UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Physics and Technology

Towards Interpretable, Trustworthy and Reliable AI

Srishti Gautam

A dissertation for the degree of Philosophiae Doctor

December 2023

This thesis document was typeset using the *UiT Thesis LaTeX Template*.
© 2023 – http://github.com/egraff/uit-thesis

Abstract

The field of artificial intelligence recently witnessed remarkable growth, leading to the development of complex deep learning models that perform exceptionally across various domains. However, these developments bring forth critical issues. Deep learning models are vulnerable to inheriting and potentially exacerbating biases present in their training data. Moreover, the complexity of these models leads to a lack of transparency, which can allow biases to go undetected. This can lead to ultimately hindering the adoption of these models due to a lack of trust. It is therefore crucial to foster the creation of artificial intelligence systems that are inherently transparent, trustworthy, and fair.

This thesis contributes to this line of research by exploring the interpretability of deep learning through self-explainable models. These models represent a shift towards more transparent systems, offering explanations that are integral to the model's architecture, yielding insights into their decision-making processes. Consequently, this inherent transparency enhances our understanding, thereby providing a mechanism to address the inadvertent learning of biases.

To advance the development of self-explainable models, this thesis undertakes a comprehensive analysis of current methodologies. It introduces a novel algorithm designed to enhance the explanation quality of one of the state-of-the art models. In addition, this work proposes a novel self-explainable model that surpasses existing methods by generating explanations through a learned decoder, facilitating end-to-end training, and addressing the prevalent trade-off between explainability and performance. Furthermore, to enhance the accessibility and sustainability of these models, this thesis also introduces a universal methodology to transform any pre-trained black-box model into a self-explainable one without the need for re-training.

Through the proposed methodology, this research identifies and counteracts the learning of artifacts – spurious correlations – from the data, further emphasizing the need for transparent models. Additionally, this thesis expands its scope to encompass the dimension of fairness for large language models, demonstrating the tendency of these models to reinforce social biases.

The results of this research highlight the efficacy of the proposed methodologies, thereby paving the way for artificial intelligence systems that are not only accurate but also transparent, fair, and reliable, to facilitate widespread adoption and trust in artificial intelligence technologies.

Acknowledgements

Paraphrasing a quote from the Dead Poet's Society, *"Science, Engineering, Medicine, and Business, these are noble pursuits, and necessary to sustain life. But guide, camaraderie, friends and family, these are what we stay alive with"*. This thesis, but more importantly, these four years would not have been possible without the support of several beautiful souls. Because sometimes, angels are just ordinary people who help us believe in magic again, and the people mentioned here have all been my angels.

First and foremost, I want to thank *Michael*. I have always said this to everyone around me, except for you, yet, that even if I would have gone around the world searching every little corner with a flashlight, I would not have been able to find a better guru to guide me through this journey. From — always being a great supporter of my stupid ideas, questioning my thinking, and constantly, without fail, motivating me every step of the way, — to helping me with everything ordinary, making me feel comfortable in an extra-terrestrial place such as Tromsø, I could never thank you enough.

Robert, thank you for your unwavering support, for your constant guidance throughout, for multiple revisions of my silly writing, and for making me feel visible in this group, in this country.

To my dearest *Ahcene*, nothing would have been possible without you. Thank you for adopting me as your student. You have changed my way of thinking about the "embeddings" for forever, for which I am eternally grateful. But more importantly, thank you for being a constant warm support in a literal cold place. I never imagined that I could ever be so lucky to find one of the closest friends in you. Life is a little sad without you around and it'll always be.

To my dearest camaraderie, the entire machine learning group at UiT. Thank you for being welcoming since the very first day. I am sure that this is the best group of people existing in this world, atleast to me. *Gigi*, thank you for being there whenever I needed you, and for my 5 out of 8 hugs a day. Everytime I see you I feel protected and a little lighter. Also, I am forever grateful to you for bringing a beautiful soul like *Tena* in my life. *Stine*, *Sara* and *Suaiba*, thank you for all the women power I needed to get through this and of course our cry sessions, which healed me several times. *Stine*, thank you for being the best office-mate I could ever ask for. *Changkyu*, you took care of me since the first day until now, I cannot imagine how you're able to be on toes to help anyone who needs it. I am forever grateful for this. *Magnus* and *Teddy*, thank you for always cheering me up in my coldest darkest times and for all the big hugs. My dearest *Suaiba*, words would never be enough for you. You have been the light of my life in Tromsø. From the first time that you took me out for coffee when I needed someone, to all of our dates in Egon, and of course our beautiful beautiful 'work' trips. I love you from the bottom of my heart.

To *Iqra* and *Riya*, my family in Tromsø, and my personal private detectives. I get tears in my eyes whenever I think of both of you. Thank you for opening my eyes to the true meaning of female friendship. Two of the strongest female role models of my life. Thank you for celebrating all the little wins, and for holding my hand through all the big losses. I don't think anyone in this world deserves you *Iqra*. Your compassion, your patience and our mutual love for chai, is what I admire the most. Thank you for opening up your home and your heart to me. *Riya*, considering how much family means to you, I feel so lucky to call you my little sister. Thank you for never ever saying no to me. Chai? yes, Wine? yes, Netflix? yes, Suryanamaskar? yes.

To dearest *Prabhjot* and my kiddo *Nihal*, I cannot express how grateful I am to have you both in my life. My strongest support in my weakest times.

To my boi, my dearest *Bans*, you are my best friend, my personal cheerleader, the warmth to my soul and my eyes to this beautiful world. You have made every second of this journey, this life, worth living. Thank you for picking me up whenever I broke down during these four years, but especially in the US. Thank you for existing.

To *Avnish*, you have been my backbone for a long time now and I cannot be more thankful for it. Thank you for taking care of me in the foreign land of US, and making Worcester and Palo Alto home for me. You are the best non-sibling sibling I could ever ask for.

Family is what I life for, every single day. *Papa*, my superhero, thank you for teaching me how to be a strong, independent, yet, compassionate person. *Mumma*, from you I learned how to love unconditionally. I am the person I am because of both of you. My brother *Sid* and bhabhi *Aman*, you still spoil me like a small child, while respecting me and being proud of me at the same time. Thank you for taking care of me since forever. My sisters, *Munnu* and *Chutka*, and my brother, *Vishu*, it makes my heart warm to think about how you still fight over me. Nothing in life would have been this easy and colorful without you.

To *Sumit*, yes you are my family because you have always taken care of me like I am yours. Thank you for your unconditional love.

Finally, to my love, *Prashant*. You have stayed strong and in turn made stronger, holding my hand from so far away through everything. From my master's thesis' acknowledgments to here, you deserve a whole page. But because I am very much unsure about the thin line between an acknowledgement and a love letter, I'll keep it short. Without you, my degree, my life, would not make sense. I love you more than yesterday and less than tomorrow.

Srishti
Tromsø, December 2023

Contents

List of Figures

List of Abbreviations

***k*-NN** *k*-Nearest Neighbor

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

CE Cross-Entropy

CNN Convolutional Neural Network

DL Deep Learning

FLINT Framework to Learn with Interpretation

GPT Generative Pretrained Transformer

KMEx K-Means Explainer

LIME Local Interpretable Model-agnostic Explanation

LLM Large Language Model

LRP Layer-wise Relevance Propagation

LSTM Long Short-Term Memory network

ML Machine Learning

MLP Multilayer Perceptron

MVC Multi-View Clustering

NLP Natural Language Processing

ProtoPNet Prototypical Part Network

ProtoVAE Prototypical Variational Autoencoder

PRP Prototypical Relevance Propagation

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SEM Self-Explainable Model

SGD Stochastic Gradient Descent

SHAP SHapley Additive exPlanations

SITE Self Interpretable Tranformation Equivariant network

VAE Variational Autoencoder

ViT Vision Transformer

XAI Explainable Artificial Intelligence

1

Introduction

Machine learning (ML), a fundamental branch of Artificial Intelligence (AI), leverages statistical techniques to enable computers to perform complex tasks through the recognition of patterns within curated datasets [17]. Deep Learning (DL), a subset of ML characterized by neural networks with multiple layers, further refines this capability, allowing for the analysis and interpretation of high-dimensional data across numerous applications. These networks excel at learning hierarchical representations, which is particularly beneficial for processing unstructured data such as images, audio, and text [18]. Recently, due to the availability of large datasets, as well as the ever expanding computing capabilities, DL has gained extensive adoption [18, 19] revolutionizing a multitude of fields such as computer vision [20, 21] and Natural Language Processing (NLP) [22, 23]. In computer vision, ML algorithms have significantly advanced the computer's ability to process visual information, enabling progress in image recognition and object detection [20, 24, 25]. In NLP, these algorithms, such as Large Language Models (LLMs), have achieved a nuanced understanding of language, improving tasks such as machine translation, sentiment analysis, and the development of conversational agents [22, 23, 26].

As these models have evolved and their capabilities have become more sophisticated, they are beginning to be deployed within domains that are more safety-critical [27, 28]. Such domains include healthcare, where ML models are being developed to assist in patient diagnoses and treatment plans [27, 29, 30], finance, where ML is used for complex tasks such as credit scoring [31, 32] and algorithmic trading [33], affecting investment strategies and fraud detection systems [28] and autonomous driving, where these models process and interpret sensor data to make real-time driving decisions [34, 35].

Nevertheless, this transition of DL into applications with significant safety and ethical implications necessitates a deeper understanding of how these models function and make decisions [36]. However, the inherent complexity of DL models, with their deep, non-linear architectures and extensive parameters, often results in opaque decision-making, posing challenges to achieving transparency and interpretability [37]. Consequently, the field of Explainable Artificial Intelligence (XAI) has emerged, focusing on developing techniques that render the inner workings of DL models more accessible and understandable to humans, thereby facilitating their responsible use in high-stakes scenarios.

Additionally, the efficacy of DL is subject to further challenges, apart from transparency, that can impact model reliability and fairness. One of these concerns is artifact learning, where DL algorithms may inadvertently learn intended or unintended correlations or artifacts present in the training data, leading to skewed results [38]. Furthermore, fairness of the decisions of these models is also an important consideration in the development and deployment of DL systems [39]. Biases in the training data can result in discriminatory outcomes, unknowingly perpetuating existing inequalities. In the light of these challenges, there is an increasing imperative to develop AI that is responsible, trustworthy, as well as transparent [36, 40], ensuring that the deployment of DL maximizes benefits while minimizing potential adverse impacts.

The focus of this thesis is to tackle these challenges by developing new methodologies in the field of trustworthy DL These challenges are presented in the following section, and addressed in the included papers in this thesis.

1.1 Key Challenges and Opportunities

This thesis will specifically address three key challenges in DL: (1) Lack of accurate and precise self-explainability methods, (2) Artifact learning in DL models, (3) Fairness in LLMs.

Lack of accurate and precise self-explainability methods DL models often lack transparency and interpretability, making it challenging to understand the reasoning behind their decisions. To address this challenge, recently, the field of XAI has emerged [36, 37]. XAI focuses on developing methods that can provide explanations for the decisions made by traditionally black-box DL models. XAI has been developed recently in two parallel branches: post-hoc methods and Self-Explainable Models (SEMs) [1].

Post-hoc methods aim to explain the decisions of black-box models retroactively. These methods analyze the internal workings of an already trained model and provide insights into which features or factors contributed to a particular decision. While these have been useful in explaining the decisions of DL models, they have demonstrated limitations in the fidelity and accuracy of the generated explanations [36], occasionally producing explanations that are imprecise or potentially misleading [41]. On the other hand, SEMs are designed to incorporate the capability to provide explanations and decisions simultaneously. These models are built with inherent interpretability, allowing them to generate decisions as well as their corresponding explanations simultaneously. SEMs are considered more desirable as they eliminate the need for separate post-hoc analysis while providing more faithful explanations [36]. However, current SEMs still face challenges in generating precise explanations [1] and often lag behind black-box models in terms of predictive performance [42].

Artifact learning in DL models DL algorithms, while effective at learning patterns and relationships from large datasets, can inadvertently pick up artifacts present in the training data. These artifacts can lead to unintended correlations and biases affecting the model's decision-making process, hindering their generalizability and potentially compromising their fairness and reliability. One example of such artifacts is the Clever Hans effect [43–45], where models learn to rely on unintended cues or correlations in the training data to achieve the desired outcome, rather than truly understanding the underlying concepts. This can result in models that appear to perform well during training but fail to generalize to new, unseen data. Moreover, the presence of artifacts in the training data can be exploited by adversaries to generate malicious attacks. For instance, backdoor attacks can be designed to manipulate the model's behavior by injecting specific patterns or triggers into the training data [46, 47]. These attacks can compromise the integrity and security of the model, leading to biased or manipulated outcomes.

Transparency and interpretability in DL models are crucial for addressing artifact learning. XAI methods can help identify biases, artifacts, and unintended correlations [44]. By understanding how the model arrives at its decisions, we can detect and mitigate the impact of artifacts, improving the fairness and reliability of the model's outcomes.

Fairness in LLMs LLMs have indeed gained significant popularity in recent years, with models like Generative Pretrained Transformer (GPT) becoming widely used in various applications [22]. The accessibility, usability, and success of LLMs have led to their widespread adoption in real-world scenarios. However, it is important to recognize that these models are still relatively young, and their reliability and trustworthiness have not been extensively studied. One critical aspect to consider is the potential bias present in the training data used to train LLMs. The data available online is known to be historically biased, reflecting societal biases and prejudices [48]. Since LLMs are trained on vast amounts of data, it is crucial to understand the fairness implications of these models. Biases and stereotypes present in the training data can be learned and perpetuated by LLMs, leading to biased or unfair outcomes in their decisions [49]. Understanding and addressing biases and stereotypes in LLMs is thus essential to enhance their fairness and accuracy, making them more reliable and trustworthy.

By focusing on these key challenges, this thesis aims to contribute to the development of responsible and trustworthy DL models. The research conducted will explore innovative approaches to enhance self-explainability, mitigate unintended artifact learning, and investigate biases in LLMs. Ultimately, the goal is to advance the field of DL and promote the adoption of transparent, safe and fair AI systems.

1.2 Research Objectives

To address the key challenges above, this thesis proposes novel methodology for DL, focusing mainly on XAI and trustworthiness. The main objectives of the thesis are summarized as follows:

1. **Enhancing *transparency* of AI systems**: Addressing the gaps in development of XAI by improving explainability of existing SEMs and development of novel SEMs.

2. **Enhancing *reliability* of AI systems**: Identifying and mitigating the unintended artifact learning in DL models, thereby preventing their reliance on erroneous features. Further, investigating the risks of artifact learning in an application of healthcare, where such spurious correlations could lead to severe repercussions.

3. **Analysing un-/*fairness* of AI systems**: Investigating fairness of large language models, involving assessment of their outputs with respect to biases across different demographics.

1.3 Proposed Approaches

The methodology developed in this thesis addresses the first research objective in three different ways: In Paper I, we investigate the quality of explanations of one of the state-of-the-art SEM, Prototypical Part Network (ProtoPNet) [42] and propose a methodology, called Prototypical Relevance Propagation (PRP), for generating more accurate and precise explanations for prototypical SEMs. In Paper III, we propose a novel SEM based on a Variational Autoencoder (VAE) [50] backbone, where a mixture of VAEs are trained, each representing a different class-prototype. This allows the model to incorporate the inherent capability to generate class-based explanations in the input-space with the help of the learned decoder without losing accuracy. Further, in Paper IV, we propose a universal method for converting any black-box method into a self-explainable one without requiring re-training, thus promoting the accessibility of SEMs.

Research objective 2 is addressed in Paper I and Paper II. In Paper I, we tackle the problem of artifact detection, specifically focusing on Clever Hans and backdoor artifacts. Utilizing the proposed PRP, we generate precise explanations for training data for all prototypes. We then apply Multi-View Clustering (MVC) on these multiple prototypical explanations to clean the data, thereby mitigating the possibility of artifact learning. In Paper II, we utilize PRP to reveal potential biases inadvertently learned by DL models within the critical domain of healthcare. Our findings indicate that models trained on data amalgamated from various hospitals or sources may inherit and propagate biases, leading to unreliable outcomes.

The research objective 3 is addressed in Paper V, where we extend our research focus into the domain of LLMs. We scrutinize the propagation of existing demographic biases in LLMs when applied to tabular tasks. This investigation is crucial for assessing the fairness and impartiality of LLMs in practical applications.

1.4 Brief Summary of Included Papers

The thesis' main contribution are the five included papers which are briefly summarized in the following. Figure 1.1 provides an overview of the topics considered in various papers.

[I] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “This looks more like that: Enhancing self-explaining models by prototypical relevance propagation.” *Pattern Recognition* 136 (2023), p. 109172.

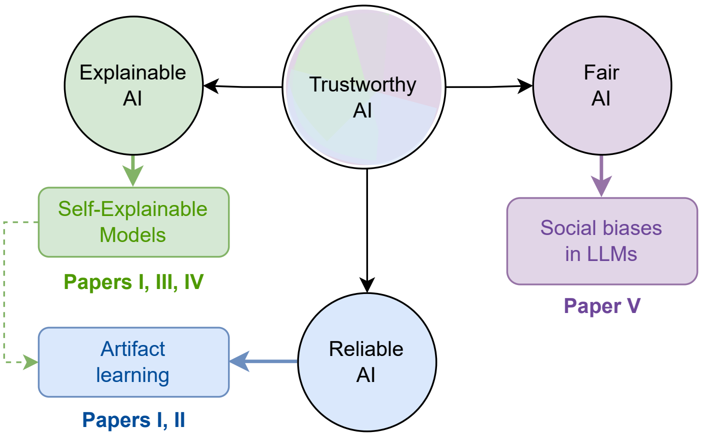


Figure 1.1: Overview of the topics that the various papers address.

[II] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. "Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation." In: *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2022, pp. 1–5.

[III] Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. "Protovae: A trustworthy self-explainable prototypical variational model." *Advances in Neural Information Processing Systems* 35 (2022), pp. 17940–17952.

[IV] Srishti Gautam, Ahcene Boubekki, Marina Höhne, and Michael C Kampffmeyer. "Prototypical Self-Explainable Models Without Re-training." *Under Review* (2023).

[V] Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. "Investigating the Fairness of Large Language Models for Predictions on Tabular Data." *Under Review* (2023).

Paper I This paper tackles the challenges associated with inadvertent Clever Hans artifacts and backdoor learning, proposing PRP as our solution. PRP is able to generate more accurate and precise explanations for existing prototypical self-explainable methods. We focus on one of the state-of-the-art self-explainable model, ProtoPNet, and demonstrate how PRP is able to capture the learning of artifacts more precisely as compared to the original explanations generated by ProtoPNet. Following this, we further propose to clean the dataset using PRP explanations and MVC, thereby suppressing the possibility of artifact-learning by the models.

Paper II This paper focuses on the problem of Pneumonia detection in Chest X-Ray images. Our investigation is particularly focused on the intricacies that arise when combining data from multiple sources or hospitals, a scenario often encountered in data-intensive DL models. We illustrate, with the aid of our proposed PRP method, how a small source-related label imbalance is sufficient for DL models to function more as "hospital detectors". For example, when a significant majority of Pneumonia cases originate from Hospital 1, these models inadvertently shift their role from being "disease-detectors" to identifying the source hospital instead. As a result, this research underscores the importance of employing SEMs within safety-critical domains.

Paper III ProtoPNet, one of the early and state-of-the-art SEMs, relies on projecting the learned prototypes (vectors in the latent space) to the training data for visualization. However, this creates a bottleneck in the end-to-end learning of the whole model, thereby impacting accuracy. In this paper, we tackle this issue by proposing a probabilistic and generative SEM based on a VAE backbone, called Prototypical Variational Autoencoder (ProtoVAE). ProtoVAE learns a transparent prototypical space by training a mixture of VAEs, each sharing the same encoder and decoder but, with a separate Gaussian prior centered on different class prototypes. The decoder enables these class-prototypes to be directly visualized in the input space. ProtoVAE achieves this transparent decision-making through end-to-end training, eliminating the need for trade-offs in accuracy often associated with other SEMs.

Paper IV In this paper we introduce K-Means Explainer (KMEx), a more generalized SEM, having the unique capability to transform any existing black-box model into a self-explainable one without requiring re-training. It achieves this by preserving the black-box model's backbone, learning prototypes within the latent space using K-Means clustering, and replacing the final classification layer with a 1-nearest-neighbor classifier based on the acquired prototypes. We further propose a comprehensive quantitative evaluation framework for prototypical SEMs, thereby highlighting key strengths and weaknesses of several state-of-the-art SEMs.

