cases. For instance, in domains such as healthcare or finance, Correctness and Consistency are crucial. In contrast, exploratory tasks may prioritize Contrastivity for better visible class distinction. Additionally, results from one dataset might not generalize to others with different characteristics. Properties such as Consistency and Continuity can behave differently based on dataset complexity and variability.

5.3 Supporting Model Selection

Benchmarking multiple models, as discussed in Section 4, offers valuable insights into the trade-offs between different methods. The comparison of MAP and MSP reveals similar results across various datasets, highlighting the strengths and weaknesses of each approach. Thus, ProtoScore enables users to select the method that best aligns with their priorities. However, caution is necessary, as several factors can lead to misinterpretations of benchmark results.

The findings in Section 4 emphasize using ProtoScore as a comprehensive evaluation tool rather than relying solely on loss values or a single criterion. Analyzing the interplay between different properties and considering dataset-specific and method-specific behaviors enables users to make well-informed decisions about model selection and optimization strategies.

5.4 Quantitative Evaluation Metrics

This paper underscores the significance of evaluating explainability methods using quantitative metrics, offering an objective approach that reduces reliance on human evaluation. However, developing a comprehensive computational benchmark applicable across all methods poses significant challenges. Thus, we focus on prototype-based methods and use the Co-12 properties as evaluation criteria. While these criteria can be applied to other methods, direct comparisons to our prototype metrics are difficult due to XAI approach specific definitions, which complicate assessments even when targeting the same Co-12 property. Moreover, interpretability remains an inherently subjective concept, influenced by the context of explanations and the backgrounds of both providers and receivers. We advocate for quantitative metrics to enhance explanations before testing them in user studies. As our understanding of human interpretability in XAI progresses, these metrics may replace qualitative assessments. As noted by Veerappa et al. [42], XAI evaluation should be an interdisciplinary effort that integrates both human and computational elements. Our library is designed to expedite the processes of development, refinement, and failure identification prior to conducting user studies.

6 Conclusion and Future Work

In conclusion, this research establishes a crucial framework for evaluating prototype-based eXplainable Artificial Intelligence (XAI) methods, particularly in the context of different data types. By integrating the Co-12 properties in an applicable and quantitative way, our proposed ProtoScore framework provides a comprehensive metric system that allows for effective and automated assessment of prototype quality across various data types. Through practical applications and detailed analyses, we demonstrate how ProtoScore can guide practitioners in selecting appropriate XAI methods, while minimizing the costs associated with user studies. Our findings underscore the importance of reliable and reproducible evaluation methods to enhance the use of explainability methods.

Future work should aim for refining these metrics and be completed by user-centric evaluations to enhance the practical applicability of XAI methods. It is crucial to extend metrics to include not only latent space considerations, but also further parts of the model, and develop a more comprehensive evaluation framework that connects quantitative metrics with human-centric interpretability assessments. With regard to synthetic data, while such datasets can serve as additional resources to assess the plausibility of explanations, they should not replace real-world tests. Given our frequent lack of knowledge about the ground truth generative process for real datasets, integrating synthetic datasets with known clusters and prototypes could facilitate future analyses. However, this integration must balance the realism of the data with the availability of ground truth. Another enhancement could include dynamic weighting of properties based on user-defined priorities, improved visualizations of property interdependencies, and mechanisms to adaptively optimize models for specific property combinations. Moreover, we neglected time series-specific metrics and only used

the Euclidean distance since our goal for the framework is to be versatile across different data types, including both time series and image data. A valuable extension of our framework is to allow users to choose from a range of metrics tailored to their specific data type and requirements. Since the best total score in our analysis was 0.76 using MAP on the SAWSINE and Wafer dataset, there is also a need for further development across prototype methods to achieve a better prototype quality.

Acknowledgments

Certain sections of this paper were developed using a company-specific version of ChatGPT (4o mini). It assisted in generating LaTeX code for tables and enhancing the drafts of specific sections. We thank our colleagues and the reviewers of the FAccT 2025 conference, who helped to improve this paper with their valuable feedback. This paper is funded in parts by the German Research Foundation (DFG) Priority Programme (SPP 2422) "Data-Driven Process Modelling in Forming Technology" (Subproject "Transparent AI-supported process modeling in drop forging" 520195047), by the Fraunhofer Gesellschaft under grant no. PREPARE 40-02702 (project "ML4Safety") and by the Federal Ministry for Economic Affairs and Climate Action under the Industrial Collective Research (IGF) program on the basis of a decision by the German Bundestag (IGF project 22929 BG of the Federation of Quality Research and Science (FQS) "AIQualify - Framework for Qualifying AI Systems in Industrial Quality Inspection").

References

[1] Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. 2024. OpenXAI: towards a transparent evaluation of post hoc model explanations. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '22)*. Curran Associates Inc., Red Hook, NY, USA, Article 1148, 16 pages. https://dl.acm.org/doi/10.5555/3600270.3601418

[2] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilovic, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2021. AI Explainability 360: Impact and Design. arXiv:2109.12151 https://arxiv.org/abs/2109.12151

[3] Anthony Bagnall. 2008. FordA Data Set. https://timeseriesclassification.com/description.php?Dataset=Forda Accessed: 2024-10-20.

[4] Anthony Bagnall. 2008. FordB Data Set. https://timeseriesclassification.com/description.php?Dataset=FordB Accessed: 2024-10-20.