Paper V In this paper, we extend our research focus into the domain of LLMs, which have gained recent widespread usage, and delve into the fairness achieved by these when applied to tabular tasks. These models differ from traditional ML approaches in that they can interpret contextual information, such as column names, within tabular data. This capability, however, raises concerns about the potential for LLMs to amplify demographic biases. We undertake a thorough examination to assess the fairness of LLMs by comparing different classification strategies for tabular data, including zero-shot learning, few-shot in-context learning, and fine-tuning of pre-trained LLMs. This multifaceted approach underscores the persistence of bias-related challenges in LLMs.

1.5 Other Contributions

During the course of this thesis, several works were contributed (as listed below), majorly focusing on the prototypical learning. These contributions have propelled advancements in AI, both through enhancements in theoretical understanding and through the demonstration of practical applications, particularly in the analysis of medical images.

[6] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. "This looks more like that: Enhancing self-explaining models by prototypical relevance propagation." *National Conference on Image Processing and Machine Learning (NOBIM)* (2021). Extended abstract and oral presentation.

[7] Stine Hansen, Srishti Gautam, Robert Jenssen, and Michael Kampffmeyer. "Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels." *National Conference on Image Processing and Machine Learning (NOBIM)* (2021). Extended abstract and oral presentation.

[8] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. "Artifact Detection with Prototypical Relevance Propagation." *Visual Intelligence Days* (2021). Oral presentation.

[9] Srishti Gautam. "Self-Explainability and Artifact detection: Along with applications to medical data." *COMP-7950-T04 – Advanced Machine Learning Event, University of Manitoba, Canada* (2021). Invited Talk.

[10] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. "Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation." *NORA Annual Conference* (2022). Extended abstract and oral presentation.

[11] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. "Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation." *Visual Intelligence Days* (2022). Poster presentation.

[12] Stine Hansen, Srishti Gautam, Robert Jenssen, and Michael Kampffmeyer. "Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels." *Medical Image Analysis* 78 (2022), p. 102385.

[13] Suaiba Amina Salahuddin, Stine Hansen, Srishti Gautam, Michael Kampffmeyer, and Robert Jenssen. “A self-guided anomaly detection-inspired few-shot segmentation network.” CEUR Workshop Proceedings. 2022.

[14] Srishti Gautam. “Bias in Machine Learning.” *Bias in Artificial Intelligence Workshop at UiT – The Arctic University of Norway* (2023). Invited Talk.

[15] Stine Hansen, Srishti Gautam, Suaiba Amina Salahuddin, Michael Kampffmeyer, and Robert Jenssen. “ADNet++: A few-shot learning framework for multi-class medical image volume segmentation with uncertainty-guided feature refinement.” *Medical Image Analysis* (2023), p. 102870.

[16] Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. “Investigating the Fairness of Large Language Models for Predictions on Tabular Data.” *NeurIPS Workshop on Socially Responsible Language Modelling Research* (2023).

1.6 Reading Guide

This thesis is structured into five parts: *I) Deep Learning Basics*, *II) Transparency and Explainability*, *III) Responsible and Fair AI*, *IV) Summary of Research*, and *V) Included Papers*.

Deep Learning Basics provides the basic machine learning concepts (Chapter 2) and deep learning theory (Chapter 3) relevant for this thesis. *Transparency and Explainability* introduces Explainable AI (Chapter 4) with a brief overview of existing approaches that are relevant for Papers I to 4 of this thesis. *Responsible and Fair AI* discusses intentional and unintentional artifact learning in DL (Chapter 5), followed by bias and fairness in LLMs (Chapter 6). *Summary of Research* provides a summary of the four included papers, their scientific contributions, and the specific contributions of the author (Chapter 7 – 11), followed by concluding remarks of the work (Chapter 12). *Included Papers* lists the included papers in the thesis.

Part I

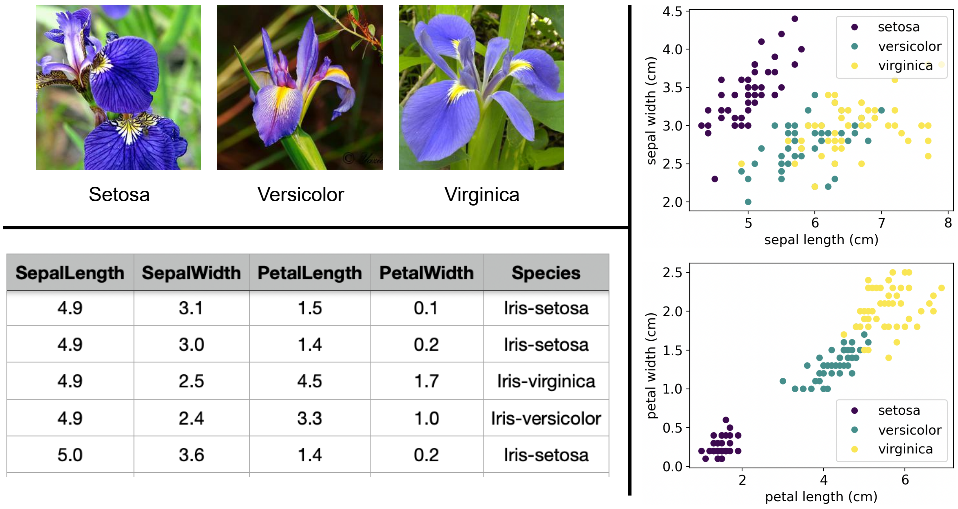
Deep Learning Basics

2

Machine Learning

ML, a significant subfield of AI, provides systems with the ability to learn from data and make decisions. The primary aim of ML algorithms is to uncover patterns in datasets, employing statistical methods to analyze and understand complex information [17]. This capability facilitates the generation of predictive insights, allowing these algorithms to make informed decisions based on the data they process [51]. The broad applicability of ML is demonstrated by its integration into various sectors, ranging from diagnostic procedures in healthcare [52, 53] to predictive analytics in finance [28, 32, 33], thereby continually expanding the horizons of machine-driven problem-solving. An ML model is built upon three essential elements: 1) the dataset that provides the basis for learning, 2) the specific task the model aims to solve along with the chosen ML algorithm for solving it, and 3) the performance metrics that evaluate the effectiveness of the model.

- *Dataset:* This is the raw material from which knowledge is extracted. It can come in various forms, such as images, text, or numerical values, and is often divided into training and testing sets. The quality, quantity, and relevance of the data directly influence the model's ability to learn effectively. Consider the case of a numerical data represented by the tuple $(\boldsymbol{X}, \boldsymbol{Y})$. In this instance, $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ denotes the input data, typically organized into a feature matrix with n instances and d features per instance, where each row is an individual observation and each column corresponds to a specific attribute of the data and $\boldsymbol{Y}$ represents the expected outcome or target. While some datasets provide both the input



SepalLength	SepalWidth	PetalLength	PetalWidth	Species
4.9	3.1	1.5	0.1	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
4.9	2.5	4.5	1.7	Iris-virginica
4.9	2.4	3.3	1.0	Iris-versicolor
5.0	3.6	1.4	0.2	Iris-setosa

Figure 2.1: Example of images from three classes of Iris dataset [54] (top left), the numerical features available for this dataset (bottom left), and 2-dimensional plots of selected features (right).

X and the corresponding targets Y, others may only include the input data.

To exemplify, the Iris dataset [54] has been used as a benchmark in ML [55, 56], encompassing a collection of morphological measurements from three varieties of the Iris plant. It consists of 150 instances, each characterized by four features: sepal length, sepal width, petal length, and petal width. The provided target variable Y categorizes each instance into one of three species: setosa, versicolor, or virginica. Figure 2.1 displays samples from this dataset, showcasing the original flower images, the extracted features, and feature-based plots.

- *ML task and algorithm*: The task defines the problem that the ML algorithm is intended to solve. This could range from simple classification or regression tasks to more complex challenges like NLP or image recognition. The task not only informs the design of the algorithm but also determines the type of output it is expected to produce. For example, with the Iris dataset, the ML task is to classify each flower into one of three species categories, utilizing the features provided. The algorithm's design is fundamentally influenced by the nature of the data it uses. If the algorithm employs the target variable, Y within the dataset, then the learning process is categorized as supervised. However, if the target variable is not utilized, the learning falls under the category of unsupervised.

We define an ML algorithm as a function f_θ parameterized by θ. For a given input instance $\boldsymbol{x}_i$, the algorithm's prediction is $\hat{\boldsymbol{y}}_i$, such that, $f_\theta(\boldsymbol{x}_i) = \hat{\boldsymbol{y}}_i$ holds. In an ideal scenario, the predicted outcome $\hat{\boldsymbol{y}}_i$ would be equal to the expected outcome ($\boldsymbol{y}_i$ for supervised learning), indicating perfect performance by the ML algorithm. However, in practice, discrepancies may occur. The goal is thus to minimize these differences by refining the parameters $\boldsymbol{\theta}$. This is accomplished by employing optimization strategies that aim to minimize a predefined loss function $\mathcal{L}_\theta$, thereby iteratively adjusting the parameters $\boldsymbol{\theta}$ of the function $\boldsymbol{f}$ to improve its predictions.

- *Evaluation*: The performance of an ML model is quantified using various evaluation metrics that are selected based on the nature of the task at hand. Common metrics include accuracy, precision, and recall, which provide insights into the model's predictive capabilities [57]. The evaluation process is integral to the model development cycle, as it informs the selection of the most appropriate algorithm, guides the tuning of algorithm's parameters, and ultimately determines the efficacy of the ML application in practical settings. For example, when working with the Iris dataset, a key metric for evaluation would be the model's accuracy, which measures the proportion of instances that are correctly classified into the three distinct species categories.

The following section will delve into the supervised and unsupervised ML paradigms, examining their characteristics, the types of tasks they are suited for, and providing examples of ML algorithms that can be employed to accomplish these tasks.

2.1 Supervised learning

Supervised learning is a subcategory of ML, consisting of algorithms trained to output the labels [58]. This approach involves using a predefined dataset with known outcomes, denoted as $\boldsymbol{y}_i$ for each input $\boldsymbol{x}_i$, to teach the model a function that maps inputs to desired outputs. Common examples of supervised learning tasks include classification (classifying data into pre-defined categories) [59], regression (predicting a continuous output value based on input) [60], object detection (locating objects within images) [61], speech recognition (translating spoken words into text) [62], among others. The subsequent sub-section will concentrate on the topic of classification, a central theme across all papers included in this thesis.

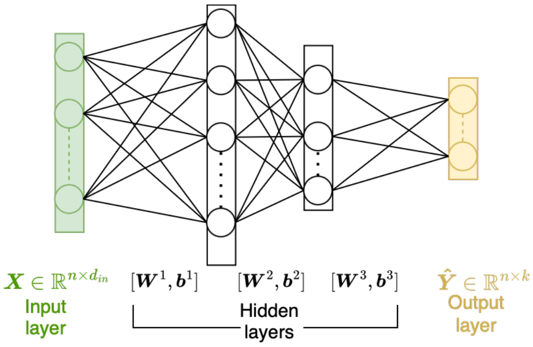

$\boldsymbol{X} \in \mathbb{R}^{n \times d_{in}}$
Input layer
$[\boldsymbol{W}^1, \boldsymbol{b}^1]$
$[\boldsymbol{W}^2, \boldsymbol{b}^2]$
$[\boldsymbol{W}^3, \boldsymbol{b}^3]$
Hidden layers
$\hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times k}$
Output layer

Figure 2.2: An MLP with 2 hidden layers.

2.1.1 Classification

In a classification task, the ML model's objective is to predict the class or category of new, unseen data $\boldsymbol{X}_{\text{test}}$, based on the knowledge gained from the training data $(\boldsymbol{X}_{\text{train}}, \boldsymbol{Y}_{\text{train}})$, where the classes are known apriori. Such tasks are prevalent in various applications, including the categorization of emails in spam detection systems [63], the identification of objects in images [64], and the diagnosis of diseases from medical imaging [52]. A multitude of ML algorithms are suitable for classification, among which are multilayer perceptron (MLP) [65], Support Vector Machines [66], and Decision Trees [67], each offering distinct advantages and potential drawbacks. This sub-section will delve into MLPs, given their relevance to the research discussed in this thesis.

Multilayer Perceptrons

MLPs are a category of artificial neural networks characterized by their layered structure of nodes or neurons [65], as shown in Figure 2.2.

Architecture The architecture of an MLP, f_{θ}, typically includes an input layer, one or more hidden layers, and an output layer (refer Figure 2.2).

1. Input Layer: This layer receives the input signal to be processed. Each node in this layer represents an attribute or feature of the input data $\boldsymbol{X} \in \mathbb{R}^{n \times d_{in}}$, where n is the number of samples and d_{in} is the number of input features.

2. Hidden Layers: These intermediate layers perform the majority of the computation through a series of weighted connections. Each neuron in a hidden layer transforms the values from the previous layer with a weighted linear summation, weights represented by $\boldsymbol{W} \in \mathbb{R}^{d_{in} \times d_{out}}$, and

biases $\boldsymbol{b} \in \mathbb{R}^{d_{out}}$, followed by a non-linear activation function, $\boldsymbol{g}$ such as the sigmoid, hyperbolic tangent, or Rectified Linear Unit (ReLU) function [68]. The forward pass through a single layer l of an MLP, for a sample i, is computed as:

$$\boldsymbol{y}_i^l = \boldsymbol{g}(\boldsymbol{x}_i^T \boldsymbol{W}^l + \boldsymbol{b}^l) \tag{2.1}$$

3. Output Layer: The final layer produces the output of the network $\hat{Y} \in \mathbb{R}^{n \times k}$. For classification tasks, this layer often includes a softmax function [68] to interpret the outputs to a probability distribution over predicted output classes, k.

Let us consider an MLP architecture with one hidden layer $h_1 \in \mathbb{R}^{d_{in_1}, d_{out_1}}$, consisting of 16 neurons ($d_{out_1} = 16$), for the Iris dataset. The choice of d_{out_1} is made arbitrarily. However, it represents a hyperparameter of the model, which could be adjusted for optimization. Accordingly, the input layer will have four neurons to match the dataset's four features, i.e $d_{in_1} = 4$, and the output layer will have three neurons, each corresponding to one of the Iris species classes.

Optimization Training an MLP involves using the training dataset, $\boldsymbol{X}_{train}$ to adjust the network parameters $\boldsymbol{\theta}$ that minimizes the difference between the predicted output $\hat{Y}$ and the actual output $\boldsymbol{Y}$. The MLP's parameters can represented as, $\boldsymbol{\theta} = [\boldsymbol{W}^1, \boldsymbol{b}^1, ..., \boldsymbol{W}^{L+1}, \boldsymbol{b}^{L+1}]$, where L are the total number of hidden layers. Using Eq. 2.1, the forward pass for an input $\boldsymbol{x}_i$ is computed sequentially for all the layers to obtain the final predicted output vector $\hat{\boldsymbol{y}}_i$. Now, the model's parameters are optimized, typically done using backpropagation [69], a method that calculates the gradient of the loss function $\mathcal{L}_\theta$ with respect to $\boldsymbol{\theta}$. The most commonly used loss function for classification is the cross-entropy (CE) [70]:

$$\mathcal{L}(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i) = -\sum_{\mathrm{k=1}}^{\mathrm{K}} \boldsymbol{y}_i^k \log \hat{\boldsymbol{y}}_i^k \tag{2.2}$$

CE increases as the predicted probability ($\hat{\boldsymbol{y}}_i^k$) of sample i for class k, diverges from the actual label ($\boldsymbol{y}_i^k$). The optimization of $\mathcal{L}_\theta(\hat{Y}, Y) = \sum_{i=1}^n \mathcal{L}_\theta(\hat{\boldsymbol{y}}_i, \boldsymbol{y}_i)$ is performed in an iterative manner to update the weights in the opposite direction of the gradient, i.e,

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}_t} \mathcal{L} \tag{2.3}$$

When the updates are applied using small subsets of the training data, known as mini-batches, and with a constant learning rate η, the method is referred to as Stochastic Gradient Descent (SGD) [69]. Other optimization techniques, such

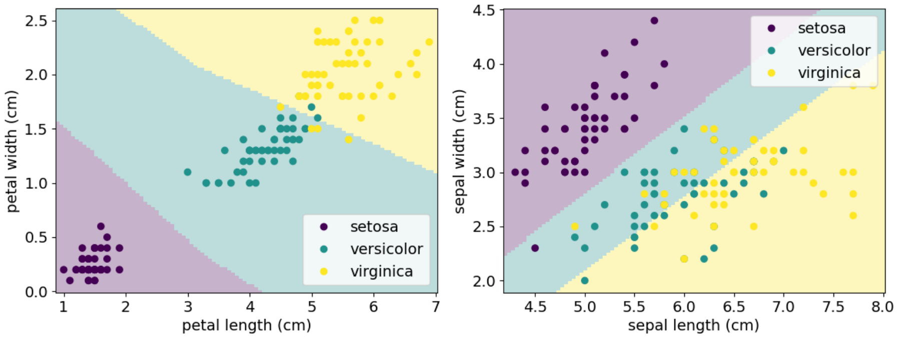


Figure 2.3: Visualization of decision boundaries learned by an MLP using two features as input for Iris dataset.

as Adam, RMSProp and AdaGrad [71] provide more sophisticated mechanisms to adapt learning rate techniques throughout training, leading to more efficient convergence.

Building up on our example of the Iris dataset, we split the data into 80% for training and 20% for testing. We train our three layer MLP model with SGD. To aid in visualizing how the model makes decisions, we train additional MLPs, each using only two of the four input features at a time. This allows us to observe the decision boundaries that the models learn for different feature combinations, as shown in Figure 2.3.

Evaluation Evaluating the performance of an MLP is a critical step in understanding its effectiveness for a given classification task. One of the primary metrics used for this purpose is accuracy, which measures the proportion of correct predictions made by the model out of all predictions [57, 72]. Accuracy is calculated by dividing the number of correct predictions by the total number of predictions, often expressed as a percentage, i.e

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{Total predictions}} \tag{2.4}$$

For the Iris dataset, for instance, accuracy would reflect how often the MLP correctly identifies the species of iris flowers. For the MLP trained with one hidden layer, the model performs with 97.5% of training accuracy and 96.6% of test accuracy, signifying robust performance.

2.2 Unsupervised learning

Unsupervised learning is the branch of ML that operates on unlabeled data; that is, only the input data X is available without any corresponding output labels [73, 74]. Such paradigm is used for discovering inherent structures and patterns within the data autonomously. Common techniques within unsupervised learning include clustering [75], where the algorithm organizes data into clusters based on similarity, and dimensionality reduction [76, 77], which simplifies data by reducing the number of variables under consideration, while still preserving the essential information. In this section, we will focus on clustering due to its relevance to Papers I and IV of this thesis.

2.2.1 Clustering

Clustering, or cluster analysis, is a representative of the unsupervised ML techniques that aims to organize a set of objects into groups, or clusters, such that objects within the same cluster are more similar to each other than to those in different clusters. The similarity is typically assessed based on the features of the objects, which are represented by the input data. A variety of clustering algorithms exist, each with distinct benefits suitable for particular applications. This section will delve into two such algorithms, namely, k-means clustering [78] and spectral clustering [79] owing their relevance to Papers IV and I respectively. This is then followed by a brief introduction MVC [80], which is pertinent to Paper I.

k-means clustering

The k-means algorithm is a method that partitions a dataset into k distinct groups or clusters, aiming to minimize the total squared distance between the points in each cluster and the cluster's centroid [78]. The loss function for k-means is defined as:

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{k} a_{ij} ||\boldsymbol{x}_i^k - \boldsymbol{\mu}_j||^2 \tag{2.5}$$

where n represents the number of samples in the dataset, k is the number of desired clusters, a_{ij} is the assignment of sample i to cluster j (corresponding to 1 if $\boldsymbol{x}_j \in j$ cluster and 0 otherwise), $\boldsymbol{x}_i$ is the i-th data point, and $\boldsymbol{\mu}_j$ is the centroid of cluster j. The algorithm begins by initializing k centroids, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, ..., \boldsymbol{\mu}_k \in \mathbb{R}^d$, and iteratively performs the following steps until convergence:

1. Assignment: Each data point $\boldsymbol{x}_i$ is assigned to the closest centroid j, based on the minimum distance criterion:

$$a_{ij} = \arg\min_j ||\boldsymbol{x}_i^k - \boldsymbol{\mu}_j||^2 \tag{2.6}$$

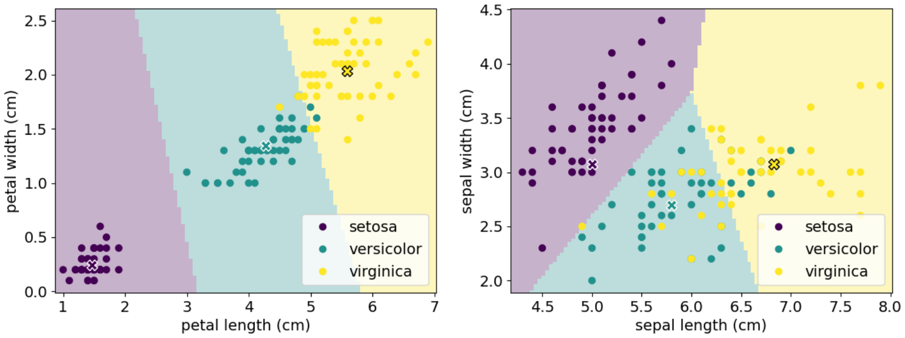


Figure 2.4: Visualization of clusters learned by k-means using two features for Iris dataset, with cluster centers represented by a 'x'.