[5] Mohamed Karim Belaid, Eyke Hüllermeier, Maximilian Rabus, and Ralf Krestel. 2022. Do We Need Another Explainable AI Method? Toward Unifying Post-hoc XAI Evaluation Methods into an Interactive and Multi-dimensional Benchmark. arXiv:2207.14160 https://arxiv.org/abs/2207.14160

[6] CJEU. 07.12.2023. SCHUFA Holding (Scoring). Judgement, C-634/21, ECLI EU:C:2023:957.

[7] CJEU. 27.02.2025. Dun & Bradstreet Austria. Judgement, C-203/22, ECLI EU:C:2025:117.

[8] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *International Journal of Human-Computer Studies* 58, 6 (2003), 697–718. doi:10.1016/S1071-5819(03)00038-7

[9] L. Wei E. Keogh. 2009. StarLightCurves. http://www.timeseriesclassification.com/description.php?Dataset=StarlightCurves Accessed: 2025-04-10.

[10] Henry Fraser, Rhyle Simcock, and Aaron J. Snoswell. 2022. AI Opacity and Explainability in Tort Litigation. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (Seoul, Republic of Korea) *(FAccT '22)*. Association for Computing Machinery, New York, NY, USA, 185–196. https://doi.org/10.1145/3531146.3533084

[11] Alan H Gee, Diego Garcia-Olano, Joydeep Ghosh, and David Paydarfar. 2019. Explaining Deep Classification of Time-Series Data with Learned Prototypes. *CEUR workshop proceedings* 2429 (2019), 15–22. http://dblp.uni-trier.de/db/conf/ijcai/khd2019.html#GeeGGP19

[12] Kazuaki Hanawa, Sho Yokoi, Satoshi Hara, and Kentaro Inui. 2021. Evaluation of Similarity-based Explanations.

[13] Peter Hase and Mohit Bansal. 2020. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior?. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 5540–5552. doi:10.18653/v1/2020.acl-main.491

[14] Anna Hedström, Leander Weber, Dilyara Bareeva, Daniel Krakowczyk, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. 2023. Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations and Beyond. *Journal of Machine Learning Research* 24, 34 (2023), 1–11. http://jmlr.org/papers/v24/22-0142.html

[15] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. 2019. Metrics for Explainable AI: Challenges and Prospects. arXiv:1812.04608 https://arxiv.org/abs/1812.04608

[16] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! Criticism for Interpretability. https://proceedings.neurips.cc/paper_files/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf

[17] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. arXiv:2009.07896 https://arxiv.org/abs/2009.07896

[18] Phuong Quynh Le, Meike Nauta, Van Bach Nguyen, Shreyasi Pathak, Jörg Schlötterer, and Christin Seifert. 2023. Benchmarking eXplainable AI - A Survey on Available Toolkits and Open Challenges. 6665–6673 pages. doi:10.24963/ijcai.2023/747

[19] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions. doi:10.1609/aaai.v32i1.11771

[20] Yang Liu, Sujay Khandagale, Sujay Khandagale, Colin White, and Willie Neiswanger. 2021. Synthetic Benchmarks for Scientific Research in Explainable Machine Learning. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* 1 (2021). https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/c16a5320fa475530d9583c34fd356ef5-Paper-round2.pdf

[21] John M. McGuirl and Nadine B. Sarter. 2006. Supporting Trust Calibration and the Effective Use of Decision Aids by Presenting Dynamic System Confidence Information. *Human Factors: The Journal of Human Factors and Ergonomics Society* 48 (2006), 656 – 665. doi:10.1518/001872006779166334

[22] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 https://arxiv.org/abs/1802.03426

[23] Sina Mohseni, Jeremy E Block, and Eric Ragan. 2021. Quantitative Evaluation of Machine Learning Explanations: A Human-Grounded Benchmark. In *Proceedings of the 26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21)*. Association for Computing Machinery, New York, NY, USA, 22–31. doi:10.1145/3397481.3450689

[24] Christoph Molnar. 2022. *Interpretable machine learning: A guide for making black box models explainable* (second edition ed.). Christoph Molnar, Munich, Germany. https://christophm.github.io/interpretable-ml-book/

[25] Davide Napolitano and Luca Cagliero. 2024. BONES: a benchmark for neural estimation of shapley values. 6 pages. doi:10.1109/AICT61888.2024.10740433

[26] Anushka Narayanan and Karianne Bergen. 2024. Prototype-Based Methods in Explainable AI and Emerging Opportunities in the Geosciences. doi:10.48550/arXiv.2410.19856

[27] Meike Nauta and Christin Seifert. 2023. The Co-12 Recipe for Evaluating Interpretable Part-Prototype Image Classifiers, In Explainable Artificial Intelligence, Luca Longo (Ed.). *Communications in Computer and Information Science Explainable Artificial Intelligence* 1901, 397–420. https://doi.org/10.1007/978-3-031-44064-9_21

[28] Meike Nauta, Jan Trienes, Shreyasi Pathak, Elisa Nguyen, Michelle Peters, Yasmin Schmitt, Jörg Schlötterer, Maurice van Keulen, and Christin Seifert. 2023. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *Comput. Surveys* 55, 13s, Article 295 (July 2023), 42 pages. doi:10.1145/3583558

[29] Christoph Obermair, Alexander Fuchs, Franz Pernkopf, Lukas Felsberger, Andrea Apollonio, and Daniel Wollmann. 2023. Example or Prototype? Learning Concept-Based Explanations in Time-Series. *Proceedings of The 14th Asian Conference on Machine Learning* 189 (12–14 Dec 2023), 816–831. https://proceedings.mlr.press/v189/obermair23a.html

[30] Robert Thomas Olszewski. 2001. ECG200 Data Set. https://www.timeseriesclassification.com/description.php?Dataset=ECG200 Accessed: 2024-10-20.

[31] Robert Thomas Olszewski. 2001. Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data.

[32] Robert Thomas Olszewski. 2001. Wafer Dataset - UCR Time Series Classification Repository. https://timeseriesclassification.com/description.php?Dataset=Wafer Accessed: 2024-10-20.

[33] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj

[34] European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council. http://data.europa.eu/eli/reg/2024/1689/oj

[35] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. doi:10.1016/0377-0427(87)90125-7

[36] Cynthia Rudin. 2019. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence* 1 (05 2019), 206–215. doi:10.1038/s42256-019-0048-x

[37] Sam Sattarzadeh, Mahesh Sudhakar, and Konstantinos N. Plataniotis. 2021. SVEA: A Small-scale Benchmark for Validating the Usability of Post-hoc Explainable AI Solutions in Image and Signal Recognition. 4141-4150 pages. doi:10.1109/ICCVW54120.2021.00462

[38] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. 2019. Towards A Rigorous Evaluation Of XAI Methods On Time Series . In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE Computer Society, Los Alamitos, CA, USA, 4197–4201. doi:10.1109/ICCVW.2019.00516

[39] Samuel Sithakoul, Sara Meftah, and Clément Feutry. 2024. BEExAI: Benchmark to Evaluate Explainable AI, In Explainable Artificial Intelligence, Luca Longo, Sebastian Lapuschkin, and Christin Seifert (Eds.). *Communications in Computer and Information Science* 2153, 445–468. https://doi.org/10.1007/978-3-031-63787-2_23

[40] Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. 2022. Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions. *IEEE Access* 10 (2022), 100700–100724. doi:10.1109/ACCESS.2022.3207765

[41] Sarthak Manas Tripathy, Ashish Chouhan, Marcel Dix, Arzam Kotriwala, Benjamin Klopper, and Ajinkya Prabhune. 2022. Explaining Anomalies in Industrial Multivariate Time-series Data with the help of eXplainable AI. 226–233 pages. doi:10.1109/BigComp54360.2022.00051

[42] Manjunatha Veerappa, Mathias Anneken, Nadia Burkart, and Marco F. Huber. 2022. Validation of XAI explanations for multivariate time series classification in the maritime domain. *Journal of Computational Science* 58 (2022), 101539. doi:10.1016/j.jocs.2021.101539

[43] Chong Wang, Yuyuan Liu, Yuanhong Chen, Fengbei Liu, Yu Tian, Davis J. McCarthy, Helen Frazer, and Gustavo Carneiro. 2023. Learning Support and Trivial Prototypes for Interpretable Image Classification. arXiv:2301.04011 https://arxiv.org/abs/2301.04011

[44] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y. Lim. 2019. Designing Theory-Driven User-Centric Explainable AI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–15. doi:10.1145/3290605.3300831

[45] Rosina O Weber, Adam J Johs, Prateek Goel, and João Marques Silva. 2024. XAI is in trouble. *AI Mag.* 45, 3 (July 2024), 300–316. doi:10.1002/aaai.12184

A Existing Evaluation Frameworks

Captum [17] offers two evaluation metrics—infidelity and sensitivity—across 22 attribution methods applicable to various data formats, including images, tabular data, and text. However, the primary focus of this library is on providing multiple implementations of XAI methods rather than establishing a comprehensive benchmark.

AIX360 [2] also addresses time series data, offering methods for both global and local explanations, as well as post-hoc, ante-hoc, and data-centered XAI approaches. Nonetheless, it includes only two evaluation metrics: faithfulness and monotonicity, and is primarily designed for XAI application rather than evaluation, excluding prototype-based methods. XAI-Bench [20] evaluates XAI algorithms based on five metrics: Faithfulness, Monotonicity, ROAR, GT-Shapley, and Infidelity, using synthetic data. This framework assesses six XAI methods: SHAP, LIME, MAPLE, SHAPR, L2X, and breakDown.

OpenXAI [1] builds upon the widely-used metrics from Quantus by introducing additional metrics to assess the fairness of explanations. The library also includes synthetic data generation tools, facilitating comprehensive ground truth comparisons, and features leaderboards for six selected tabular datasets, enabling comparative evaluations of Neural Network and Logistic Regression models. *Compare-xAI* [5] serves as a quantitative benchmark tool designed to evaluate and compare XAI algorithms based on functional testing methods. These tests are categorized into five groups: fidelity, fragility, stability, simplicity, and stress. The benchmark employs a hierarchical scoring system to provide a comprehensibility score, allowing users to evaluate algorithms based on their specific needs and levels of expertise. However, this tool is limited to XAI methods that provide feature importance (global explanations).

The *SVEA* benchmark [37] aims to evaluate various attribution methods, including Grad-CAM, Integrated Gradients, and RISE, across different metrics. It enables researchers to assess the usability of attribution methods based on properties such as soundness, Completeness, contextfulness, and actionability, without imposing high computational demands by using the small MNIST-1D dataset.

BEExAI [39] is proposed as a benchmark tool facilitating large-scale comparisons of different post-hoc XAI methods through a selected set of evaluation metrics. This library calculates nine evaluation metrics focused on three core properties of explainability: Faithfulness, Robustness, and Complexity, and is designed to evaluate eight explainability methods, including LIME, Shapley Values, and Integrated Gradients, but only on tabular datasets.