2. Update: The centroids are updated to be the mean of all points assigned to their cluster:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} a_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{n} a_{ij}} \tag{2.7}$$

Applying k-means to the Iris dataset with $k = 3$, we achieve an accuracy of 88.6% across the entire dataset without the use of any labels for training, thus showcasing the effectiveness of clustering algorithms in unsupervised learning tasks. The clustering results for the Iris dataset, using two features for visualization, are presented in Figure 2.4.

The k-means algorithm is recognized for its simplicity and computational efficiency. However, one of its limitations is the requirement to predefine the number of clusters, k, which may not always be known a priori and can affect the outcome of the clustering [81].

Spectral clustering

This technique uses the eigenvalues of a similarity matrix to reduce dimensionality before clustering the data [79]. The process typically involves the following steps:

1. Similarity Matrix Formation: Create a similarity graph where nodes represent data point and edges are weighted by a measure of similarity between the corresponding nodes. Construct an adjacency matrix $\boldsymbol{A}$, from the similarity graph using a function such as, a Gaussian kernel.

2. Laplacian Matrix Calculation: Compute the graph Laplacian matrix, as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is a diagonal matrix where each diagonal element

d_{ii} is the sum of the weights of all edges connected to i.

3. Dimensionality Reduction: Perform eigenvalue decomposition of $\boldsymbol{L}$, followed by selecting a subset of the eigenvectors (based on the smallest eigenvalues) to form a new feature space. The number of selected eigenvectors corresponds to the desired number of clusters.

4. Clustering in Reduced Space: Use the new feature space to perform clustering. The clusters obtained in this reduced space are mapped back to the original data points, providing the final clustering result.

Spectral clustering is particularly useful when the structure of the individual clusters is highly non-convex, or when the clusters are separated by non-linear boundaries. It is also effective in identifying clusters based on the graph connectivity properties rather than using the Euclidean distance, therefore not necessarily conforming to assumptions of isotropic clusters that methods like k-means rely on.

Multi-View Clustering

MVC is an advanced ML technique that aims to integrate information from multiple distinct feature sets, or “views”, to improve the quality of clustering by utilizing multi-view data [80]. Each view represents a different set of features that describe the data, and these views can originate from various sources or modalities, such as text, images, or sensor data. The central premise of MVC is that by leveraging the complementary information available across different views, one can achieve a more robust and accurate partitioning of the data into clusters than by using any single view alone. A large number of MVC algorithms exist in literature methods such as co-training approaches [82, 83], which refine clusters by maximizing inter-view agreement; multiple kernel learning approaches [84, 85] that merge kernels from different views to enhance clustering; subspace learning approaches that seek a shared latent space for joint data representation and clustering. [86, 87]; and DL based approaches that employ neural networks to extract complex, non-linear features to achieve MVC [88, 89].

In the next chapter, we will delve into more sophisticated models known as deep neural networks, expanding upon the foundational concepts of MLPs discussed thus far.

3

Deep Learning

Deep Learning, a specialized branch of ML, which utilizes neural networks with multiple layers—referred to as "deep" architectures—to uncover intricate patterns within data [18]. These advanced models are designed to autonomously learn high-level abstractions from raw inputs by progressively extracting features at various levels of complexity. This hierarchical feature extraction process has catalyzed significant breakthroughs across numerous domains, including computer vision [90], NLP [23], and autonomous systems [34].

An MLP, as introduced in the preceding chapter, with multiple hidden layers, serves as an instance of a deep neural network. Its depth is a function of the number of hidden layers it contains; more layers signify a deeper network. This depth enables the models to discern more complex relationships within the data [18, 91]. To illustrate, consider the Iris dataset example from the previous chapter: by increasing the number of hidden layers in an MLP from one to four (h_1 to h_4) and adjusting the number of neurons to $d_{out_1} = 8, d_{out_2} = 16, d_{out_3} = 8, d_{out_4} = 4$ for the new layers, the network gains the ability to learn more intricate decision boundaries, as depicted in Figure 3.1.

In this chapter, we delve into various DL architectures that are integral to the research presented in this thesis. Specifically, we will discuss Convolutional Neural Networks (CNNs) in Section 3.1, which are widely used for the classification of spatial data such as images. This is followed by VAEs in Section 3.2, which are generative models that use a probabilistic approach to encode input data. Finally, we conclude with an overview of Transformers in Section 3.3,

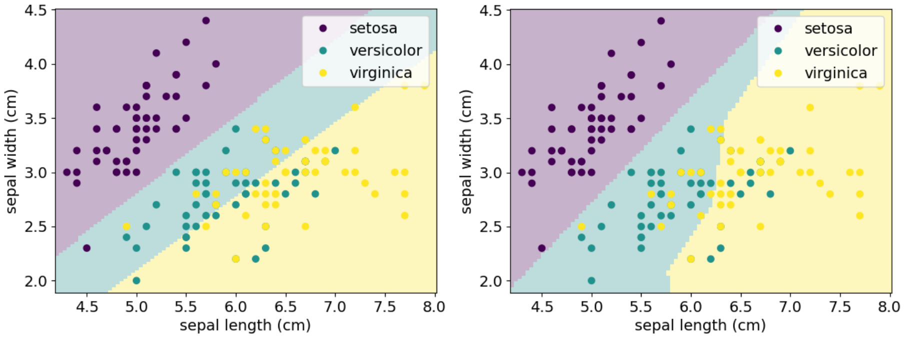


Figure 3.1: Visualization of decision boundaries learned by a shallow MLP with one hidden layers (left) and a deep MLP with 4 hidden layers (right), using two features as input for Iris dataset.

which are attention-based models that have revolutionized the field of language modeling [22].

3.1 Convolutional Neural Networks

CNNs are specialized neural networks, predominantly used for visual analysis due to their ability to efficiently process and learn from image data [92]. Their proficiency in handling visual information has resulted in achieving high accuracy in applications like image classification [93, 94] and object detection [25, 95]. The architecture of a CNN is designed to learn spatial hierarchies of features from images through three main types of layers:

1. Convolution layer: These are the core building blocks of a CNN. They apply a set of learnable filters to the input image to create feature maps. As a filter slides (or convolves) across the image, a two-dimensional activation map is created that gives the responses of that filter at every spatial position. Let $\boldsymbol{W}$ be a single filter matrix; the forward propagation for this filter can be expressed as:

$$y^{l}_{i,j} = \boldsymbol{g}\left(\sum_{m=-(a-1)/2}^{(a-1)/2} \sum_{n=-(b-1)/2}^{(b-1)/2} \boldsymbol{W}^{l}_{m+(a-1)/2,n+(b-1)/2} \; y^{l-1}_{i+m,j+n} + \boldsymbol{b}^{l} \right) \tag{3.1}$$

   This illustrates that the output at a specific location (i, j) in the l^{th} layer is the sum of the element-wise multiplication of the filter matrix $\boldsymbol{W}$ of size a, b and the region in the $(l-1)^{th}$ layer's output (y^{l-1}) corresponding

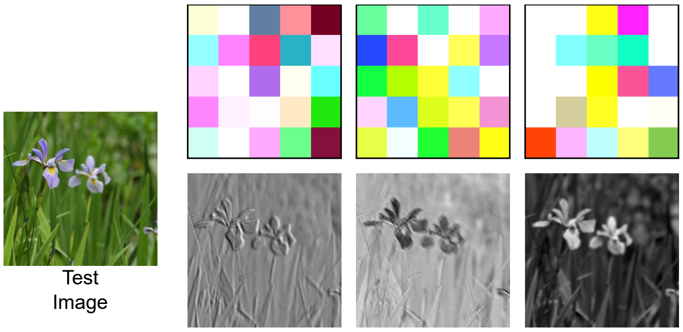


Figure 3.2: Illustration of convolution operation on an image from Iris dataset. The top row shows three filters of size 5 × 5, each possessing three channels corresponding to the RGB channels of the input image. These are picked at random from the first layer of a CNN consisting four convolutional layers (number of filters = $[32, 16, 8, 8]$), three max-pooling layers, and three fully connected layers (number of neurons = $[1000, 100, 3]$). The bottom row shows the test image convolved with the filters.

to the filter's location. $\boldsymbol{b}^l$ corresponds to the bias term at layer l and $\boldsymbol{g}$ corresponds to the activation function. This process is repeated for every spatial location on the input, resulting in a feature map that captures the spatial hierarchies in the input image.

An example of convolution performed on an image from Iris dataset with three filters, chosen randomly, of size 5×5 learned for each input channel by the first convolutional layer of a CNN are shown in Figure 3.2.

2. Pooling layer: Following the convolutional layers, pooling layers reduce the spatial size of the of the feature maps. This reduction not only enlarges the receptive field of subsequent layers but also instills a degree of spatial invariance, as well decreasing the number of parameters in the model, thereby enhancing the model's ability to generalize. The most common pooling operation is max pooling [96], represented as:

$$y^l_{i,j} = \max_{m=-(a-1)/2}^{(a-1)/2} \max_{n=-(b-1)/2}^{(b-1)/2} y^{l-1}_{i+m,j+n} \tag{3.2}$$

   Here, the output y at a location (i, j) in layer l is the maximum value in the spatial neighborhood of the corresponding location in the previous layer $(l-1)$.

3. Fully connected layers: Subsequent to a series of convolutional (optionally followed by pooling layers), fully connected layers, also known as MLPs, are employed for high-level reasoning. These layers usually make the final classification decision, utilizing the high-level features extracted by the preceding convolutional and pooling layers.

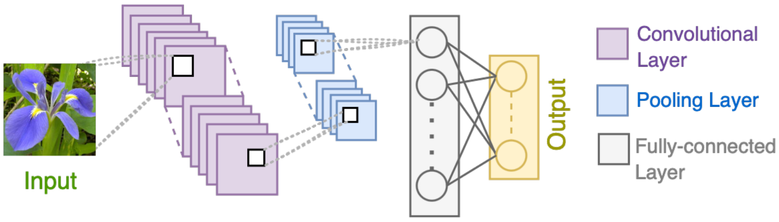


Figure 3.3: A CNN with three types of layers i.e, convolutional, pooling and fully connected layers.

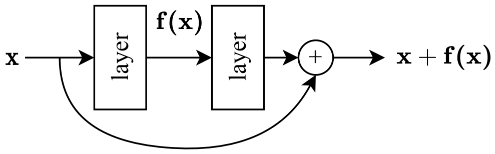


Figure 3.4: Skip connection in the ResNet architecture.

An exemplar architecture with these three layers is shown in Figure 3.3. Additional layers, such as batch normalization [97] and dropout [98], are also often integrated into CNN architectures, enhancing the network's efficacy and its capacity for generalization.

Indeed, several CNN architectures [99] have gained prominence over the past few years due to their performance in various computer vision tasks including, LeNet-5 [100], AlexNet [93], VGGNet [101], GoogleNet [102], ResNet [103] and DenseNet [104]. We briefly review ResNet in this section due to its relevance to Papers I to IV of this thesis.

ResNet The ResNet architecture, developed by Kaiming He et al. [103], represents a significant breakthrough in the field of DL. Prior to ResNet, training extremely deep networks was challenging due to issues such as vanishing gradients, where the gradient signal becomes too small to make meaningful updates to the weights during backpropagation. ResNet addresses this problem by introducing the concept of residual connections. Instead of learning direct mappings from input to output, ResNet layers learn residual functions with reference to the layer inputs. This is achieved through the use of skip connections, which bypass layers by performing identity mapping and adding their outputs to the outputs of the stacked layers, as shown in Figure 3.4. Various ResNet architectures, such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152, have been developed [103], each differing in the number of layers and thus offering a spectrum of complexities and performance levels for diverse applications.

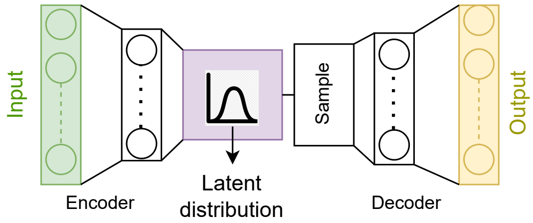


Figure 3.5: Schematic of a VAE.

3.2 Variational Autoencoders

VAEs belong to the category of generative models, which are a class of algorithms capable of creating new data instances that resemble the input data [50]. These models have the ability to generate novel, yet realistic, data samples, making them particularly useful in a wide range of applications, from image synthesis [105, 106] to anomaly detection [107, 108]. The architecture of VAEs is built upon the foundation of autoencoders, a type of artificial neural network that aim to learn a compressed representation of the input data [109]. At their core, autoencoders consist of two main components: an encoder $\boldsymbol{f}$ and a decoder $\boldsymbol{g}$. The encoder's role is to compress the input, $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, into a lower-dimensional latent space, $\boldsymbol{Z} \in \mathbb{R}^{n \times k}$, where n are the number of examples in the dataset, d and k are the dimensions in the input space and latent space, respectively, and $k < d$, i.e,

$$\boldsymbol{z}_i = \boldsymbol{f}_\theta(\boldsymbol{x}_i) \tag{3.3}$$

The decoder then attempts to reconstruct the input from $\boldsymbol{Z}$,

$$\hat{\boldsymbol{x}}_i = \boldsymbol{g}_\phi(\boldsymbol{z}_i) \tag{3.4}$$

The entire network is trained end-to-end by minimizing the difference between the input, $\boldsymbol{x}_i$ and its reconstruction $\hat{\boldsymbol{x}}_i$, typically using a loss function such as mean squared error, i.e:

$$\mathcal{L}_{\theta,\phi} = \frac{1}{n}\sum_{i=1}^{n} ||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||^2 \tag{3.5}$$

A notable variant of the traditional autoencoder is the convolutional autoencoder [110], which incorporates convolutional layers in both the encoder and decoder components. These are particularly effective for tasks involving image data, such as image denoising [111], as they leverage the spatial hierarchy of features through convolutional operations, resulting in more efficient and robust feature extraction.

Building upon the foundation of autoencoders, VAEs introduce a probabilistic framework for the encoding process (Figure 3.5). Instead of encoding an input

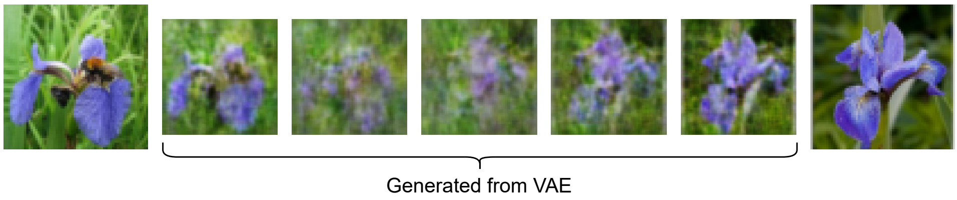


Figure 3.6: Images generated via the decoder of a VAE, trained on Iris dataset, for interpolations between latent vectors of two input images (displayed on the extreme left and right).

as a single point in the latent space, VAEs map the input to a distribution, typically Gaussian, characterized by mean ($\boldsymbol{\mu}$) and variance ($\boldsymbol{\sigma}$) parameters. These parameters are thus the output of the encoder, $\boldsymbol{f}_\theta(\boldsymbol{x}_i) = (\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$. This probabilistic encoding allows for the generation of new samples by sampling from the latent space distributions and mapping them to the input space via the learner decoder $\boldsymbol{g}_\phi$. The training of VAEs involves optimizing not only the reconstruction loss but also a regularization term derived from the Kullback-Leibler divergence (D_{KL}) [50], which encourages the learned distributions to approximate a prior distribution, often chosen to be the standard normal distribution. The loss function therefore looks like:

$$\mathcal{L}_{\theta,\phi} = ||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||^2 + D_{KL}(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)||\mathcal{N}(0_k, \boldsymbol{I}_k)) \tag{3.6}$$

This results in a smooth and continuous latent space with good generalization properties, making VAEs particularly powerful for generative tasks. To illustrate, we present the samples generated from a VAE for the Iris dataset in Figure 3.6. These samples are the result of latent vector interpolations between two images from the training dataset.

3.3 Transformers

Attention-based models in DL have revolutionized the field by enhancing focus on pertinent parts of input data, especially for sequence processing. With attention mechanisms [112], these models excel in tasks with long-range dependencies, outperforming traditional sequential architectures like Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs) [113]. The Transformer architecture exemplifies this, setting new benchmarks in NLP and beyond due to its parallel processing capabilities and its effectiveness in complex tasks such as machine translation and language modeling [114].

Transformers avoid the recurrent layers used in previous sequence-to-sequence models and instead rely entirely on a mechanism known as self-attention to

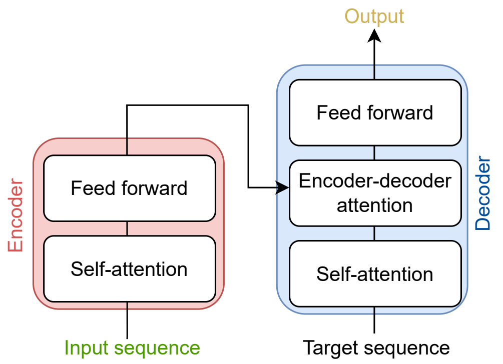


Figure 3.7: Schematic of the Transformer architecture.

draw global dependencies between input and output. It works by first creating a Query (Q), Key (K), and Value (V) vectors from the embedding vectors of the input (X), using weight matrices W^q, W^k, and W^v, respectively, via:

$$Q = XW^q, \quad K = XW^k, \quad V = XW^v \tag{3.7}$$

The self-attention is then calculated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3.8}$$

where d_k is the dimension of the key vectors. Self-attention allows each position in a sequence to attend to all positions within the same layer, enabling the model to dynamically weigh and integrate information from different parts of the input, capturing intricate interdependencies regardless of their distance in the sequence. This is particularly useful for tasks where understanding the relationship between words in a sentence is crucial, such as language understanding. The architecture of Transformer consists of two main components (Figure 3.7):

1. Encoder: The encoder maps an input sequence of symbolic representations, also called as tokens (i.e, words, phrases, or symbols, or other meaningful elements of text) to a sequence of continuous representations. It is composed of a stack of identical layers, each containing two sub-layers: a multi-head self-attention mechanism and a simple, position-wise fully connected feed-forward network.

2. Decoder: The decoder is also composed of a stack of identical layers. In addition to the two sub-layers found in the encoder, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. This allows each position in the decoder to attend to all positions in the input sequence.

Since their introduction, transformers have led to the creation of numerous influential models such as Bidirectional Encoder Representations from Transformers (BERT) [115] and GPT [22] for NLP, and extending their application to images with adaptations like the Vision Transformer (ViTs) [116]). We discuss GPT here, considering its relevance to Paper V.

3.3.1 GPT

The GPT series, developed by OpenAI [22], represents a set of advanced language models using the transformer architecture, specifically its decoder component, for a variety of NLP applications. GPT modifies the original architecture by using only the transformer's decoder stack for its tasks. Unlike the bidirectional context used by the transformer encoder, GPT's decoder operates in an autoregressive manner, predicting the next token in a sequence given all the previous tokens, making it inherently unidirectional. This setup aligns well with language modeling and generation tasks, where the goal is to produce a coherent continuation from a given text prompt. Pretrained on vast datasets, GPT models gain extensive knowledge of linguistic patterns, enabling nuanced understanding and creation of contextually rich text. With each iteration, from GPT-1 through GPT-4, the models have grown with increases in parameter count, complexity, and learning capabilities, resulting in progressively more advanced text generation and better handling of diverse NLP tasks.

Various strategies have been employed to tailor GPT, as well as other pre-trained LLMs [**workshop2023bloom**, 117], to execute several downstream tasks. These include:

- Zero-shot learning: It refers to the model's ability to understand and execute tasks it has not been explicitly trained on, relying solely on its pretraining to generalize from seen to unseen tasks [118]. This therefore showcases the model's generalization capabilities.

- In-context learning: It involves guiding the model using provided examples within the prompt to infer the task and generate appropriate responses without any gradient updates or further training [118, 119]. This demonstrates the model's ability to quickly adapt with minimal data.

- Fine-tuning: It is a more traditional approach where the pretrained model is further trained (i.e., its weights are updated) on a specific dataset to specialize its responses to a particular domain or task.

The models addressed in this chapter are typically characterized by their complexity, with their architectures often encompassing millions to trillions of parameters. This high dimensionality and intricacy render them unexplainable, posing a significant challenge in understanding their inner workings and decision-making processes. The following chapter will delve deeper into this area of research, exploring potential strategies and methodologies to enhance the explainability of these complex models.

Part II

Transparency and Explainability

4

Explainable AI

The intricate nature of DL models, characterized by millions of parameters, often results in them being perceived as black-box entities. This lack of transparency can give rise to numerous challenges, particularly in domains where safety is paramount [120–122]. The inability to comprehend the basis of the model's decisions can lead to issues such as unpredictability, lack of trust, and potential bias in decision-making [123]. These problems could have serious implications, ranging from incorrect predictions to ethical concerns. This existing gap in DL research has led to the advent of XAI [124]. XAI aims to make the decision-making process of AI models transparent and understandable, thereby enhancing the models' reliability and accountability.

XAI methods are developed to answer the question of *why* in addition to the original goal of the DL algorithm, for e.g., *what* in context of classification problems. The primary issue with black-box models is their complexity and the incomprehensibility of their parameters, which are not in a form that humans can easily understand. While the answer to *what* is often presented in a human-understandable form, such as class probabilities, the *why* behind these decisions also needs to be comprehensible. XAI strives to bridge this gap between AI's "language" and human understanding. An intuition of this is provided in Figure 4.1. XAI methods typically accomplish this by identifying and highlighting the features or factors in the input space that influence the decision, thereby making them understandable to humans. This approach demystifies the decision-making process of AI models.

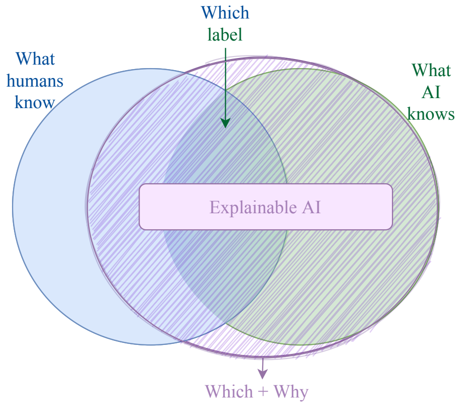


Figure 4.1: Explainable AI expands the "language" of AI – what AI knows – to have overlap with language of humans, thereby providing the reasoning of *why* in addition to *which* in human-understandable form.

XAI approaches have been developed in two parallel branches of 1) post-hoc methods, and 2) Self-Explainable Models (SEMs). Post-hoc methods aim to elucidate the behavior of an existing black-box model after its operation, providing retrospective explanations [125, 126]. On the other hand, SEMs strive to offer answers to both *what* and *why* concurrently, making them inherently interpretable [42, 127]. However, these methods differ not only in their approach but also in the type of explanations they provide. To understand this better, we will first discuss the different kinds of explanations, particularly in the context of image classification. Subsequently, we will review how different methods employ diverse approaches to generate these explanations.

4.1 Explanations for deep learning models

To provide answers to the *why* question, it is crucial that the explanations are delivered in a format that is easily understandable by humans. We start from a broad perspective of *locality* of explanations and then go deeper to understand several explanation methods. On the basis of said *locality*, the explanations can be *local* or *global*. While local explanations focus on explaining the effect of a prediction from a model on a single instance, global explanations aim to provide general explanations for the model. Various methods to accomplish this have been explored in recent literature, and these will be the focus of our discussion in this section. We review several methodologies in this section, followed by the review of post-hoc and self-explainable models in Sections 4.2 and 4.3, respectively, grounded in the argument of *locality*. A visualization of the proposed taxonomy is provided in Figure 4.2.

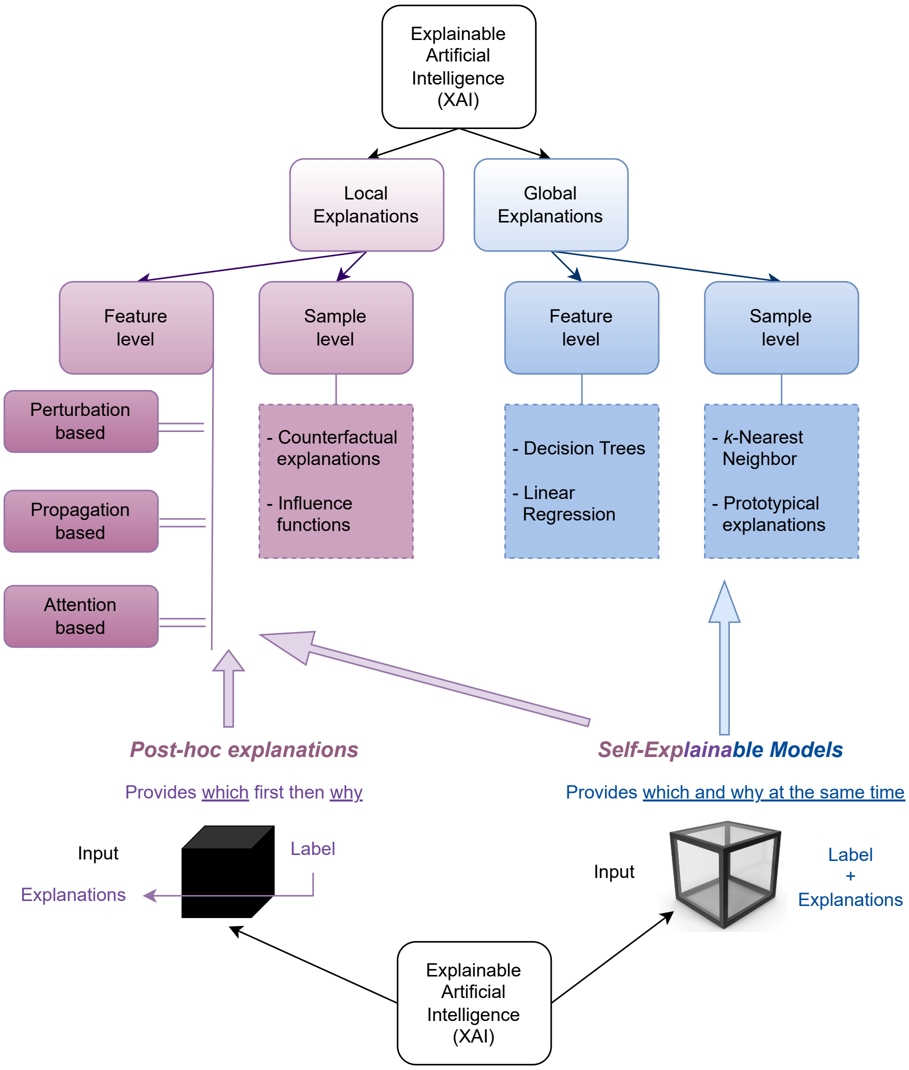


Figure 4.2: Taxonomy of XAI based on *locality* of explanations (from top to bottom) and the nature of explanations (from bottom to top). While post-hoc explanations produce local explanations, self-explainable models are capable of generating both global as well as local explanations.

4.1.1 Local explanations

Local explanations focus on interpreting the impact of the model's decision on a specific individual sample or example [128]. These help answer the question "why did the model make this particular prediction for this instance?" Local explanations can produce feature-level attributions or sample-level explanations, as explained below.

Feature-level explanations

Feature-level explanations play a crucial role in understanding the decision-making process of complex models. These explanations help in interpreting and justifying the output of the model by identifying the specific features that significantly influenced the prediction [129]. In image analysis, this often involves assessing the impact of specific pixels or pixel clusters [130]. In NLP, the focus shifts to the words or phrases that are most indicative of the model's decisions [131]. For tabular data, the analysis might examine how individual columns in the dataset affect the outcome [132]. The upcoming discussion will delve into a range of explainability techniques, starting with perturbation-based methods that observe changes in predictions when input features are altered. Following this, we will discuss propagation-based methods such as gradient propagation, that trace the flow of information through the model, with a specific focus on a method called Layer-wise Relevance Propagation (LRP) which is the foundation for Paper I of this thesis. Lastly, we will examine attention-based methods, which are particularly prevalent in NLP [133] to highlight influential parts of the input text.

- *Perturbation-based:* Local Interpretable Model-agnostic Explanation (LIME) [126] trains a surrogate explainable model, such as a linear regression or a decision tree, in the local neighborhood of an instance of interest, thereby generating feature-based explanations. SHapley Additive exPlanations (SHAP) [134] quantifies the significance of each feature to a model's output. It assesses the individual impact of each feature as it contributes along a pathway, aggregating all feature contributions before averaging them. KernelSHAP uses a weighted local linear regression to estimate SHAP values. KernelSHAP attempts to approximate the results from a complex model using a simpler, interpretable model that is fitted with respect to a kernel. Both LIME and SHAP are model-agnostic XAI methods, operating independently of a model's internal mechanics, unlike other model-aware methods discussed below.

- *Propagation-based:* Multiple methods utilize the network's gradients for computing the effect of a prediction on a sample. For example, saliency maps use backpropagation to compute gradients of the output with respect to the input [135]. Guided backpropagation [136] enhances saliency maps by filtering out negative gradients during the backpropagation process. Integrated gradients [137] measure the cumulative influence of input features on the model's output prediction by integrating the gradients along a straight path from a baseline input to the actual input. Class Activation Maps [138] add a global average pooling layer after the convolutional layers, which averages the features from all previous layers. The resulting vector is then fed into a linear model to get the importance of each feature map to generate class-discriminative saliency maps. Grad-CAM on the other hand, does not require any additional layers to be added. Instead, it uses the gradients of the output neuron with respect to the ReLU feature maps of a convolutional layer to approximate the importance of each spatial location [139]. Guided Grad-CAM is a technique that combines Grad-CAM and Guided Backpropagation to provide fine-grained visual explanations along with class-discriminative features [139]. Grad-CAM++ [140], an extension of the Grad-CAM, calculates the weights by considering the first and second order derivatives to capture the more detailed relationship between the neuron and the class score. It helps to capture not just the 'peak' response but also the 'distribution' of related pixels thus providing more clarity on where the model is looking to make decisions.

LRP [125] another backpropagation based method, is a key component in Papers I and II of this thesis. It works by attributing the prediction of a neural network to its individual input features. LRP stands out among other backpropagation based methods due to its ability to redistribute relevance scores in a layer-wise manner, therefore mitigating gradient shattering effect. At each layer of the network, it computes a relevance score for each neuron which represents the contribution of that neuron towards the final prediction. These relevance scores are then accumulated and assigned to each input feature to indicate their importance in the prediction. The general rule for LRP can be represented as follows [125]:

$$R_j = \sum_k \frac{z_{jk}}{\sum_{j'} z_{j'k}} R_k \tag{4.1}$$

where R_j is the relevance of neuron j, z_{jk} is the contribution of neuron j to neuron k, and the denominator is the sum over all neurons j' that contribute to neuron k.

Several variations of rules have been proposed, catering to different scenarios and different layers of a neural network [141], including, the LRP-ϵ that introduces a stabilizing term in the denominator to avoid numerical instabilities:

$$R_j = \sum_k \frac{z_{jk} + \epsilon \cdot sign(z_{jk})}{\sum_{j'}(z_{j'k} + \epsilon \cdot sign(z_{j'k}))} R_k \tag{4.2}$$

The LRP-γ rule, another variant, introduces a weighting term γ to control the balance between positive and negative contributions:

$$R_j = \sum_k \frac{z_{jk} + \gamma \cdot z_{jk}^-}{\sum_{j'}(z_{j'k} + \gamma \cdot z_{j'k}^-)} R_k \tag{4.3}$$

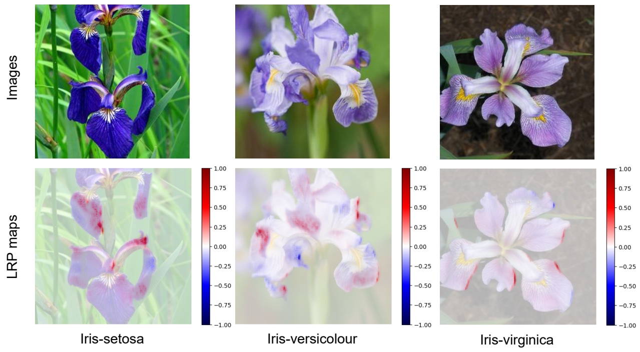


Figure 4.3: Visualization of LRP relevance maps generated for the CNN trained on the Iris dataset. The blue and red regions denote negative and positive relevance to the ground truth class, respectively. As observed, LRP maps are efficient in generating input-wise feature-based attributions, highlighting the decision making of the black-box model.

The LRP-$\alpha\beta$ rule is a more flexible variant that allows different weighting for positive and negative contributions:

$$R_j = \sum_k \left(\frac{\alpha \cdot z_{jk}^+}{\sum_{j'}(\alpha \cdot z_{j'k}^+)} - \frac{\beta \cdot z_{jk}^-}{\sum_{j'}(\beta \cdot z_{j'k}^-)} \right) R_k \tag{4.4}$$

In Figure 4.3, we visualize LRP maps for the CNN trained in the previous chapter on the Iris dataset. For this example, we use the LRP-$\alpha\beta$ rule for convolutional layers with $\alpha = 1$ and $\beta = 0$, and the LRP-ϵ rule for the fully connected layers. LRP offers several benefits over other techniques, such as its versatility and applicability to numerous neural network architectures, along with the capability to provide fine-grained attributions by assigning importance to each input feature. Furthermore, LRP adheres to the 'conservation principle', ensuring the sum of the input layer's prediction explanation matches the network's pre-softmax output.

- *Attention-based:* Attention maps are a feature of certain types of ML models, particularly relevant in models like Transformers [114] and attention-based RNNs used for sequence prediction tasks such as language translation and text generation. An attention map visually represents which parts of the input data the model has 'attended to' or focused on while making a particular prediction [133]. For local explanations, attention maps are especially insightful. For instance, in a sequence-to-sequence translation task, attention maps can vividly illustrate which words in the source sentence the model paid attention to when translating it into the target sentence. Thus, they assist in understanding how the model is associating words or phrases across different languages.

Sample-level explanations

Sample-level local explanations provide or generate data samples as explanations for individual instances. For example, counterfactual explanations [142] are a form of model interpretation that provides insights into how a model's output would change if the inputs were altered in specific ways. Consider the instance of a model trained to classify images of animals. Given an image of a dog misclassified as a cat, a counterfactual explanation would involve modifying the image until the model correctly classifies it as a dog. Other methods include influence functions, which can trace a model's prediction back to the training data, identifying which samples were most influential in determining the outcome for a specific test instance [143].

4.1.2 Global explanations

Global explanations provide a broader understanding of the model's behavior and decision-making process across the entire dataset. Such explanations offer insights into the overall trends, patterns, and feature importance that the model has learned from the data. Methods like feature importance analysis [60], class-representative extraction [42, 127], and rule extraction techniques [67, 144] can be employed to generate global explanations. These approaches help identify the most influential features across the entire dataset, providing a comprehensive understanding of how the model generally operates and what features it deems significant for making predictions [128]. Global explanations can be on the feature-level or sample-level, as described below.

Feature-level explanations

Linear regression [60] and decision trees [67] are two ML methods that inherently provide global explanations of model predictions. In the case of linear regression, the global explanation is provided through the model's coefficients. Each coefficient represents the average change in the output variable given a one-unit change in the corresponding input feature, assuming all other features are held constant. This provides a straightforward interpretation of the overall importance of each feature in making predictions. On the other hand, decision trees provide a global explanation via the structure of the tree itself [132]. The decision tree algorithm chooses the features that best split the data at each node based on a certain criterion (e.g., Gini impurity or entropy for classification, variance reduction for regression). Features that appear closer to the root of the tree generally have a more significant impact on the output variable, thus providing an intuitive visual summary of the features' importance in the model's decisions.

Additionally, in terms of global explanations, attention maps can be aggregated over multiple instances to identify broader patterns in the model's attention [145]. Similarly, to create a global perspective, SHAP values from multiple instances can be combined to derive an overall importance for each feature. This is typically visualized using summary plots that display the distribution of SHAP values for each feature across all data points [134].

Sample-level explanations

As the name suggests, sample-level explanations provides 'data samples' as proxies for explaining the learning of a model. For example, k-Nearest Neighbor (k-NN) [146] is a simple, intuitive ML algorithm that can serve as a global explainer for understanding complex models. It works by classifying new instances based on their similarity to existing instances in the training dataset. In the context of interpretability, it can thus provide a global explanation by showing how data points are grouped together based on their feature similarities, and how these groupings relate to the output variable. Another method for feature-visualization involves optimization based visualization. For example, in activation maximization [147], a noisy image is optimized to maximize the activation value of a specific neuron or class via techniques like gradient ascent in the input space.

Prototypical explainable models are DL models that seek to provide explanations for their predictions based on class-representatives or prototypes [42]. In the process of making a prediction, these models do not just map an input to an output, instead, they also identify the prototype that is closest to a given input data point, followed by basing the prediction on this prototype. Using prototypes to explain predictions has a significant advantage as it relates to how humans naturally understand categories. We often explain categories and concepts to each other via exemplars or prototypical representations, thus making prototypical explainable models highly interpretable and intuitively comprehensible.

4.2 Post-hoc methods and local explanations

Post-hoc explanation methods are concerned with interpreting the decision-making processes of already trained, black-box DL models. These techniques are mostly oriented towards providing local explanations, which illuminate the reasoning behind individual predictions made by the model. Among the array of techniques explored in Section 4.1.1, with the exception of attention mechanisms, local explanation strategies generally form a subset of post-hoc methods.

These strategies are applied after the model's training phase to demystify specific outputs rather than offering a global understanding of the model's overall behavior. However, while these methods can offer valuable insights, they can sometimes be unreliable due to their detachment from the model's actual decision process. For instance, they may rely on approximations or assumptions that do not fully capture the model's complex internal dynamics, leading to explanations that are not entirely faithful to the reasons behind a model's predictions [148, 149]. Additionally, the quality of post-hoc explanations can be sensitive to the choice of parameters or the specific data used, potentially resulting in inconsistent or misleading interpretations across different scenarios [36, 41, 150].

4.3 Self-Explainable Models: local and global explanations

SEMs are inherently interpretable, integrating transparency within their architecture. This inherent transparency ensures that the explanations are directly tied to the model's computations. Thus, SEMs are designed as "glass-box" models, with their inner workings accessible and understandable, as depicted in Figure 4.2. Since the decision-making process of SEMs mirrors human decision-making, as opposed to the post-hoc models, SEMs are considered to be more reliable with less misunderstanding and easier control based on human inputs [36, 144]. Because the interpretability of SEMs is built-in, the explanations they provide are more likely to accurately reflect the true reasoning of the model. Further, SEMs offer consistent explanations since their interpretability is part of their structure and not dependent on separate algorithms or additional parameters that can vary between uses, as is the case with post-hoc methods [151]. Using SEMs can simplify the workflow for developers and end-users, as there is no need to apply and interpret separate post-hoc explanation methods. This can reduce the complexity of deploying DL systems and make it easier for users to trust and understand the model's outputs. The transparency of SEMs can facilitate debugging and model improvement. When developers can see how the model arrives at its conclusions, they can more easily identify and correct errors or biases in the model, as discussed in Chapter 5. In summary, SEMs offer a more direct and reliable route to model interpretability, which can enhance trust, compliance, and the overall utility of DL systems.

While most global explainers previously discussed in Section 4.1.2 fall into the category of SEMs, some SEMs go beyond offering global insights and are also capable of producing local, instance-wise feature attribution maps. These dual-capability models provide a comprehensive understanding of model predictions, both at the individual and aggregate levels.