BONES [25] is a benchmark tool for neural estimation of Shapley Values. It provides researchers with a suite of state-of-the-art neural and traditional Shapley value estimators, such as KernelSHAP, FastSHAP, and DASP, along with a selection of commonly used benchmark datasets for tabular and image data. The library includes ante-hoc modules for train black-box models and specific functions for computing popular evaluation metrics, such as L1, L2 distances, and AUC for image data, as well as visualizing results. It also features functions for quantifying estimation errors, assessing computational costs, and comparing different explainers.

B Experiments

B.1 Used Datasets

The FordA dataset [3] is designed for fault detection in Ford's powertrain systems and contains sensor data collected from a Ford vehicle's powertrain system under various operational conditions. It is commonly used for time series classification tasks and contains 3,601 train instances and 1,320 test instances. Each time series corresponds to a measurement of engine noise captured by a motor sensor. For FordA the train and test dataset were collected in typical operating conditions, with minimal noise contamination.

The dataset FordB [4] is similar to FordA. This dataset contains 3,636 training instances and 810 test samples, each sample consists of 500 measurements of engine noise and a classification. For FordB the train data was collected in typical operating conditions, but the test data samples were collected under noisy conditions.

The Wafer dataset [32] is used for time series classification in semiconductor manufacturing, specifically for identifying conditions or faults in wafers. The dataset features two classes: normal and abnormal. The dataset contains 1,000 train instances and 6,164 test samples. However, there is a significant class imbalance, with only 10.7 % of the training data and 12.1 % of the test data classified as abnormal.

The ECG200 dataset [30] consists of measurements of cardiac electrical activity recorded from electrodes placed at various locations on the body. Each dataset contains recordings from a single electrode during one heartbeat. The ECG dataset comprises a total of 200 datasets, of which 133 are identified as normal and 67 as abnormal. This classification is crucial for detecting conditions such as bradycardia, which is characterized by a slower than normal heart rate. The dataset is part of a larger effort to leverage deep learning techniques to improve the detection and interpretation of critical cardiac events, particularly in high-risk populations such as preterm infants. Both the Wafer dataset and ECG200 dataset were formatted by Olszewski [31].

The SAWSINE dataset was created by Obermair et al. [29] to simulate signals from machine sensors operating in a noisy environment. This dataset comprises four basic time series shapes that serve as ground truth signals. To introduce realism and complexity, both multiplicative and additive noise were added to these signals. The noise levels varied, with amplitudes ranging from 0 to 1.1, and were drawn from a uniform distribution. The dataset we used has 8,000 samples divided into two equal classes with 4,000 and 4,000 data points.

The StarLightCurve dataset [9] contains a collection of phase-aligned starlight curves that represents the brightness of a celestial object (e.g., a star) as a function of time. The analysis of these curves is important in astronomy, particularly for studying the variability of celestial sources. Each starlight curve in this dataset has a length of 1,024 data points, capturing the intensity of light over time. The class labels (3 distinct classes) for each curve have been determined by an expert in the field. The dataset has a training set of 1,000 samples and a test set of 8,236 samples.

B.2 Used Prototype Methods

MAP (Model-Agnostic Prototype) [29] focuses on generating interpretable explanations for time series classification models using a model-agnostic approach based on prototypes derived from an autoencoder. The method employs an autoencoder architecture consisting of an encoder that maps input time series data to a latent space and a decoder that reconstructs the original data. Prototypes are derived from this latent space using k-means clustering, serving as abstract representations of concepts relevant to the data. The train objective optimizes for reconstruction accuracy while also including a similarity loss to ensure diversity among prototypes. This model-agnostic design enables the method to work independently of any specific classification model, allowing it to generate explanations for existing models without modification.

The prototype method *MSP* (Model-Specific Prototype) [11] is a model specific approach. It integrates an autoencoder architecture to enhance the interpretability of deep learning models applied to time series classification tasks. The model consists of an encoder that transforms input time series data into a lower-dimensional latent space, capturing essential features while minimizing information loss. The decoder then reconstructs the original data from this latent representation. To classify the data, the learned latent representations are fed into a prototype network that identifies and learns multiple prototype vectors corresponding to different classes. A significant advancement in this approach is the introduction of a prototype diversity penalty in the loss function. This penalty encourages the model to learn diverse prototypes, improving their distribution across the latent space and enhancing the model's ability to classify overlapping classes. The loss function balances several components, including classification loss, reconstruction loss, and the diversity penalty, ensuring that the learned prototypes are both informative and distinct.

B.3 Architecture and Hyperparameter Used in the Prototype Models

The hyperparameters and resulting loss values for the trained models are presented in this chapter in Table 3, offering insights into the configurations used during model training.

The architecture used for the MAP models includes an Encoder and Decoder, and a separate trained classifier. The Encoder comprises a Flatten layer that transforms the input data into a one-dimensional vector, making it suitable for processing by subsequent dense layers. It also includes a dense layer that outputs a latent representation with a specified dimension, as detailed in Table 3. This layer uses an L1 activity regularizer to encourage sparsity in the latent representation, which helps mitigate overfitting. The Decoder is responsible for reconstructing the original input from the latent representation generated by the Encoder. It consists of a dense layer with 300 units that initiates the reconstruction process, followed by another dense layer with 300 units that employs a sigmoid activation function. This function constrains the output values between 0 and 1. The final and third dense layer outputs a vector with dimensions equal to the product of the original input dimensions, effectively reconstructing the data to its original form. Lastly, the Classifier includes multiple convolutional blocks. Each block applies a 1D convolution with varying filter sizes: the first block employs 128 filters with a kernel size of 8, the second block uses 256 filters with a kernel size of 5, and the third block uses 128 filters with a kernel size of 3. Each convolutional layer is followed by batch normalization and ReLU activation to introduce non-linearity. After the convolutional layers, an average pooling layer condenses the feature maps into a single vector, which is then processed by a dense layer with softmax activation to provide class probabilities for classification tasks.