Many SEMs have been proposed in the literature, for example, [144] learns deep logic rule reasoning by leveraging global level human priors about rules (e.g., desirable form and property of candidate atoms) and generate explanations by optimizing rule confidence, approximated using the training data. However, in this work, the generation of atoms (smallest level of explanations, as proposed by authors), remains manual, which can affect both the accuracy as well as the interpretability of the model. Other line of work follows generalized additive models [152], where each feature's shape is learned independently, followed by their addition to learn complex models. Following this, [153] introduces Neural Additive Models which are trained jointly to learn a complex relationship among several linear combination of neural networks that each attend to a single input feature. However, these require deep neural networks with tens of thousands of parameters. To remedy this, Radenovic et al. [154] proposes Neural Basis Model, where shared basis functions are learned for all features, thereby reducing computational complexity.

Other line of work in SEMs follow the learning of class-representative concepts or prototypes for achieving interpretability [42, 127], referred to here as Prototypical Self-Explainable Models.

4.3.1 Prototypical Self-Explainable Models

Prototypical SEMs are ML models that inherently provide explanations for their predictions based on prototypes or representative examples. As an intrinsic part of their architecture, they relate new instances to the prototypes they have learned during training. This allows for interpretability, as their predictions are based on comparing a new instance to these representative examples, providing a recognizable point of reference. These prototypes are considered as global explanations, while for generating instance-based local explanations, post-hoc methodologies have been used in majority of the work in this field [42, 155, 156]. Since this thesis majorly focuses on prototypical SEMs, we delve deeper into these models in this section.

SENN [127] was the first work to propose a general self-explaining neural network architecture, consisting of a concept or prototype encoder, an input-dependent parameterizer that generates relevance scores and an aggregation function that produces a prediction. This structure has been followed by several works on prototypical SEMs, such as ProtoPNet [42], Self Interpretable Transformation Equivariant network (SITE) [156], and Framework to Learn with Interpretation (FLINT) [155]. We discuss these in the following subsections, considering their usability as baselines in Papers I to IV of this thesis.

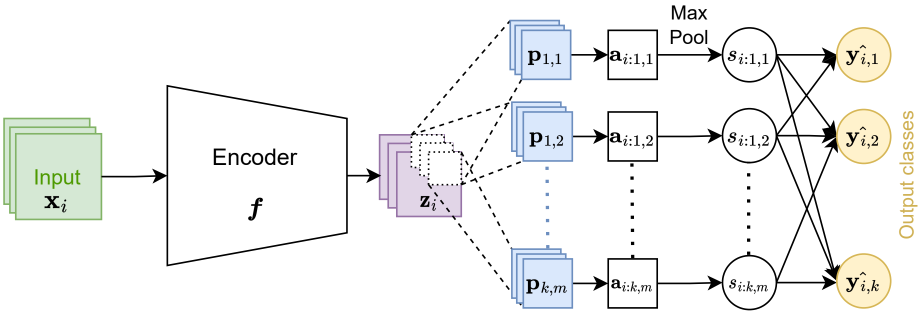


Figure 4.4: Schematic of ProtoPNet architecture.

4.3.2 Prototypical Part Network

Prototypical Part Network (ProtoPNet), introduced in the paper "This looks like that" [42], is an example of a prototypical SEM designed to provide intuitive and interpretable explanations for its decisions. It is one of the state-of-the-art SEMs, which has been followed by several line of works [157–161]. It is a type of neural network trained to make predictions based on learned prototypes. ProtoPNet learns a dictionary of m prototypes for all classes k $\boldsymbol{P} = \{\boldsymbol{p}_{c=1,j=1}^{k,m}\} \in \mathbb{R}^{h_p \times w_p \times d_p}$, each with height h_p, width w_p and depth d_p, which are representative examples of different output classes. The architecture of ProtoPNet consists of a convolutional backbone, referred to here as encoder ($\boldsymbol{f}$), followed by prototypical layer, and finally a fully connected layer ($\boldsymbol{h}$) for classification (refer Figure 4.4). During its training, the prototypes are replaced by the closest training image patches, thereby maintaining direct interpretability. ProtoPNet follows the following steps for classification of a new instance $\boldsymbol{x}_i$:

1. Encoding the input via the learned backbone, producing convolutional output $\boldsymbol{z_i}$ as:
$$\boldsymbol{z_i} = \boldsymbol{f}(\boldsymbol{x}_i) \tag{4.5}$$

2. Generating $m \times c$ activation maps $\boldsymbol{a}_i(c, j)$ for $\boldsymbol{z_i}$, corresponding to each prototype, by calculating similarity between all patches, $\tilde{z}_i$, of $\boldsymbol{z_i}$ with the same size as p, i.e
$$\boldsymbol{a}_i(c, j) = \log\left(\frac{||\tilde{z}_i - \boldsymbol{p}_{c,j}||_2^2 + 1}{||\tilde{z}_i - \boldsymbol{p}_{c,j}||_2^2 + \epsilon}\right) \tag{4.6}$$

3. Computing similarity scores $s_i(c, j)$ from $\boldsymbol{a}_i(c, j)$ using max pooling, i.e,
$$s_i(c, j) = \text{MaxPool}(\boldsymbol{a}_i(c, j)) \tag{4.7}$$

4. Finally, converting the similarity scores into class probabilities using the fully connected layer, i.e, $\hat{\boldsymbol{y}}_i = \boldsymbol{h}(\boldsymbol{s}_i)$.

This process thus provides similarity of an instance with all prototypes, along with activation maps, as well as the final class probabilities. This therefore allows the network's predictions to be understood and interpreted in simple terms, i.e, *this* new instance was classified into this class because it closely resembles *that* learned prototype. Additionally, the prototypes themselves can be inspected in the input space and understood by domain experts, providing even deeper insights into how the model is making decisions. ProtoPNet generates global explanations in terms of visualized prototypes in the input space, and local explanations by upsampling the activation maps to the input size.

However, ProtoPNet is not without its limitations. These include, the disconnected training process due to reliability of visualization using training images, the loss in accuracy when compared to the corresponding black-box model and the unreliable local explanations. We address these issues in this thesis by proposing more faithful and precise local explanation method for ProtoPNet in Paper I and novel SEMs in Paper III and IV.

4.3.3 Self Interpretable Tranformation Equivariant network

SITE, the Self Interpretable Tranformation Equivariant network [156], also introduces a novel paradigm where prototypes of input classes are learned while ensuring the transformation-equivariant aspect of model interpretations. The class prototypes are learned via a generative model (G) that maps the latent representation $\boldsymbol{(}z\boldsymbol{)}$ to instance-based prototypes ($\boldsymbol{p}^{n,k}_{i=1,c=1}$), where each prototype corresponds to one specific class. The final classification is determined by the inner product of the prototypes and the latent representation::

$$\hat{\boldsymbol{y}} = \sigma(G(\boldsymbol{z})^T\boldsymbol{z}) \tag{4.8}$$

Here, σ represents the softmax activation, which is used to derive class probabilities. The model achieves transformation equivariance by imposing constraints on the loss to ensure consistency in the explanations (i.e., the learned prototypes) regardless of any transformations applied to the input. Since the prototypes learned by this model are dynamic and generated for each test image, this method is only able to provide local interpretations and lacks global interpretations.

4.3.4 Framework to Learn with Interpretation

Similarly to other prototypical models, the Framework to Learn with Interpretation (FLINT) [155] also learns an attribute or prototype dictionary. However, unlike previous models, it utlizes a dual-model architecture to achieve interpretability. Its architecture consists of: the original predictive model, referred to as FLINT-f, and a newly proposed interpreter model, denoted as FLINT-g. The role of the interpreter model (FLINT-g) is to learn a dictionary of attribute functions or prototypes (Φ) by utilizing the outputs of chosen hidden layers of FLINT-f, typically selected from the latter layers. The output from the learned attribute functions are then forwarded through an interpretable function ($\boldsymbol{h}$), i.e, a single fully connected layer with weights W to get the predicted class probabilities

$$\hat{\boldsymbol{y}}_i = \boldsymbol{h}(\Phi(\boldsymbol{x}_i)) = \sigma(W^T \Phi(\boldsymbol{x}_i)) \tag{4.9}$$

where σ represents the softmax activation [68].

FLINT is able to generate both local as well as global explanations, achieved via utilizing the activation maximization [147] method. However, a trade-off between explainability and accuracy arises in FLINT as its interpreter model is not able to perfectly replicate the performance of the predictor model.

In conclusion, XAI, with its ability to represent complex relationships with both global as well as local explanations can significantly facilitate the creation of transparent models. However, there might exist artifacts and biases in the data which can hinder the development of responsible and fair AI. We discuss this further in the next chapter.

Part III

Responsible and Fair AI

5

Artifact Learning

Artifact learning in DL refers to the scenario where a model inadvertently learns from features within the training data that are irrelevant or misleading — termed as "artifacts". These artifacts can significantly distort the learning process, leading the model to base its predictions on spurious correlations rather than on meaningful attributes [44]. This reliance on non-essential features can result in a model that is not only less accurate but also less generalizable to new data, as it may fail to recognize the correct patterns when the artifacts are absent.

Consider an exemplar scenario in which a ML model is being trained to classify images into categories of dogs and wolves. It might be observed that within the training dataset, a majority of the wolf images are captured within snowy environments, while images of dogs predominantly feature non-snowy settings. If the model begins to associate the presence of snow with the classification of an image as a wolf, it has inadvertently engaged in artifact learning. In this context, the snow acts as an artifact rather than a salient feature for the task of distinguishing between the two species. This model, therefore, acts as a 'snow detector' instead of the intended dogs vs wolves detector [126]. This misdirected learning underscores the importance of ensuring that models focus on relevant features for prediction rather than extraneous contextual cues that may lead to biased or incorrect generalizations. This scenario exemplifies the implications of artifact learning: it can lead to high accuracy during training or validation but can cause the model to perform poorly on new, unseen data that does not follow the same pattern (for example, a picture of a wolf not in

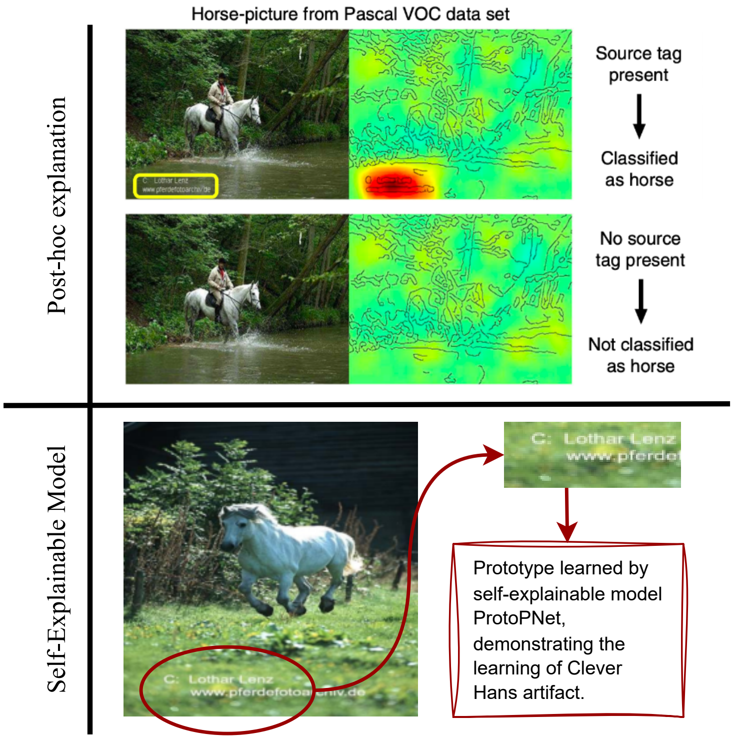

Horse-picture from Pascal VOC data set
Post-hoc explanation
Source tag present
Classified as horse
No source tag present
Not classified as horse
Self-Explainable Model
C: Lothar Lenz
Prototype learned by self-explainable model ProtoPNet, demonstrating the learning of Clever Hans artifact.

Figure 5.1: Clever Hans artifact (a watermark in the Horse class) in Pascal VOC dataset [166] captured by LRP, a post-hoc explanation method applied on a black-box Fisher vector classifier (top) and ProtoPNet, a self-explainable model (bottom). Top image is an example from Lapuschkin et al. [44].

the snow, or a dog in the snow).

Artifact learning can manifest both unintentionally, as a byproduct of the data or model biases [38], and intentionally, often as a result of adversarial intervention [46, 47, 162, 163]. This chapter delves into the multifaceted nature of artifact learning, examining various scenarios where artifacts influence model behavior.

5.1 Unintentional artifact learning

DL models are prone to learning unintentional artifacts in the training data [45, 164]. This tendency to gravitate towards simpler, more superficial solutions can detract from the models' or datasets' ability to address the core complexities of the problem statement [45, 165]. There are many ways in which models can learn these unintended correlations, such as:

- Spectral Bias: DL models tend to learn the low-frequency components of a function before high-frequency ones, which can lead to a preference for smoother functions that may not capture the true underlying patterns [79].

- Dataset imbalances: When training datasets are not representative of the real-world distribution or contain imbalances, models may develop correlations that are artifacts of these imbalances rather than genuine features of the data [164].

- Noise Artifacts: Sometimes, DL models learn the noise present in the training data, which, too, would lead to poor generalization on novel, unseen data. This often happens when highly complex models are trained on small datasets - the models end up learning the data's noise and randomness rather than the underlying patterns [167]. Further, label noise can exist where inaccurate or inconsistent labeling can introduce misleading correlations that DL models might learn, resulting in a divergence from the true signal that the model is intended to capture [168].

- Confounding Variable Artifacts: These occur when the models learn a correlation between the target and input features that is actually driven by a hidden factor. A famous example of this is the "Clever Hans" artifact.

  - Clever Hans artifact: The term "Clever Hans" artifact is derived from a famous horse named Clever Hans that appeared to understand complex human language, including solving mathematical problems, but was later discovered to be reacting to subtle cues from his human handlers [43]. In DL, Clever Hans artifact learning refers to scenarios where models seem to be doing the right thing for the wrong reasons, thereby essentially reacting to unintended cues in the data. For example, a model may be trained to classify images of horses and it may have a high success rate in doing so. However, upon deeper investigation, it could be found that the model is not identifying visual clues with respect to a horse, but rather associating the background of the image (a watermark from the photographer in this case) with the horse class [44] (see Figure 5.1). Similarly, it was noticed by the authors in [44] that if the same watermark is added on an image of a car, it is now classified as a horse. These misleading cues can thus lead to poor generalization to real-world, unseen data. Additionally, in safety-critical scenarios, such as healthcare, this kind of artifact learning can lead to serious complications when, as an example, a disease is predicted based on 'wrong' or spurious correlations in the data [164]. This therefore emphasizes the use of XAI, such that these intended as well as unintended correlations do not go unnoticed. In Figure 5.1, we demonstrate the capture of a Clever Hans artifact by a post-hoc XAI method (LRP) and an SEM (ProtoPNet).

5.2 Intentional artifact learning

Intentional artifact learning in DL occurs when a model is purposefully compelled to learn from certain artifacts. This deliberate process is often employed in scenarios where understanding or mitigating the impact of these artifacts is crucial. Examples include adversarial attacks [46, 47], where models are exposed to subtly modified inputs designed to cause misclassification. By training on these adversarial examples, models can develop a resistance to such attacks. Other instances might involve watermarking techniques for digital rights management, where models need to detect specific patterns signifying ownership or authenticity [169]. Additionally, intentional artifact learning can be used in domain adaptation, helping models to recognize and adjust to domain-specific cues that would otherwise be considered noise [170]. In the following sub-section, we delve deeper into adversarial attacks, considering the applicability of Backdoor attacks (a subset of adversarial attacks) to Paper I of this thesis.

5.2.1 Adversarial attacks

A significant challenge in DL is the susceptibility of models to adversarial attacks, a vulnerability that is exacerbated by the black-box nature of these models [46]. Adversarial attacks in DL involve the intentional and strategic manipulation of input data with the aim of misleading machine learning models into making erroneous predictions. These manipulations, known as adversarial perturbations, are typically designed to be subtle to evade human detection, yet they are capable of inducing profound deviations in the model's output [47]. Adversarial attacks, with respect to intentional artifact learning in DL, encompass a variety of techniques, including but not limited to:

1. Data Poisoning Attack: In this attack, the adversary introduces incorrect data into the training dataset to skew the final results [171]. The main aim of this attack is to impact the overall performance of the targeted model.

2. Adversarial Examples: These are subtly modified inputs designed to confuse ML models into making incorrect predictions [172]. An example might be perturbing pixel values in an image just enough to trick an

image classifier into mistaking a cat for a dog while remaining nearly identical to the human eye.

3. Backdoor attacks: In the realm of ML, “backdoor” artifacts refer to the specific vulnerabilities, patterns or features in a model, that are intentionally designed or inserted by a malicious entity, as a covert way to control the model’s decisions or behavior. These malicious entities can be attackers who poison the training data by introducing these special patterns or features, often known as “trigger" into the dataset. When a model encounters this specific trigger in the inputs, it causes the model to produce incorrect outputs or behave in ways that serve the attacker’s intent [46]. For instance, in an image recognition system, a backdoor artifact might be a specific logo hidden in the image. The model could then be manipulated to incorrectly classify any image with this logo as a specific category, regardless of the actual content. This can pose serious security threats and cause reliability issues, particularly in critical systems like self-driving cars, where a stop sign can be manipulated to be recognized as a speed limit sign [173], or facial recognition systems [174]. Therefore, it becomes crucial to deploy defenses and countermeasures such as robust training methods and model interrogation techniques to mitigate their impact.

5.3 XAI and artifact detection

XAI plays a pivotal role in addressing the challenge of artifact learning in ML models. By providing transparency and interpretability, XAI enables researchers and practitioners to understand the decision-making processes of complex models, uncovering the reasons behind specific predictions [42, 126, 141]. This insight is crucial for identifying when a model has learned to rely on spurious correlations or artifacts rather than the substantive characteristics that genuinely inform the task at hand [44, 45, 164].

Recent research has concentrated on leveraging post-hoc explanations to detect spurious learning, with several studies documenting the efficacy of these methods [44, 45, 175]. Nonetheless, there is an ongoing debate regarding the limitations of post-hoc explanations in accurately fulfilling this role [150]. Further, given the concerns about the faithfulness of post-hoc explanations, this thesis advocates for the adoption of SEMs as a more reliable alternative for the detection of artifact learning. The effectiveness of SEMs in identifying and mitigating artifact learning is demonstrated in Paper I and Paper II.

As an example, as shown in Figure 5.1, ProtoPNet, in particular, has the capability to globally capture artifact learning, which facilitates the easier identification of artifacts across the model. This global perspective contrasts with post-hoc explanation methods, which typically only provide local explanations for individual predictions. Local explanations, while useful, may not always give a comprehensive view of the model's reliance on artifacts, potentially making the detection process more laborious and less systematic.

6

Bias and fairness

Biases in DL represent systematic deviations that can significantly impact the performance and fairness of neural network models [38]. These biases often originate from the data used to train the models, which may encapsulate historical disparities, societal stereotypes, or sampling that is not reflective of the broader population [48, 176]. For example, a facial recognition system trained predominantly on images of individuals from a single ethnic group may exhibit reduced accuracy for people of other ethnicities, illustrating *dataset bias* [177].

Algorithmic bias arises when the assumptions embedded within learning algorithms inherently favor certain patterns or outcomes. This can lead to models that are predisposed to specific decisions, irrespective of the representative power of the data [178, 179]. This type of bias is thus introduced by the algorithm itself, often through the assumptions made during the development process. For example, an algorithm might be biased towards simpler patterns if it has a complexity penalty [165].

Confirmation bias is another concern, where the preconceptions of researchers or developers may inadvertently guide the selection of data or the fine-tuning of models, thereby reinforcing existing beliefs or hypotheses [180, 181].

The ramifications of such biases are profound, especially in high-stakes domains such as recruitment, credit scoring, and criminal justice, where they can lead to outcomes that are unjust or discriminatory. Some real-world examples of AI doing harm due to biases include: Unlawful and unethical use of facial recognition software by UK police [182], amazon's recruiting tool showing bias against women [183], automated anti-blackness implemented by facial recognition AI in New York [184], Apple's credit card's gender bias [185] and Microsoft's chatbot learning racism through Twitter [186]. To combat these biases, the field of DL necessitates meticulous approaches to data curation, algorithmic design, and continuous monitoring to ensure that AI systems function in a manner that is both equitable and ethical [39].

Promoting transparency in algorithmic decision-making and enhancing the interpretability of ML models can aid in identifying, understanding and mitigating these biases [187]. In the next section, we focus on bias in LLMs which is the main focus of Paper V of this thesis.

6.1 Bias in Large Language Models

Biases in LLMs, and other large-scale ML, are indeed often a reflection of the datasets on which they are trained. These models, which are often trained on extensive collections of text from the internet, books, articles, and other written materials, can inadvertently learn and propagate the biases that are embedded within those texts. The biases can be multifaceted, encompassing gender [188], race [189], culture [190], and socio-economic status, among others [191, 192]. Fairness in LLMs, therefore, becomes a critical concern, as these biases can lead to the reinforcement of stereotypes and unfair treatment of certain groups when the models are used for tasks like text generation, conversation, or decision-making aids [49].

To address these concerns, it is essential to implement strategies aimed at mitigating bias in LLMs. This can include:

- Curating Diverse and Balanced Datasets: Ensuring that the training data includes a wide array of perspectives and is representative of different groups to reduce the risk of overfitting to biased samples. However, the implementation of this strategy faces significant challenges, particularly due to the vast amounts of data required for pre-training these models [22]. The sheer scale of data needed to effectively train LLMs means that any comprehensive dataset is likely to contain biases, as historical and existing data inevitably reflect the prejudices and inequalities present in society.

- Bias Detection and Evaluation: Considering the infeasibility of the previous step, effectively detecting and evaluating bias becomes the first major step towards bias mitigation. This involves employing metrics and evaluation frameworks specifically designed to detect and quantify biases in model outputs [49, 193].

- Debiasing Techniques: Algorithmic interventions play a crucial role in mitigating biases in LLMs. An example of this includes counterfactual data augmentation, which involves generating and incorporating synthetic data that represents counterfactual scenarios with the aim to balance the dataset thus enabling the model to learn to disentangle the protected attribute from the prediction task [194].

- Transparency and Interpretability: Transparency and interpretability are essential for understanding and addressing biases in . Leveraging XAI to understand how models generate specific outputs can help identify and correct biased decision-making pathways. Several methodologies have been developed recently for generating explanations for , such as chain-of-though reasoning [195], as well as post-hoc explanations such as Vanilla Gradients, Gradient x Input and contrastive explanations [196].

To take an initial step in this direction, as a part of this thesis, we investigate fairness exhibited by LLMs, tested rigorously on tabular data in Paper V. Tabular data, often structured with clear attribute-value pairs, is a common format in many real-world applications [197], such as finance, healthcare, and human resources. It is crucial for LLMs to handle such data without perpetuating or amplifying existing biases. To this end, we apply LLMs to tabular datasets and scrutinize their outputs for signs of bias with respect to protected attributes like gender and race.

Part IV

Summary of Research

7

Paper I

This looks More Like that: Enhancing Self-Explaining Models by Prototypical Relevance Propagation

Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, Michael Kampffmeyer

Pattern Recognition, 2023

In this paper, we identify and address the main shortcomings of the explanations generated by one of the state-of-the-art SEM, ProtoPNet [42]. Through an extensive case study, we examine the performance of ProtoPNet when confronted with various types of artifacts. Additionally, we propose a systematic approach for the effective detection and elimination of such artifacts from the training dataset.

We start by arguing that for prototypical SEMs, ideally artifacts in the training data should be captured by some of the class prototypes, removal of which from the model should result in an artifact-free model. However, we demonstrate that due to model-agnostic upsampling used by ProtoPNet, the local explanation

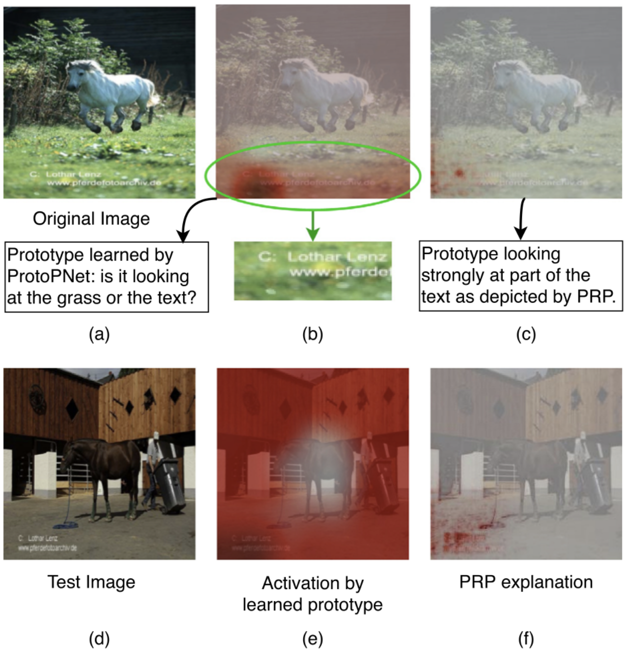

Original Image
Prototype learned by ProtoPNet: is it looking at the grass or the text?
C. Lothar Lenz www.pferdet
Prototype looking strongly at part of the text as depicted by PRP.
(a)
(b)
(c)
Test Image
Activation by learned prototype
PRP explanation
(d)
(e)
(f)

Figure 7.1: (a) Horse image from PASCAL VOC [166], (b) Prototype learned by ProtoPNet, (c) Prototypical explanation by PRP, (d) Test image from horse class, (e) Activation by prototype (b), (f) Fine and precise PRP explanation obtained. Example from Paper I.

maps generated are coarse and spatially imprecise. To address this, we propose a novel method called PRP, a backpropagation-based explanation method for prototypes, inspired by LRP [125], which attains more accurate model-aware explanations. Our aim is thus to maintain the advantage of the self-explanatory architecture through prototypes as well as simultaneously improving the quality of prototypical explanations by adding a model-aware explanation strategy. The improved explanations by PRP are shown in Figure 7.1.

We demonstrate the effectiveness of PRP in detecting Clever-Hans and Backdoor artifacts, which might go unnoticed otherwise. Additionally, in this work, we go one step further and suppress the potential artifact learned by the models. Since with the help of PRP we demonstrate that the artifact learning is entangled in the whole model, we propose automated cleaning of the data instead of pruning the prototypes. We filter out the artifact data using multi-view clustering applied on the multiple views generated from prototypical explanations. All experiments were conducted on subsets of LISA traffic sign dataset [198].

This work, in addition to contributing by advancing XAI, demonstrates the importance and efficiency of XAI for development of reliable AI by detecting as well as mitigating artifact learning, thereby increasing trust in ML systems.

Contributions by the author

- I developed the methodology in collaboration with my co-authors.
- I made all implementations and conducted all experiments.
- I wrote the original draft of the manuscript.

8

Paper II

Demonstrating the Risk of Imbalanced Datasets in Chest X-Ray Image-Based Diagnostics by Prototypical Relevance Propagation

Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, Michael Kampffmeyer

IEEE 19th International Symposium on Biomedical Imaging (ISBI), 2022

Building on Paper I, in this work, we address the problem of unintentional artifact learning in the safety-critical area of healthcare, specifically, Chest X-Ray analysis. In the era of data-driven DL models, the scarcity of labeled data is commonly addressed by amalgamating datasets from various sources [199]. However, this can prompt the models to inadvertently learn source-based spurious correlations to solve the task [164]. In this work, we show that models can rely on textual annotations within Chest X-Ray images, which includes source metadata, to make predictions. This can lead to falsification of performance statistics as the model appears to be working well when in reality failing to capture class-related pathology-based characteristics.

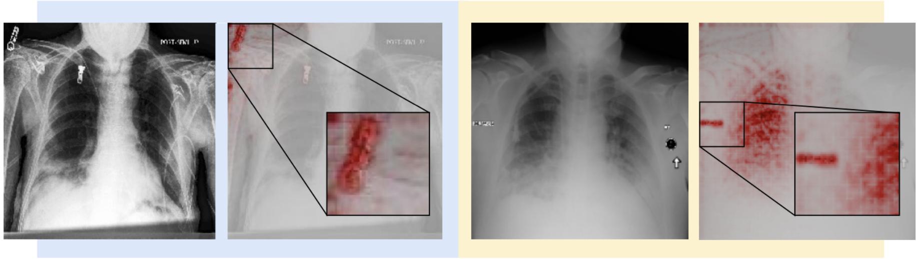


Figure 8.1: PRP maps of models with 90% (blue) and 60% (yellow) imbalance in source-related pneumonia labels. Example from Paper II.

The experiments are performed on the combination of two commonly used Chest X-Ray datasets, ChestX-Ray14 [200] and CheXpert [201] for the problem of pneumonia detection. We deliberately introduce a gradual imbalance in the prevalence of pneumonia images from one hospital system to assess the behavior of the model. Additionally, and more importantly, we argue that this unanticipated behavior can go unnoticed with black-box models, thus advocating the use of XAI. We, therefore, illustrate how these spurious correlations can be detected with the help of the proposed PRP from Paper I. Experimental results demonstrate that the model learns source related text-annotations, as shown in Figure 8.1. We therefore conclude that in the presence of source-related disease imbalance, the DL methods quickly start acting as an unintentional source-detector instead of the intentional disease-detector.

Contributions by the author

- I developed the methodology in collaboration with my co-authors.
- I made all implementations and conducted all experiments.
- I wrote the original draft of the manuscript.

9

Paper III

ProtoVAE: A Trustworthy Self-Explainable Prototypical Variational Model

Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina M-C Höhne, Michael Kampffmeyer

Advances in Neural Information Processing Systems 35 (NeurIPS), 2022

This paper addresses multiple shortcomings in existing prototypical SEMs. Current approaches often simulate prototype transparency by visualizing nearest training samples [127, 155], while some use actual training images as prototypes, which obstructs end-to-end optimization and constrains model flexibility [42]. Moreover, existing methods fail to ensure inter and intra class diversity of prototypes [42]. Additionally, there is often a trade-off between predictive accuracy and self-explainability. [42, 158].

To overcome these limitations, we introduce ProtoVAE, a new prototypical SEM built upon a VAE backbone. Specifically, the model learns a mixture of VAEs, each having its own Gaussian prior centered on one of the prototypes, while sharing the same encoder and the decoder, as shown in Figure 9.1. The prototypes generated by ProtoVAE thus serve as genuine transparent global ex-

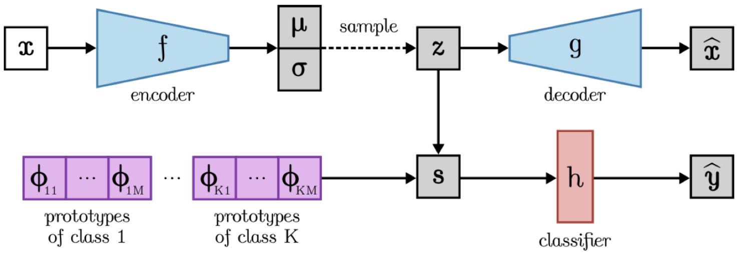


Figure 9.1: Schematic of ProtoVAE from Paper III.

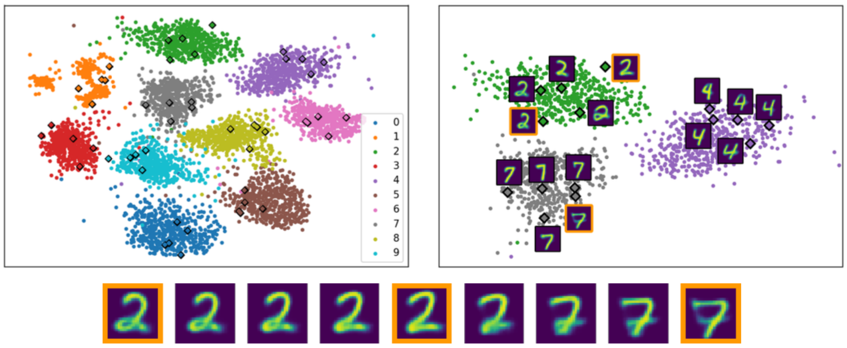


Figure 9.2: UMAP representation [202] of transparent prototypical space learned by ProtoVAE for 'MNIST' dataset [203]. Bottom row shows interpolation between prototypes of the same class (2) and between prototypes of different classes (2-7). Example from Paper IV.

planations, which can be decoded into the input space for visualization (Figure 9.2). Further, the local pixel-wise explanations are generated using PRP, the methodology proposed in Paper I.

Additionally, we define three essential predicates for an efficient and comprehensible formalization of SEMs: transparency, diversity, and trustworthiness. While transparency is inherent to the design of the model, diversity relates to the quality of prototypes learned, and trustworthiness translates to the quality of decisions, as well as explanations, achieved by the model. We demonstrate that the proposed ProtoVAE is able to produce transparent, diverse, and trustworthy predictions, as well as explanations, while relying on an end-to-end optimization. Empirically, ProtoVAE validates its trustworthiness by delivering robust performance as well as generating faithful explanations on several open-source datasets.

Contributions by the author

- I developed the methodology in collaboration with my co-authors.
- I made all implementations and conducted all experiments.
- I wrote the original draft of the manuscript.

10

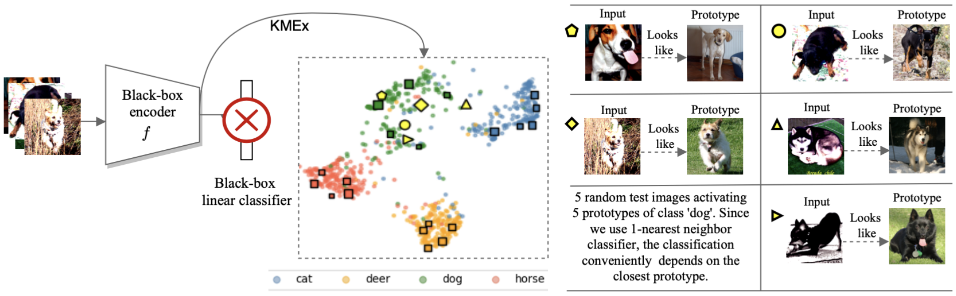
Paper IV

Prototypical Self-Explainable Models Without Re-training

Srishti Gautam, Ahcene Boubekki, Marina M-C Höhne, Michael Kampffmeyer

Under Review, 2023

Existing SEMs incorporate complex designs, based on large DL models as backbones [42, 155]. This necessitates intricate training strategies, further associated with large computational requirements, thereby limiting their accessibility. In this work, we propose a universal method, called KMEx, which is the first approach that aims to convert a trained black-box model into a prototypical SEM, without requiring retraining. The class-representative prototypes are learned from the latent representations extracted from a pre-trained model by utilizing K-means clustering, as shown in Figure 10.1. The final classification is then achieved by comparing the similarities of the input data with the prototypes and utilizing a 1-nearest neighbor classifier, thereby maintaining transparency in the decisions (10.1). The global explanations are achieved by visualizing the nearest training example to the prototype, while the local explanations utilize PRP, proposed in Paper I.



5 random test images activating 5 prototypes of class 'dog'. Since we use 1-nearest neighbor classifier, the classification conveniently depends on the closest prototype.

Figure 10.1: Illustration of proposed KMEx, along with transparent decision making shown on 'STL-10' dataset [204]. Example from Paper IV.

Additionally, we address the lack of comprehensive evaluation strategies for prototypical SEMs by proposing quantitative measures for the predicates defined in Paper III. This enables a comprehensive objective evaluation between existing methodologies, as opposed to largely qualitative evaluations used until now. Results on multiple open-source datasets demonstrates the efficacy of KMEx in matching the performance of the corresponding black-box models, while offering inherent interpretability without altering the embedding. This makes KMEx an efficient benchmark for prototypical SEMs.

Further, our proposed evaluation framework uncovers several advantages and disadvantages of existing prototypical SEMs. Specifically, we find that existing methods have the tendency to ghost the prototypes, i.e, never utilizing them for predictions, therefore rendering them useless. Further, the large variations in the design and regularizations of other SEMs lead to drastically different learned representation spaces, unlike KMEx. Finally, while diversity is ensured in several existing SEMs using regularization of losses, it is not reflected in the learned prototypes. We illustrate how KMEx can be leveraged, without the need for retraining, to improve the prototype positioning, thereby achieving better diversity, on already trained SEMs's embeddings.

Contributions by the author

- I developed the methodology in collaboration with my co-authors.
- I made implementations and conducted experiments in collaboration with Ahcene Boubekki.
- I wrote the first draft of the manuscript and refined it in collaboration with my co-authors.

11

Paper V

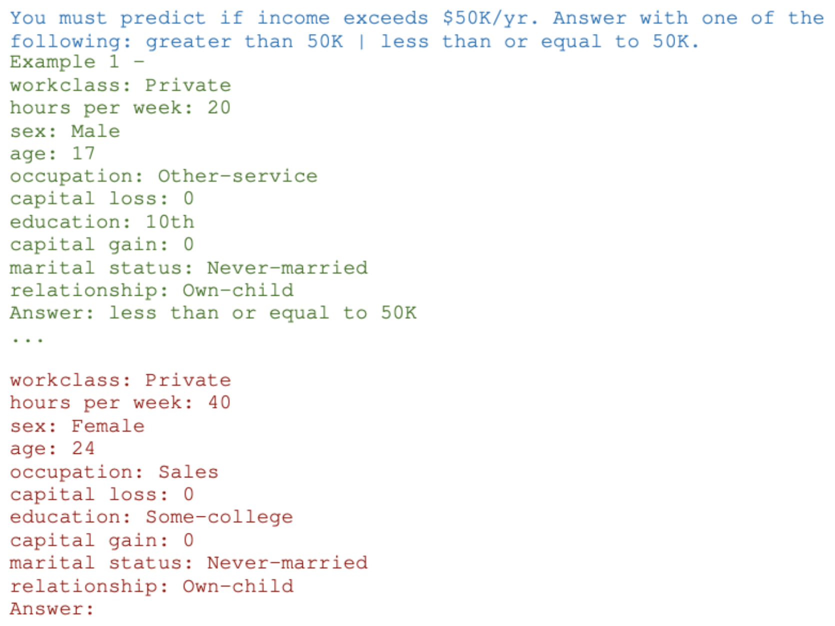
Investigating the Fairness of Large Language Models for Predictions on Tabular Data

Yanchen Liu, Srishti Gautam, Jiaqi Ma, Himabindu Lakkaraju

Under Review, 2023

Tabular data, often structured in tables as a result of relational databases [205], is widely used in critical decision-making processes [206]. Recent studies have suggested employing LLMs for tabular predictions by converting tables into natural language descriptions [197]. Contrary to traditional ML models, which lack context such as column names or descriptions, LLMs can utilize this contextual information, making them more perceptible to propagating demographic biases. However, while fairness in traditional ML methods for tabular data has been thoroughly investigated [207], the fairness implications of using LLMs for such tasks remains rather unexplored.

In this paper, we delve into this vital issue, aiming to uncover the information that LLMs depend on when making predictions from tabular data. Our goal is to determine the extent to which LLMs may be influenced by societal biases and stereotypes in their predictions.



```
You must predict if income exceeds $50K/yr. Answer with one of the
following: greater than 50K | less than or equal to 50K.
Example 1 -
workclass: Private
hours per week: 20
sex: Male
age: 17
occupation: Other-service
capital loss: 0
education: 10th
capital gain: 0
marital status: Never-married
relationship: Own-child
Answer: less than or equal to 50K
...

workclass: Private
hours per week: 40
sex: Female
age: 24
occupation: Sales
capital loss: 0
education: Some-college
capital gain: 0
marital status: Never-married
relationship: Own-child
Answer:
```

Figure 11.1: Exemplar of prompt template from Paper V for Adult dataset [208].

Our experiments with GPT-3.5 in a zero-shot setting, in comparison with traditional ML approaches, i.e neural networks and random forests [209], reveal that LLMs indeed manifest significant social biases. This suggests that these models inherit and leverage biases from their training data during prediction. We also find that both few-shot in-context learning (example prompt shown in Figure 11.1) as well as fine-tuning have a moderate impact on reducing bias. Furthermore, our research shows that label-flipping of few-shot examples can improve fairness metrics across different demographic groups, albeit with a trade-off in predictive accuracy, further highlighting the presence of inherent biases. These findings emphasize on the need for more sophisticated approaches, such as XAI techniques, to efficiently detect and mitigate biases and promote fairness in LLMs deployments.

Contributions by the author

- I developed the methodology in collaboration with my co-authors.
- Together with Yanchen Liu, I implemented and conducted the experiments.
- I contributed to the writing of the manuscript.

12

Concluding Remarks

This thesis contributes towards the enhancement of trustworthiness and dependability of AI models. The research centered around three key objectives: 1) Enhancing *transparency* and interpretability of AI systems, 2) Enhancing *reliability* of AI systems, and 3) Analysing un-/*fairness* of AI systems.