The architecture for the MSP models mirrors that of the MAP models, featuring the same three components. The Encoder is designed as a convolutional network to process input data, starting with the first convolutional block that applies a 1D convolution with 128 filters and a kernel size of 8, followed by batch normalization and ReLU activation. The second block uses 256 filters with a kernel size of 5, again followed by batch normalization and ReLU activation. The third block consists of 128 filters with a kernel size of 3, also followed by batch normalization and ReLU activation. A Global Average Pooling layer condenses the feature maps into a single vector, which is then passed through a dense layer with a specified latent dimension and ReLU activation. The Decoder for the MSP models reconstructs the original input from the latent representation, which includes a dense layer with 300 units, followed by another dense layer with 300 units using a sigmoid activation function to ensure output values fall within the range of 0 to 1. The final dense layer outputs a vector with dimensions equal to the product of the original dimensions, effectively reconstructing the input data. Additionally, the models feature a prototype layer containing a set of randomly initialized, trainable prototype representations, which are used for concept learning based on the latent dimension and number of concepts. The Classifier in the MSP models includes a dense output layer with softmax activation, providing class probabilities.

For both models, the learning rate for the autoencoder is set at 0.0001, while the classifier uses a learning rate of 0.001. The autoencoder employs mean squared error (MSE) as the loss function, whereas the classifier uses cross-entropy loss. The models are validated with a separated dataset using also the MSE.

B.4 Results for Experiments on All Datasets

Additional experiments were performed on the Wafer, SAWSINE, FordB, FordA, and StarLightCurves datasets to evaluate our benchmark framework, as presented in Table 4. The ECG200 results from Table 2 are also included for a better overview. These results evaluate the effectiveness of both the MAP and MSP methods across various metrics, which are further discussed in Section 4.

Table 5 presents the results of the outlier analysis. Models for the SAWSINE dataset are evaluated with outliers incorporated either in the evaluation dataset alone or in both the training and evaluation phases. These models are compared to the original models for the unmodified dataset. Figure 4 provides a visual comparison of exemplary latent spaces for the MAP model with index 5 evaluated with and without outliers.

Table 3: Hyperparameter and loss values used in the prototype models.

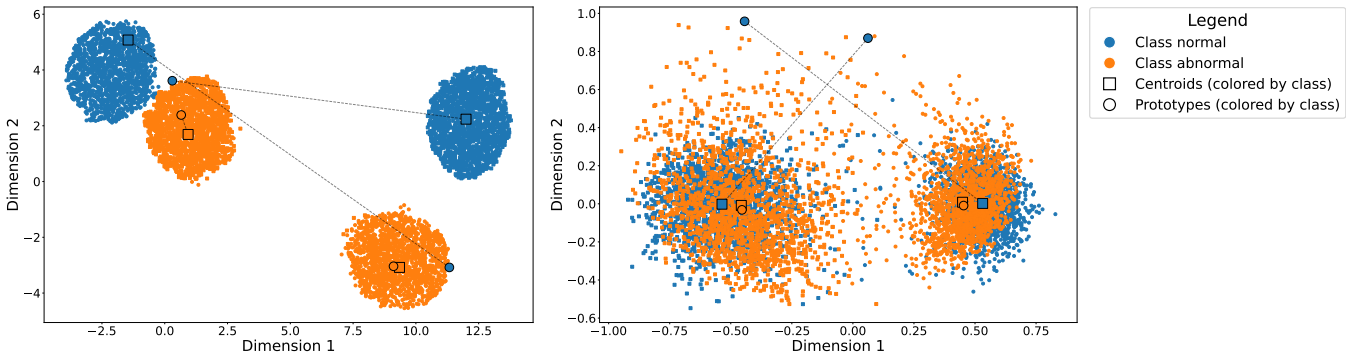
Index	Dataset	Method	Val Loss	Epochs	Loss Autoencoder	Loss Classifier	Latent Dimension Autoencoder
1	Wafer	MAP	1.26	1	1.52	0.06	20
2	Wafer	MAP	0.58	3	0.71	0.06	20
3	Wafer	MAP	0.27	15	0.55	0.06	20
4	Wafer	MAP	0.28	20	0.32	0.06	20
1	Wafer	MSP	0.86	1	0.04	0.37	4
2	Wafer	MSP	0.35	10	0.02	0.04	4
3	Wafer	MSP	0.35	15	0.02	0.02	4
4	Wafer	MSP	0.23	30	0.01	0.00	4
1	SAWSINE	MAP	0.37	1	0.49	0.01	20
2	SAWSINE	MAP	0.18	5	0.18	0.01	20
3	SAWSINE	MAP	0.17	10	0.17	0.01	20
4	SAWSINE	MAP	0.14	30	0.15	0.01	20
5	SAWSINE	MAP	0.13	50	0.13	0.01	20
1	SAWSINE	MSP	0.18	1	0.01	0.10	4
2	SAWSINE	MSP	0.14	10	0.01	0.00	4
3	SAWSINE	MSP	0.14	15	0.01	0.00	4
4	SAWSINE	MSP	0.14	30	0.01	0.00	4
1	FordB	MAP	1.02	1	1.11	0.52	20
2	FordB	MAP	0.83	10	0.90	0.52	20
3	FordB	MAP	0.62	30	0.72	0.52	20
4	FordB	MAP	0.49	50	0.60	0.52	20
1	FordB	MSP	1.00	10	0.05	0.42	4
2	FordB	MSP	1.00	30	0.05	0.17	4
3	FordB	MSP	1.00	50	0.05	0.22	4
4	FordB	MSP	1.00	100	0.05	0.09	4
1	FordA	MAP	1.16	1	1.23	0.22	20
2	FordA	MAP	1.05	5	1.02	0.22	20
3	FordA	MAP	0.86	10	0.86	0.22	20
4	FordA	MAP	0.70	30	0.70	0.22	20
5	FordA	MAP	0.58	50	0.56	0.22	20
1	FordA	MSP	1.00	10	0.05	0.51	4
2	FordA	MSP	1.00	30	0.05	0.31	4
3	FordA	MSP	1.00	100	0.05	0.15	4
4	FordA	MSP	0.99	150	0.05	0.14	4
1	ECG200	MAP	1.45	3	1.52	0.57	20
2	ECG200	MAP	0.70	30	0.71	0.57	20
3	ECG200	MAP	0.55	70	0.56	0.57	20
4	ECG200	MAP	0.38	200	0.40	0.57	20
5	ECG200	MAP	0.26	250	0.29	0.57	20
1	ECG200	MSP	1.04	1	0.06	0.64	4
2	ECG200	MSP	0.78	1	0.05	0.75	4
3	ECG200	MSP	0.36	10	0.02	1.48	4
4	ECG200	MSP	0.30	50	0.01	0.24	4
5	ECG200	MSP	0.29	200	0.01	0.12	4
1	StarLightCurve	MAP	1.07	1	1.10	1.30	30
2	StarLightCurve	MAP	0.76	5	0.79	0.29	30
3	StarLightCurve	MAP	0.55	10	0.57	0.27	30
4	StarLightCurve	MAP	0.31	30	0.32	0.27	30
5	StarLightCurve	MAP	0.13	50	0.14	0.27	30
1	StarLightCurve	MSP	0.42	1	0.03	0.89	6
2	StarLightCurve	MSP	0.28	5	0.01	0.50	6
3	StarLightCurve	MSP	0.22	20	0.01	0.30	6
4	StarLightCurve	MSP	0.20	50	0.01	0.14	6
5	StarLightCurve	MSP	0.12	100	0.01	0.11	6

Table 4: Benchmark results of the experiments for MAP and MSP for different datasets. The abbreviations in the column headings are the tested properties: (CR) Correctness, (CS) Consistency, (CN) Continuity, (CT) Contrastivity, (CC) Covariate Complexity, (CP) Compactness, (CF) Confidence, (IC) Input Completeness, (CLS) Cohesion of Latent Space.

Index	Dataset	Method	Val Loss	CR	CS	CN	CT	CC	CP	CF	IC	CLS	Total
1	Wafer	MAP	MSE: 1.26	0.85	0.28	1.00	0.30	0.68	0.79	0.64	0.00	0.70	0.58
2	Wafer	MAP	MSE: 0.58	0.94	0.46	1.00	0.27	0.73	0.79	0.71	0.50	0.75	0.68
3	Wafer	MAP	MSE: 0.27	0.93	0.73	0.99	0.29	0.76	0.79	0.64	0.75	0.70	0.73
4	Wafer	MAP	MSE: 0.28	0.93	0.79	1.00	0.28	0.76	0.79	0.71	0.50	0.73	0.72
1	Wafer	MSP	MSE: 0.86	1.00	0.48	1.00	0.43	0.54	0.79	0.90	0.00	0.54	0.63
2	Wafer	MSP	MSE: 0.35	0.71	0.47	0.65	0.27	0.66	0.79	0.75	0.40	0.76	0.61
3	Wafer	MSP	MSE: 0.35	0.87	0.48	0.93	0.36	0.72	0.79	0.64	0.20	0.83	0.65
4	Wafer	MSP	MSE: 0.23	0.41	0.32	0.23	0.27	0.82	0.79	0.88	0.75	0.78	0.58
1	SAWSINE	MAP	MSE: 0.37	0.74	0.76	0.98	0.70	0.57	0.79	0.69	1.00	0.54	0.78
2	SAWSINE	MAP	MSE: 0.18	0.99	0.32	0.99	0.41	0.64	0.79	0.52	0.75	0.59	0.67
3	SAWSINE	MAP	MSE: 0.17	0.99	0.61	0.99	0.32	0.71	0.79	0.55	1.00	0.64	0.73
4	SAWSINE	MAP	MSE: 0.14	0.99	0.71	0.98	0.36	0.74	0.79	0.53	0.75	0.69	0.73
5	SAWSINE	MAP	MSE: 0.13	0.99	0.76	1.00	0.30	0.75	0.79	0.57	1.00	0.68	0.76
1	SAWSINE	MSP	MSE: 0.18	0.51	0.16	0.94	0.38	0.61	0.79	0.36	0.00	0.74	0.50
2	SAWSINE	MSP	MSE: 0.14	0.50	0.44	0.95	0.27	0.78	0.79	0.81	0.25	0.77	0.62
3	SAWSINE	MSP	MSE: 0.14	1.00	0.43	0.94	0.28	0.67	0.79	0.84	0.33	0.73	0.67
4	SAWSINE	MSP	MSE: 0.14	0.50	0.30	0.87	0.24	0.82	0.79	0.92	0.50	0.77	0.63
1	FordB	MAP	MSE: 1.02	0.54	0.34	0.94	0.49	0.51	0.79	0.41	1.00	0.50	0.61
2	FordB	MAP	MSE: 0.83	0.54	0.40	0.96	0.53	0.52	0.79	0.40	0.75	0.51	0.60
3	FordB	MAP	MSE: 0.62	0.54	0.40	0.96	0.58	0.51	0.79	0.39	1.00	0.51	0.63
4	FordB	MAP	MSE: 0.49	0.46	0.38	0.96	0.58	0.51	0.79	0.39	0.75	0.51	0.59
1	FordB	MSP	MSE: 1.00	0.49	0.35	0.85	0.34	0.65	0.79	0.55	0.00	0.67	0.52
2	FordB	MSP	MSE: 1.00	0.49	0.42	0.61	0.31	0.74	0.79	0.77	0.50	0.68	0.59
3	FordB	MSP	MSE: 1.00	0.47	0.37	0.51	0.39	0.56	0.79	0.49	0.00	0.74	0.39
4	FordB	MSP	MSE: 1.00	0.54	0.29	0.52	0.24	0.73	0.79	0.73	0.40	0.66	0.54
1	FordA	MAP	MSE: 1.16	0.47	0.32	0.95	0.44	0.56	0.79	0.42	1.00	0.51	0.61
2	FordA	MAP	MSE: 1.05	0.47	0.34	0.96	0.44	0.54	0.79	0.43	1.00	0.51	0.61
3	FordA	MAP	MSE: 0.86	0.54	0.36	0.96	0.50	0.52	0.79	0.41	0.75	0.51	0.59