Focusing on enhancing *transparency* of AI systems, SEMs have been advanced in this thesis, first by improving the explanations provided by an existing state-of-the-art SEM method, ProtoPNet [42]. Leveraging the proposed method, PRP, in the context of ProtoPNet, more spatially accurate and fine-grained local explanations can be obtained. Additionally, novel SEMs were proposed, improving both the predictive performance as well as the explanation quality compared to existing methodologies. Specifically, a novel SEM was proposed, built on a VAE as the backbone, thereby learning a transparent prototypical space visualizable in the input space with the learnt decoder. The unique architecture enables end-to-end training resulting in no loss in predictive performance when compared to equivalent black-box models, unlike existing SEMs. Furthermore, a universal method, called KMEx, was proposed, which is able to convert any black-box model into a self-explainable one. KMEx achieves this inherent interpretability without requiring re-training of the black-box model, unlike existing methods, thereby enhancing the SEMs' accessibility. Additionally, Paper III contributed to the formalization of this relatively new domain of research by introducing a set of predicates for SEMs which facilitate a thorough comparison of existing models in terms of transparency, diversity, and trustworthiness. Paper IV further advanced the field by proposing a comprehensive quantitative evaluation framework that leverages the predicates established in Paper III, thus allowing for more effective comparative analysis.

In the intersection between research objective 1 and 2, for improving the *reliability* of AI systems, the phenomenon of artifact learning was addressed through using the proposed PRP method. This approach successfully identified Clever Hans and Backdoor artifacts within the models by generating precise prototypical explanation maps for the datasets. Subsequently, these maps facilitated the removal of instances containing such artifacts by employing multi-view clustering technique. The efficacy of the PRP method was further evaluated in a healthcare context, revealing that when DL models are trained on amalgamated source datasets, they tend to function as source detectors rather than as intended disease detectors. This discovery underscores the critical role of XAI in domains where safety and accuracy are paramount, highlighting their need to ensure that models perform their intended tasks without being misled by confounding artifacts.

Our research further explored the *fairness* of LLMs in the context of processing tabular data. To assess fairness across multiple dimensions, a range of distinct learning techniques was employed. Zero-shot learning was used to gauge the model's unbiased performance on unfamiliar tasks, in-context learning was used to assess how context influences model predictions, and fine-tuning was used to determine if further training could correct or worsen biases. Our studies of the GPT-3.5 [22] model revealed a tendency to perpetuate biases from their training data, raising concerns about potential unfair outcomes based on demographic attributes such as race and gender. These findings highlight the necessity for rigorous fairness evaluations and bias mitigation in the development and application of LLMs.

12.1 Limitations and Outlook

This section delves into the limitations of the studies incorporated within this thesis. Furthermore, it outlines the potential paths for future work that build upon the methodological foundation laid out in this thesis.

Paper I A drawback of Paper I is the necessity for manual examination of the clusters to differentiate between the 'artifact' and 'clean' data clusters. While this manual intervention did not present a substantial issue for the datasets evaluated in this study, which contained only one type of artifact, it may not be scalable for more complex datasets with multiple artifact types. To address this challenge in more intricate datasets, advanced methods for 'artifact' cluster identification could be employed, such as the approach suggested in [45]. Moreover, it is critical to assess how the clustering performs when confronted with multiple artifacts within the same category. The design of explainable approaches with the inherent capability to leverage artifactual data in addition to clean data without capturing the artifact features would be ideal instead of removing the data.

Paper II As Paper II builds upon the methodology proposed in Paper I, it also inherits its limitations. Moreover, given that the artifacts in this study – source-identifying annotations – are intrinsically embedded within the dataset, the multi-view clustering method for excluding data containing artifacts proves to be ineffective. Consequently, there is a call for a more refined strategy that focuses on the unlearning of artifacts directly within the model's design.

Paper III The current approach, proposed in III, assumes a fixed number of prototypes for each class. This assumption potentially limits the representational capacity, as it enforces a uniform number of prototypes across classes, regardless of their complexity. For instance, considering two classes from the ImageNet [93] dataset, a class consisting of 'motor scooter' images might embody more variability than a 'balloon' class, necessitating a more nuanced prototype representation. A potential remedy could be a distance-based prototype pruning method. Another prospective strategy can involve imposing a prior on the prototypical similarity distribution, thereby selectively prioritizing prototypes based on the frequency of their utilization in making predictions. Moreover, the quality of our global explanations is fundamentally dependent on the performance of the backbone VAE. However VAEs are commonly associated with the generation of blurry images [210, 211]. Therefore, adoption of other advanced methodologies, such as Very Deep VAEs [212] and normalizing flows [213], needs to be explored to enhance the method's applicability to more complex datasets.

Paper IV Similar to Paper III, the proposed KMEx relies on predefining the number of prototypes, a feature it has in common with other SEMs. It is also important to acknowledge that the detailed quantitative evaluation framework introduced is intended to complement, rather than replace, the qualitative assessments of SEMs. We argue that both assessments are essential owing to the subjective nature of explanations, which requires qualitative insights to fully capture the effectiveness of the SEMs.

Paper V The findings discussed in Paper V pertain exclusively to a single model GPT-3.5 [22], which may not reflect the broader landscape of LLMs. Future research could broaden this scope by incorporating a variety of models, including LLaMA [117] and BLOOM [214], to provide a more comprehensive assessment of fairness. The research is also limited to plain prompting methods; experimenting with additional prompting methods, such as, Chain of Thought prompting [195] could offer deeper insights into improving fairness. Furthermore, there is an opportunity to investigate sophisticated bias mitigation strategies that could contribute to the development of more equitable LLMs.

Future directions The advent of SEMs marks a significant stride towards achieving interpretability in AI systems. Nonetheless, the challenge of generalizing SEMs, such as to ensure their applicability across a wide range of domains, tasks and model, especially in the context of LLMs, presents a potential area with future research. By focusing on this aspect, we can make substantial progress in not only enhancing the interpretability but also in promoting the fairness of AI systems by identifying and mitigating biases. The explainability of LLMs has been explored through natural language explanations, such as Chain of Thought [195] and Tree of Thought methods [215], as well gradient based post-hoc methods [196]. Subsequent research has delved into the issues of unfaithfulness [216] and the perpetuation of social biases in these explanations [217]. However, SEMs still remain a rather unexplored category.

As we continue to refine SEMs, there is a compelling opportunity to design more sophisticated versions tailored for artifact mitigation. By integrating mechanisms for detection and mitigation, SEMs can ensure that explanations and decisions stem from pertinent and legitimate features. This would not only improve the trustworthiness of AI systems but also enhance their robustness against adversarial attacks or dataset biases that could otherwise compromise their performance.

Further, to maximize the potential of SEMs, the development of comprehensive evaluation tools is essential. While this thesis introduces detailed evaluation frameworks for Prototypical SEMs, the broader spectrum of SEMs still lacks robust evaluation mechanisms. Existing tools like Quantus [151] and OpenXAI [218] provide extensive metrics and visualizations for post-hoc explanations, yet similar frameworks for SEMs are notably lacking. Availability of such tools can help enable a more systematic comparison between models and facilitate the identification of best practices for generating self-explanations.

Part V

Included Papers

Paper I

This looks More Like that: Enhancing Self-Explaining Models by Prototypical Relevance Propagation

Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, Michael Kampffmeyer

Pattern Recognition, 2023

Pattern Recognition 136 (2023) 109172

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

This looks *More* Like *that*: Enhancing Self-Explaining Models by Prototypical Relevance Propagation

Srishti Gautam[a,*], Marina M.-C. Höhne[a,b,c], Stine Hansen[a], Robert Jenssen[a], Michael Kampffmeyer[a]

[a] *UiT The Arctic University of Norway, Tromsø, Norway*
[b] *Leibniz Institute for Agricultural Engineering and Bioeconomy*
[c] *University of Potsdam*

ARTICLE INFO

Article history:
Received 21 December 2021
Revised 19 June 2022
Accepted 9 November 2022
Available online 12 November 2022

2010 MSC:
00-01
99-00

Keywords:
Self-explaining models
Explainable AI
Deep learning
Spurious Correlation Detection

ABSTRACT

Current machine learning models have shown high efficiency in solving a wide variety of real-world problems. However, their black box character poses a major challenge for the comprehensibility and traceability of the underlying decision-making strategies. As a remedy, numerous post-hoc and self-explanation methods have been developed to interpret the models' behavior. Those methods, in addition, enable the identification of artifacts that, inherent in the training data, can be erroneously learned by the model as class-relevant features. In this work, we provide a detailed case study of a representative for the state-of-the-art self-explaining network, ProtoPNet, in the presence of a spectrum of artifacts. Accordingly, we identify the main drawbacks of ProtoPNet, especially its coarse and spatially imprecise explanations. We address these limitations by introducing Prototypical Relevance Propagation (PRP), a novel method for generating more precise model-aware explanations. Furthermore, in order to obtain a clean, artifact-free dataset, we propose to use multi-view clustering strategies for segregating the artifact images using the PRP explanations, thereby suppressing the potential artifact learning in the models.

1. Introduction

When applying AI models, especially in safety-critical areas, such as medical applications, autonomous driving, or criminal justice, we need to understand their underlying behavior to decide the model's trustworthiness. Here, the field of explainable AI (XAI) has established itself, where methods are being developed to illuminate the so-called black box models [1,2]. XAI serves as an essential support in ethical, legal, and social issues and ultimately also contributes to an increased acceptance by the end user [3] by revealing the input features that led to a certain model prediction.

Using those XAI methods, recent work has shown that models can learn artifacts that are present in the training data [4]. Such artifacts can be based on a so-called selection bias in the training data, where, for example, objects of a class have a certain background, and as a result the background is learned instead of the object. Furthermore, the training data can be manipulated by inserting a special trigger called "backdoor" which, if present in a sample, always leads to the prediction of a specific target class - i.e. a "backdoor" to this target class [5] In addition, a phenomenon called "Clever Hans", refers to an artifact that is correlated with a certain class in the training data and hence, used for classification such that the model could make a right prediction, but for the wrong - the artifact - reason [4]. In order to guarantee a faithful use of AI systems, it is important to find and suppress those artifacts either from the model, i.e., from the learnt representations or from the data itself, thereby enabling the retraining of the model with a clean dataset.

Recently, so-called post-hoc XAI methods, such as Layerwise Relevance Propagation (LRP) [6] were able to uncover this undesirable behavior of AI models [4]. Post-hoc refers to the fact that the XAI method explains the prediction of the model after (post) the prediction is made. However, [7] suggested to use an influential alternative to post-hoc explainability, called self-explaining neural networks, which can intrinsically explain their decision making process. Towards this goal, [8] recently proposed a network (ProtoPNet) that provides a transparent prediction by introducing a prototype layer between the final convolution layer and the output layer. This prototype layer consists of a fixed number of pro-

[*] Corresponding author.
E-mail addresses: srishti.gautam@uit.no (S. Gautam), mhoehne@atb-potsdam.de (M.M.-C. Höhne), s.hansen@uit.no (S. Hansen), robert.jenssen@uit.no (R. Jenssen), michael.c.kampffmeyer@uit.no (M. Kampffmeyer).

https://doi.org/10.1016/j.patcog.2022.109172

totypes for each class, which can be thought of as representative instances for each class of the training data. During the classification process, for each image that is passed through the network, prototype-specific activation maps are computed based on the similarity between the image and the prototypes. The visualization is performed by upsampling the activation maps to the input size, thus highlighting the most relevant pixels contributing to the classification. Doing this procedure for both, the prototype (training) images and the test image, the regions of interest can be visualized, serving as a direct comparison for the user to capture the relation between the test image and the prototype images from the training set. This accordingly helps in comprehending the decision of the network by “this relevant feature of the test image looks like that relevant feature from the class-specific prototype image” (*This* looks like *that*).

Recalling the artifacts issue, the solution now appears to be clear when using self-explaining neural networks, such as ProtoPNet: If the model learned a feature corresponding to the artifact, then it must be reflected by at least one of the prototypes of the class consisting of such artifacts. Consequently, once the artifact prototypes have been identified, their influence on the prediction can be stopped by pruning.

Interestingly, in this work we demonstrate that this idea of removing the artifact prototypes is not feasible owing to the coarse and spatially imprecise explanations provided by ProtoPNet, which is, due to its model-agnostic upsampling. Therefore, building on the principles of the post-hoc explanation method LRP, we propose a novel method referred to as Prototypical Relevance Propagation (PRP) to attain more accurate model-aware explanations (example shown in Fig. 1). We demonstrate that PRP efficiently captures the learned artifact, which might go unnoticed otherwise. Additionally, in this work, we go one step further and suppress the potential artifact learned by the models: using PRP, we illustrate that artifact information is entangled within the ProtoPNet, such that most prototypes capture artifact related features, making the above-mentioned pruning procedure not applicable. Therefore, we propose to clean the data instead of pruning the network. Knowing the ability of PRP of generating multiple views of the input in terms of learned prototypical explanations, we filter out the data points containing the artifact using multi-view clustering approaches. Our presented approach preserves the strength yielded by ProtoPNet of obtaining “*This* looks like *that*” explanations, while at the same time suppressing potentially learned artifacts. Moreover, we show that utilising multiple views through multi-view clustering is more efficient than a single-view LRP-based clustering approach, SpRAy [4].

Our main contributions are as follows:

- We identify and address key issues with inaccurate explanations provided by the self-explaining model, ProtoPNet.
- We propose a novel PRP method for enhancing ProtoPNet’s explanations by generating more precise model-aware explanations.
- We compare PRP with ProtoPNet’s explanation heatmaps, both qualitatively and quantitatively and show that eradicating learned artifact features, such as the Clever Hans and Backdoor artifacts, from ProtoPNet is unfeasible.
- We show the ability of PRP in utilizing multiple explanations from different prototypes, which can be utilized to suppress artifacts from the data by using multi-view clustering.

2. Related work

2.1. Explainability methods

Recently, there has been increased interest in both post-hoc explanation methods and self-explaining neural networks. Post-hoc explainability methods can be separated into two overarching categories: model-agnostic and model-aware approaches. Model-agnostic approaches [10], such as LIME [11] and SHAP [12], consider the models as black-boxes and are thus applicable to arbitrary model architectures and can be used to compare models based on the explanations that they produce. In contrast, model-aware approaches [13] take the internal structure of the model into account, yielding more precise model based explanations. Here LRP [6] has been widely used to explain the decisions of various deep neural networks, such as convolutional neural networks, recurrent neural networks and graph neural networks [14]. LRP assigns relevances to the input features by backpropagating the prediction score, i.e., the output relevance, successively layer by layer until it is distributed over the input features. Hence, the distribution of relevance is based on how much a particular node contributed to the output.

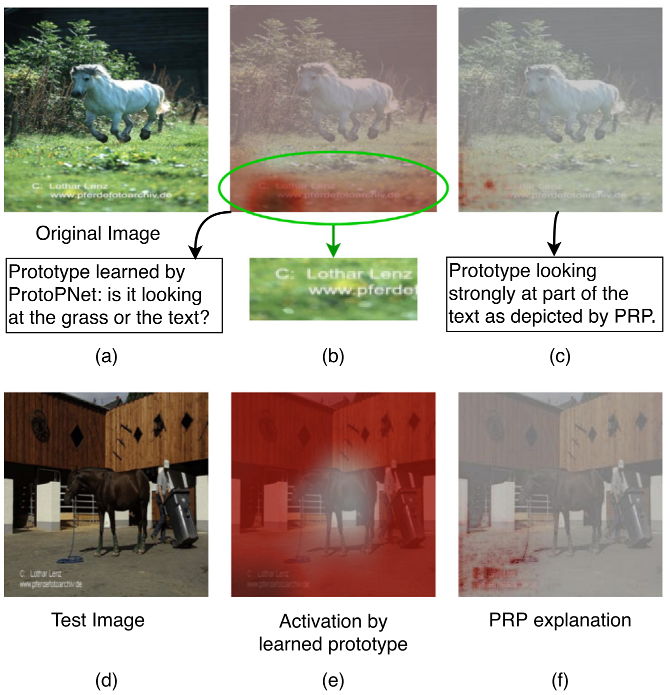

C. Lothar Lenz
www.pferdefotoarchiv.de
Original Image
Prototype learned by ProtoPNet: is it looking at the grass or the text?
C. Lothar Lenz
www.pferde
Prototype looking strongly at part of the text as depicted by PRP.
(a)
(b)
(c)
Test Image
Activation by learned prototype
PRP explanation
(d)
(e)
(f)

Fig. 1. (a) Visualization of a horse image from the PASCAL VOC 2007 dataset [9], (b) activation for a prototype of class *horse* learned by ProtoPNet, and (c) its PRP explanation. A Clever Hans artifact is present in the form of a watermark at the bottom of the image. Both, the ProtoPNet and the PRP explanation yield relevance to the bottom of the image, however, in the case of ProtoPNet, it remains unclear if the green grass, the text, or both together, were relevant for the prediction. Whereas the PRP explanation clearly shows that the text was used as relevant feature for the model’s prediction. For a test image (d), the ProtoPNet’s explanation and the PRP explanation for the learned prototype (b) are given in (e) and (f), respectively. The PRP explanation again corroborates the emphasis on the watermark text as opposed to ProtoPNet’s explanation which is more widely spread across the image. The ProtoPNet explanation in (e) thus exhibits ‘*This* looks like *that*’ behavior i.e explanation in (e) looks like prototype in (b). The PRP explanation in (f) exhibits ‘*This* looks *more* like *that*’ behavior i.e, enhanced explanation in (f) looks *more* like that in (c). . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Another new and promising category of explanation methods are self-explaining networks, which inherently explain the decisions they make, thereby making the models transparent by design. These include networks that align the latent space to known visual concepts in order to increase transparency in the decisions [8,15]. These also include models that utilize attention mechanisms [16] and thus also provide some form of self-explainability. Other works consider self-explainability in terms of concept learning [17,18]. Further, recently, some research has been originated to develop frameworks with a joint architecture consisting of an explainer and a classifier which learn in conjunction [19,20]. ProtoPNet [8] proposes to learn a specific number of class based prototypes as a part of the architecture. These are then used for visualizing lower spatial dimensional concepts from the training images, thus providing explanations during the decision process itself. SENN [21] is a type of general self-explaining model that is fully transparent and designed by progressively generalizing linear classifiers to complex models. Although the self-explainable concepts in SENN are using prototypes similar to ProtoPNet, the former only shows which training images are important for a decision. ProtoPNet, on the other hand, additionally shows what part of the test image looks like which part of the training images, thus providing more comprehensible information. The Classification-By-Components (CBC) network [22] is designed based on Biederman's theory in psychology, which assigns positive, negative, and indefinite reasoning to different components used for classification. Unlike CBC, ProtoPNet is more flexible in terms of *i*) learning components (prototypes) of varying sizes in the input domain, and *ii*) having the capability of being incorporated into any network architecture.

Inspired by ProtoPNet, XProtoNet [23] was recently introduced for automated diagnosis in chest radiography. It addresses the issue that ProtoPNet looks at fixed patch sizes in the feature map while computing its similarity with the prototypes. As a remedy, [23] adds an occurrence module in the network for learning features of dynamic size for the prototypes. However, the issues that we address in this work do remain in XProtoNet, making it prone to misleading explanations due to the model-agnostic upsampling used for prototype visualizations.

2.2. Artifacts

Real-world data used for training deep neural networks are prone to containing spurious, incomplete, or wrongly labeled samples thus leading to unwanted artifactual data. In this work, we acknowledge this inherent problem and focus on two common artifacts, Clever Hans and Backdoor, whose suppression is the focus of this work. Clever Hans artifacts refer to the unintentional spurious correlations present in the training data, which a model might use to base or strengthen their decisions on and is thus likely to fail in a real-world scenario, where the artifact is absent. This undesirable setting has also been explored recently by [4], in which they propose a semi-automated method, SpRAy, based on spectral cluster analysis on LRP maps, to discover prediction strategies based on an artifact. In other scenarios, the network might be forced to learn undesirable features based on the malicious addition of hidden associations in the data with the goal to produce incorrect inference results, referred as backdoor attacks. These kinds of attacks — where, in contrast to the Clever Hans scenario, both the data and labels are intentionally modified — are addressed in detail in [5,24].