4	FordA	MAP	MSE: 0.70	0.47	0.37	0.96	0.53	0.51	0.79	0.40	0.75	0.51	0.59
5	FordA	MAP	MSE: 0.58	0.47	0.39	0.97	0.56	0.51	0.79	0.39	0.75	0.51	0.59
1	FordA	MSP	MSE: 1.00	0.40	0.54	0.56	0.41	0.74	0.79	0.75	0.40	0.64	0.58
2	FordA	MSP	MSE: 1.00	0.49	0.46	0.59	0.29	0.76	0.79	0.79	0.25	0.62	0.56
3	FordA	MSP	MSE: 1.00	0.58	0.31	0.61	0.31	0.72	0.79	0.75	0.50	0.67	0.58
4	FordA	MSP	MSE: 0.99	0.36	0.32	0.50	0.24	0.71	0.79	0.75	0.33	0.67	0.52
1	ECG200	MAP	MSE: 1.45	0.69	0.28	0.98	0.43	0.68	0.79	0.67	0.67	0.63	0.65
2	ECG200	MAP	MSE: 0.70	0.78	0.34	0.98	0.37	0.66	0.79	0.68	0.80	0.64	0.67
3	ECG200	MAP	MSE: 0.55	0.85	0.40	1.00	0.41	0.65	0.79	0.69	0.67	0.66	0.68
4	ECG200	MAP	MSE: 0.38	0.88	0.46	1.00	0.30	0.69	0.79	0.61	0.57	0.66	0.66
5	ECG200	MAP	MSE: 0.26	0.88	0.46	0.97	0.30	0.69	0.79	0.60	0.57	0.64	0.66
1	ECG200	MSP	MSE: 1.04	1.00	0.37	1.00	0.49	0.50	0.79	0.55	0.00	0.60	0.59
2	ECG200	MSP	MSE: 0.78	1.00	0.45	1.00	0.46	0.50	0.79	0.43	0.00	0.52	0.57
3	ECG200	MSP	MSE: 0.36	1.00	0.55	0.95	0.37	0.53	0.79	0.47	0.00	0.39	0.56
4	ECG200	MSP	MSE: 0.30	0.86	0.60	0.86	0.33	0.74	0.79	0.66	0.20	0.73	0.64
5	ECG200	MSP	MSE: 0.29	0.96	0.60	0.83	0.38	0.61	0.79	0.57	0.00	0.73	0.61
1	StarLightCurve	MAP	MSE: 1.07	0.82	0.26	0.98	0.32	0.68	0.67	0.76	0.67	0.58	0.64
2	StarLightCurve	MAP	MSE: 0.76	0.85	0.26	0.93	0.34	0.64	0.67	0.61	0.71	0.57	0.62
3	StarLightCurve	MAP	MSE: 0.55	0.85	0.32	0.94	0.32	0.54	0.79	0.63	0.63	0.55	0.62
4	StarLightCurve	MAP	MSE: 0.31	0.85	0.54	0.95	0.34	0.61	0.67	0.65	0.57	0.56	0.64
5	StarLightCurve	MAP	MSE: 0.13	0.85	0.65	0.97	0.31	0.65	0.67	0.63	1.00	0.57	0.71
1	StarLightCurve	MSP	MSE: 0.42	0.21	0.38	0.97	0.33	0.54	0.67	0.36	0.00	0.58	0.45
2	StarLightCurve	MSP	MSE: 0.28	0.29	0.34	0.99	0.32	0.55	0.67	0.59	0.00	0.65	0.49
3	StarLightCurve	MSP	MSE: 0.22	0.38	0.56	0.60	0.35	0.59	0.67	0.66	0.00	0.66	0.50
4	StarLightCurve	MSP	MSE: 0.20	0.39	0.42	0.25	0.24	0.58	0.67	0.62	0.11	0.66	0.44
5	StarLightCurve	MSP	MSE: 0.12	0.40	0.62	0.72	0.27	0.80	0.67	0.76	0.38	0.74	0.59

Table 5: Benchmark results of the outlier analysis for MAP and MSP for SAWSINE datasets. Here, "mixed" refers to the original models evaluated on the outlier dataset, while "Outlier" denotes the models that was both trained and evaluated using the outlier dataset. The abbreviations in the column headings are the tested properties: (CR) Correctness, (CS) Consistency, (CN) Continuity, (CT) Contrastivity, (CC) Covariate Complexity, (CP) Compactness, (CF) Confidence, (IC) Input Completeness, (CLS) Cohesion of Latent Space.

Index	Dataset	Method	Val Loss	CR	CS	CN	CT	CC	CP	CF	IC	CLS	Total
1	SAWSINE	MAP	MSE: 0.37	0.74	0.76	0.98	0.70	0.57	0.79	0.69	1.00	0.54	0.78
2	SAWSINE	MAP	MSE: 0.18	0.99	0.32	0.99	0.41	0.64	0.79	0.52	0.75	0.59	0.67
3	SAWSINE	MAP	MSE: 0.17	0.99	0.61	0.99	0.32	0.71	0.79	0.55	1.00	0.64	0.73
4	SAWSINE	MAP	MSE: 0.14	0.99	0.71	0.98	0.36	0.74	0.79	0.53	0.75	0.69	0.73
5	SAWSINE	MAP	MSE: 0.13	0.99	0.76	1.00	0.30	0.75	0.79	0.57	1.00	0.68	0.76
1	SAWSINE	MSP	MSE: 0.18	0.51	0.16	0.94	0.38	0.61	0.79	0.36	0.00	0.74	0.50
2	SAWSINE	MSP	MSE: 0.14	0.50	0.44	0.95	0.27	0.78	0.79	0.81	0.25	0.77	0.62
3	SAWSINE	MSP	MSE: 0.14	1.00	0.43	0.94	0.28	0.67	0.79	0.84	0.33	0.73	0.67
4	SAWSINE	MSP	MSE: 0.14	0.50	0.30	0.87	0.24	0.82	0.79	0.92	0.50	0.77	0.63
1	SAWSINE mixed	MAP	MSE: 0.42	0.71	0.76	0.98	0.70	0.57	0.79	0.69	1.00	0.54	0.75
2	SAWSINE mixed	MAP	MSE: 0.22	0.97	0.32	0.98	0.41	0.64	0.79	0.51	0.75	0.58	0.70
3	SAWSINE mixed	MAP	MSE: 0.21	0.98	0.61	0.99	0.32	0.71	0.79	0.55	1.00	0.64	0.74
4	SAWSINE mixed	MAP	MSE: 0.18	0.97	0.71	0.98	0.36	0.74	0.79	0.53	0.75	0.68	0.72
5	SAWSINE mixed	MAP	MSE: 0.17	0.98	0.76	0.99	0.30	0.75	0.79	0.56	1.00	0.67	0.75
1	SAWSINE mixed	MSP	MSE: 0.22	0.50	0.16	0.93	0.38	0.68	0.79	0.33	0.00	0.75	0.54
2	SAWSINE mixed	MSP	MSE: 0.21	0.50	0.44	0.95	0.27	0.74	0.79	0.79	0.25	0.77	0.63
3	SAWSINE mixed	MSP	MSE: 0.19	0.99	0.43	0.93	0.28	0.74	0.79	0.81	0.11	0.76	0.68
4	SAWSINE mixed	MSP	MSE: 0.19	0.49	0.30	0.85	0.24	0.83	0.79	0.90	0.50	0.81	0.68
1	SAWSINE Outlier	MAP	MSE: 0.62	0.92	0.36	0.97	0.58	0.59	0.79	0.57	1.00	0.54	0.74
2	SAWSINE Outlier	MAP	MSE: 0.35	0.96	0.45	0.98	0.54	0.61	0.79	0.52	0.75	0.55	0.71
3	SAWSINE Outlier	MAP	MSE: 0.30	0.98	0.42	0.98	0.45	0.61	0.79	0.50	1.00	0.56	0.74
4	SAWSINE Outlier	MAP	MSE: 0.27	0.98	0.44	0.98	0.40	0.63	0.79	0.47	1.00	0.57	0.73
5	SAWSINE Outlier	MAP	MSE: 0.24	0.98	0.50	0.99	0.42	0.65	0.79	0.47	1.00	0.60	0.70
1	SAWSINE Outlier	MSP	MSE: 0.22	0.98	0.54	0.98	0.37	0.54	0.79	0.52	0.00	0.75	0.62
2	SAWSINE Outlier	MSP	MSE: 0.21	0.84	0.65	0.94	0.38	0.64	0.79	0.67	0.20	0.76	0.65
3	SAWSINE Outlier	MSP	MSE: 0.19	0.57	0.21	0.98	0.33	0.68	0.79	0.72	0.25	0.77	0.63
4	SAWSINE Outlier	MSP	MSE: 0.19	0.53	0.64	0.96	0.29	0.77	0.79	0.82	0.50	0.80	0.68



(a) Without outliers **(b) With Outliers**

Figure 4: Dimension-reduced representation of the latent space for the SAWSINE MAP model with index 5 evaluated with and without outliers. Due to the dimension reduction distances are distorted. Small dots represent input data, with lines connecting each prototype (circle) to its corresponding cluster centroid (square). Points are color-coded by class.