3. An evaluation of ProtoPNet

While the effectiveness of post-hoc explainability methods has been investigated extensively [25,26] and their benefit has been questioned [7], there is a significant gap in the research for the analyses of the effectiveness of self-explainable approaches regarding quantitative analysis of the provided explanations [27]. Therefore, in this section, we provide a detailed analysis of ProtoPNet and its inherent explanations using a case study of Clever Hans artifact detection. As a representative for the self-explaining model, we focus on ProtoPNet as it claims to provide easily comprehensible case-based reasoning and is applicable to arbitrary CNN architectures by inserting a single prototype layer [8]. Additionally, it not only provides information about the features that the model's decision is based on, but also links this information to similar features in the training data, captured by the prototypes, thus imitating human decision making.

3.1. ProtoPNet

ProtoPNet introduces self-explanation in a deep learning network by incorporating a prototype layer between the last convolutional layer and the output layer. Thereby, each class is associated with a fixed number of prototypes. The output of the prototype layer is connected linearly to the output layer to generate class logits. The network is optimized by iterating the following three steps: 1) The whole network, except the last layer, is trained using stochastic gradient descent. For each prototype, the squared L_2 similarity between the patches of the convolutional output from the backbone and the prototype is calculated, thus generating an activation map. Global max pooling is applied to the activation map to generate a single similarity score corresponding to a single prototype. The loss function is a combination of the cross entropy loss, a cluster loss and a separation loss. The cluster loss encourages the training images to have a patch close to at least one of their own class prototypes. The separation loss, on the other hand, encourages the training image patches to be far from the prototypes of other classes [8]. For completeness, the losses are provided in the Appendix. 2) All prototypes are projected onto the patch of the training image from the same class as the prototype with the highest similarity score, thus maintaining inherent interpretability. These can be visualised in the input space by upsampling the activation map of the prototype image to the input size. 3) Finally, a convex optimization of the last layer is performed to further improve accuracy, while keeping the learned prototypes fixed. The prototype activations are visualized by upsampling the similarity between the prototypes and the embedded input image to the input image size. This highlights the parts of the image which strongly activate the respective prototype, thus creating a concept of "*this* looks like *that*" while making the decisions.

3.2. Evaluation of ProtoPNet's explanations

Although self-explaining models as ProtoPNet appear promising, as more transparent alternatives to the typical black-box neural networks, we demonstrate that, atleast for ProtoPNet, the explanation capability still lacks precision. In the case of ProtoPNet, the relevant areas on which the model decision is based on do not concisely depict the relevant features of a prototype as shown in Fig. 1. The original image (a) in Fig. 1 shows a horse image containing a watermark in the lower left corner. One of the 10 prototypes for class Horse was learned by ProtoPNet from image 1(a). The ProtoPNet's explanation for this prototype is shown in Fig. 1(b). From 1(b), we can observe that the lower left corner was important for the model to predict the image as a horse. However, the exact pixels, that significantly contributed to the predictions remain unknown. Now, using the model-aware PRP method, we backpropagate the prototype information from the prototype layer through the network to the input image, which allows us to reveal and visualize the model-aware, faithfully distributed relevance scores on the input image as shown in Fig. 1(c). From the PRP explanation, we observe that high relevance (dark red pixels) was allocated to parts of the text. Thus, the PRP explanation leads to an increased understanding of the underlying behavior of the model. For a randomly chosen test image, shown in Fig. 1(d), the activation for the learned prototype 1(b) as visualized by ProtoPNet and PRP are given in Fig. 1(e) and (f), respectively. The PRP explanation identifies the watermark (Clever Hans) as a relevant feature for predicting the horse class, in contrast to the ProtoPNet explanation, which

Table 1
Comparison of the model accuracies for the stop sign class between the artifact test (artifacts in 100% test images) and clean test (artifacts in 0% test images) dataset for : 1) CH-100, 2) CH-50 datasets, along with the accuracies for pruning artifact prototypes as well as retraining the last layer after pruning.

Test set	CH100	CH-100 Remove prototype 6 & 8	CH-100 Retraining last layer	CH50	CH-50 Remove prototype 4 & 9	CH-50 Retraining last layer
Artifact	100%	21.6%	88.8%	100%	100%	100%
Clean	6.5%	38.2%	38.2%	94.6%	93.0%	94.5%

is too crude to identify important features and is therefore widely spread across the entire image.

Accordingly, we detect and address the following drawbacks of ProtoPNet:

- The activation maps used for the prototype visualizations in ProtoPNet have a low resolution due to downsampling and feature aggregation functions in the network. From this significantly low resolution activation map, ProtoPNet performs model-agnostic upsampling using bilinear interpolation to the size of the input image, thus leading to very **coarse explanations**.
- The effective receptive field of a position in the activation map tends to cover large parts of the image, which is not captured by the naive upsampling. Consequently, there is no truthful spatial localization of the relevance to the correct input area, leading to **spatially imprecise explanations**.

In the next subsection, we discuss in detail these drawbacks of ProtoPNet's explanations using the Clever Hans artifact as an example.

3.3. Case study: Clever Hans artifact detection with ProtoPNet

Ideally, ProtoPNet should capture any artifact in the data as an "artifact prototype" if it is using the artifact for prediction. However, due to its coarse and spatially imprecise explanations, the heatmaps of ProtoPNet hinder the detection of artifact prototypes. In the following, we investigate the behavior of ProtoPNet in the presence of Clever Hans artifacts in the data.

We aim to detect the aforementioned artifact prototypes using ProtoPNet's explanations combined with the difference in classification results in the presence and absence of artifacts in the test data. Following this, we prune the detected artifact prototypes, thus hypothetically suppressing the artifacts learnt by the model. However, due to its misleading explanations, we demonstrate experimentally that ProtoPNet's heatmaps are deficient in capturing and identifying the learned artifact by the model, thus proving the task of pruning artifact prototypes futile for making the model artifact-free.

For considering a controlled environment, we use the 5-class version of the LISA traffic sign dataset [28] and place a Clever Hans artifact, a yellow square (see Fig. 2), in 100% of the training data of the stop sign class (dataset details are provided in Section 5.1), which we refer to as CH-100. We train the ProtoPNet (for implementation details see 5.2), with 10 prototypes per class as in [8] for ease of comparison.

To evaluate the impact of an artifact on the model, we evaluate the performance on two test data sets: an **Artifact Test** data set, where the Clever Hans, i.e., the yellow square, is inserted into 100% of the images of the stop sign class ; and a **Clean Test** data set, which contains no yellow square. The accuracy results for both test data sets are shown in Table 1. We can observe that the model, trained on the CH-100 dataset, has 100% classification accuracy on the artifact test data and only 6.5% on the clean test data. This large drop in the accuracy indicates that the model has learned the inserted artifact.

In order to detect the prototypes that are responsible for this behavior, we visualize the 10 prototypes learned by the network for the stop sign class in Fig. 2, where the upsampled activation heatmap is overlayed, such that the relevant areas of each prototype can be identified visually. Although no prototype is clearly focusing on the artifact, it appears that prototypes 6 and 8 might be learning a part of the artifact. By removing individual prototypes as well as combinations of prototypes for the stop sign class, we can confirm that prototypes 6 and 8 are the most responsible ones for detecting the artifact (Fig. 3) – the accuracy for artifact test data only drops when prototypes 6 or 8 are removed, with the biggest drop of 78.39% when both of these are removed together. Also note that no retraining is done yet after pruning the prototypes.

Now, trusting the explanations provided, we remove the artifact prototypes 6 and 8 and assume that this leads to the elimination of the artifact effect. As can be seen in Table 1, the accuracy for the artifact stop sign class drops considerably after removing prototypes 6 and 8. However, this is not the case as seen after retraining the last layer i.e, reweighing the connection of the prototypes to the final classification layer. The accuracy for the artifact stop sign class increases again to 88.8% once the last layer weights are retrained. Moreover, for clean test data, the accuracy remains the same, i.e, 38.2% before and after retraining the last layer, thus refuting the potential learning of meaningful features for the stop sign class by the model after retraining. Hence, the results indicate that the remaining prototypes include artifact information as well, highlighting the lack of accurate explanations by ProtoPNet.

Thus, as shown in the above experiment, the explanations provided by the upsampling strategy of ProtoPNet are insufficient in order to reveal the model's behavior and detect the artifacts faithfully.

4. Prototypical Relevance Propagation and enhanced suppression of artifacts

In the following we will address the two main drawbacks of ProtoPNet's visualizations, i.e., low resolution activation maps and spatially imprecise prototype explanations (as investigated in the section above), by our proposed method called Prototypical Relevance Propagation (PRP). Our aim is to maintain the advantage of self-explanatory architecture through prototypes and simultaneously improve the quality of prototypical explanations by adding, inspired by LRP, a model-aware explanation strategy.

4.1. Prototypical Relevance Propagation (PRP)

The original prototype visualization step in ProtoPNet is achieved through upsampling and is therefore decoupled from the other steps in its end-to-end training. Instead of upsampling, inspired by LRP, we suggest as a novel solution to use the knowledge of the inner workings of the network when backpropagating the similarity values of a prototype to the input, such that we obtain model-aware prototypical explanations. We refer to our method as PRP and the generated explanation maps as PRP maps.

For the following considerations, let the input images be represented as $\mathbf{x}$ and convolutional output from the backbone CNN as $\mathbf{z} \in \mathcal{R}^{H \times W \times D}$. Let $\mathbf{P} = \{\mathbf{p}_m\}_{m=1}^{n}$ be the n prototypes learned by the network, each with a shape of $H_1 \times W_1 \times D$. Following [8], we set $H_1 = W_1 = 1$ and $D = 128$. Moreover, let $\mathbf{S} = \{\mathbf{s}_m\}_{m=1}^{n}$ be the similarity scores and $\mathbf{A} = \{\mathbf{a}_m\}_{m=1}^{n}$ the activation maps for each proto-

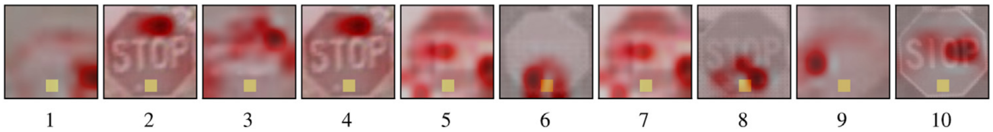


Fig. 2. CH-100: Visualization of the prototypes learned for the stop sign class for the scenario where Clever Hans artifacts were inserted in 100% of the stop sign class images for the modified LISA dataset. As observed, while prototype 6 and 8 can be considered as artifact prototypes, none of the prototypes clearly highlight the artifact.

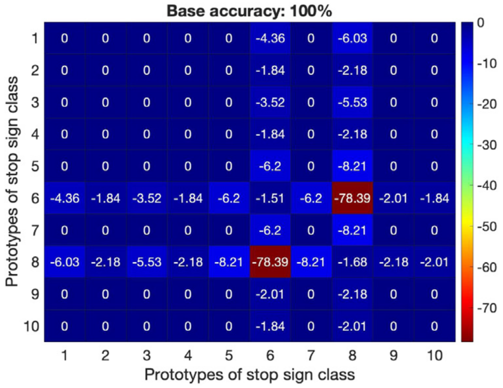


Fig. 3. CH-100: Detection of artifact prototypes by removing individual stop-sign class prototypes (1 to 10) (diagonal) and their combinations (non-diagonal) for artifact test data. The accuracies are represented as a drop from the base accuracy of 100% when no prototypes are removed. The highest drop of 78.39% is observed when prototypes 6 and 8 are removed together thus highlighting them as artifact prototypes.

type. The forward computations in ProtoPNet, illustrated in Fig. 4, are defined as follows:

1. The computation from the input to the convolutional output is given by $\mathbf{z} = f(\mathbf{x})$, where the function f represents the trained backbone CNN.
2. The activation maps are computed as squared L_2 similarities between the last convolutional output layer and the prototypes in the prototype layer:

$$\mathbf{a}_m = \log\left((||\tilde{\mathbf{z}} - \mathbf{p}_m||_2^2 + 1)/(||\tilde{\mathbf{z}} - \mathbf{p}_m||_2^2 + \epsilon)\right) \tag{1}$$

   where $\tilde{\mathbf{z}}$ are patches of $\mathbf{z}$ of the same size as the prototypes $\mathbf{p}_m$ and $\epsilon = 10^{-4}$ is a small constant introduced for numerical stability.
3. The similarity score based on the activation maps is calculated as $\mathbf{s}_m = \max(\mathbf{a}_m)$

The similarity scores of the test image with prototypes are the inputs to the final fully connected layer, which produces the logits for all output classes. Hence, the final classification is based on a linear combination of the similarity scores of different prototypes.

Now, to improve the precision of the prototype visualizations, we calculate a certain prototype m by propagating the relevance of this prototype back to the input features. Note that the relevance of a specific prototype is exactly its similarity score. Therefore, the first backpropagation step considers the redistribution of the similarity scores towards the activation map with respect to the max pooling layer:

1. An activation map is computed by backpropagating the respective similarity score with the LRP rule in the Max pooling layer:

$$\mathbf{R}_{mij}^{(AM,S)} = \begin{cases} \mathbf{R}_m^{(S)} & \text{if } \mathrm{argmax}_{ij}(\mathbf{a}_m), \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

   where S refers to the similarity score layer, AM to the activation map layer and i, j specify the spatial location in the respective layers. We define the relevance at layer S as $\mathbf{R}_m^{(S)} = \mathbf{s}_m$.
2. To distribute the relevance from the activation map back to the convolutional output, we need to incorporate the information from the forward pass. The forward computation as given in Eq. (1) computes the similarity between each prototype and each output patch of the convolutional layer ($CONV$), with both having D channels, thus compressing the channel dimension to 1 in the activation map. In this step, we redistribute the relevance from the one channel activation map back to the D channels of the convolutional output, weighted by the corresponding channel-wise L_2 similarities computed during the forward pass. We define the channel-wise similarities between each CNN patch $\tilde{\mathbf{z}}$ and the prototype $\mathbf{p}_m$ as:

$$\gamma_{mc} = \frac{1}{d_{mijc} + \epsilon} \tag{3}$$

   where, with $d_{mijc} = ||\tilde{\mathbf{z}}_\mathbf{c} - \mathbf{p}_{mc}||_2^2$ for each channel c. Afterwards, we use the LRP$_\epsilon$ [6] rule to distribute relevances to convolutional output according to γ_{mc}:

$$\mathbf{R}_{mijc}^{(CONV,AM)} = \frac{\gamma_{mc}}{\sum\limits_{k=1}^{D} \gamma_{mk} + \epsilon} \mathbf{R}_{mij}^{(AM)} \tag{4}$$

3. Finally, the PRP maps are computed by distributing the relevance from the convolutional output to the input features with the LRP CoMPosite ($\mathbf{LRP}_{CMP}$) rule [29]: First, the LRP$_{\alpha\beta}$ rule is applied to the convolutional layers

$$\mathbf{R}_{i\leftarrow j}^{(l,l+1)} = \left(\alpha \frac{z_{ij}^+}{z_j^+} + \beta \frac{z_{ij}^-}{z_j^-}\right) \mathbf{R}_j^{(l+1)}, \tag{5}$$

   where $z_{ij} = x_i w_{ij}$ is the mapping of the input x from neuron $i \rightarrow j$ with weight w_{ij}, $z_j = \sum_i z_{ij}$, $\alpha + \beta = 1$ and $\alpha \geq 1$. Note that positive and negative activations are treated separately and we use $\alpha = 1$ and $\beta = 0$.[1]
   Second, the Deep Taylor Decomposition based rule DTD$_{z^{\mathcal{B}}}$ [30] is applied to the input features

$$\mathbf{R}_{i\leftarrow j}^{(l,l+1)} = \left(\frac{z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}\right) \mathbf{R}_j^{(l+1)}, \tag{6}$$

   where l_i and h_i are the smallest and largest pixel values.

The algorithm for generating PRP maps is summarized in Algorithm 1.

4.2. Multi-view clustering

In order to analyse the class-wise prediction strategies and reveal potential strategies that are based on artifacts, [4] introduced SpRAy, a method that utilizes spectral cluster analysis to cluster LRP explanations into their key prediction strategies. Similar to SpRAy, we want to make use of the PRP maps to identify class specific global discriminative features. However, we do have multiple explanations for each image, i.e., the prototype explanations, which can be thought of as multiple views of an image explanation. Thus,

[1] Note, for notation simplicity, we follow previous works [6,29] and consider the convolutional layers as fully-connected layers with shared weights.

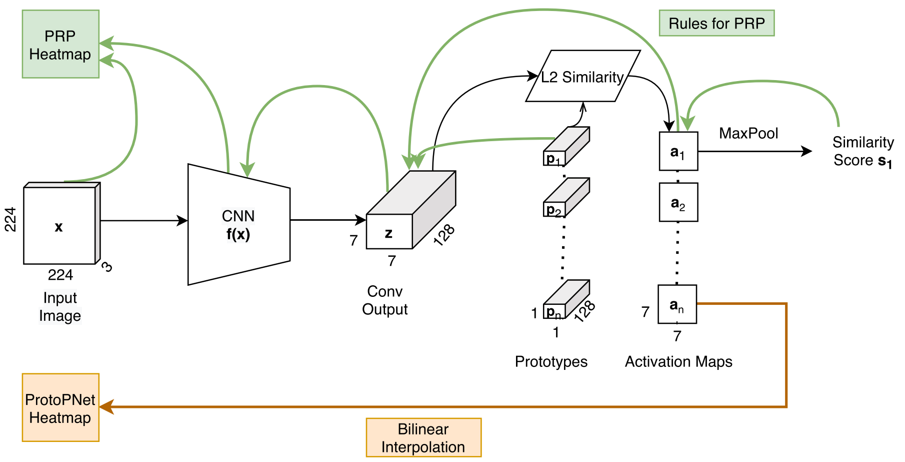


Fig. 4. ProtoPNet: Forward propagation and backward propagation for PRP maps (green) and ProtoPNet Heatmaps (orange). The input image $\mathbf{x}$ is first passed through a CNN f, which computes $f(\mathbf{x})$ to give output $\mathbf{z}$. The squared L_2 similarity is then computed between $\mathbf{z}$ and individual prototypes $\mathbf{p}_m$ to get activation maps $\mathbf{a}_m$. These are then upsampled to get ProtoPNet heatmaps. On the other hand, similarity scores $\mathbf{s}_m$ are used to compute model-aware PRP heatmaps. All the parameters in the figure are depicted according to the experiment settings used in this work. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

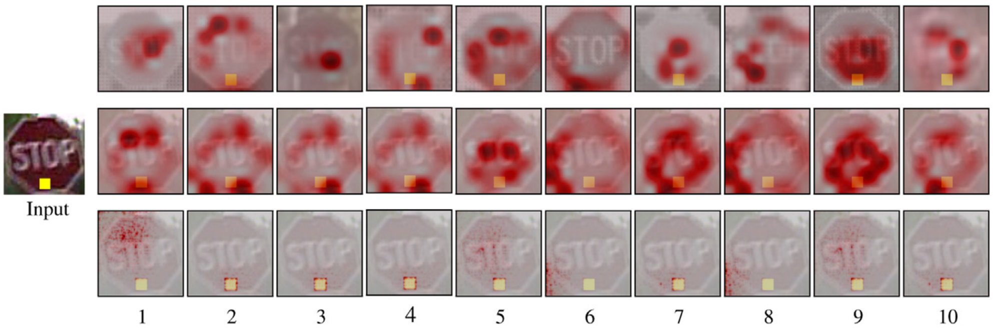


Fig. 5. CH-50: Top row depicts the learned prototypes 1 to 10 for the stop sign class with Clever Hans in 50% of the training data, the middle row depicts the ProtoPNet's heatmaps corresponding to the respective prototypes for the test image shown on the left, while the bottom row shows the corresponding PRP maps, which, we can observe, capture more precise information.

```
Algorithm 1: Psuedocode for the proposed PRP.
Input: Model $f$, image $\mathbf{x}$, prototype number $m$
$\mathbf{z} = f(\mathbf{x})$;                         /* Forward computation */
Compute $\mathbf{a}_m$          ; // Eq. 1
$\mathbf{R}_m^{(S)} = \mathbf{s}_m = \max(\mathbf{a}_m)$;
Compute $\mathbf{R}_{mij}^{(AM,S)}$     // Eq.2;   /* Backward computation */
Compute $\mathbf{R}_{mijc}^{(CONV,AM)}$ ; // Eq. 4
for $l \in CONV - 1, \ldots, 1$ do
    $\mathbf{R}_{i \leftarrow j}^{(l,l+1)}$ using $\mathbf{LRP}_{CMP}$ rules;
end
Output: $\mathbf{R}^{(1)}$
```

unlike SpRAy, which uses one LRP explanation for one image, our proposed method exploits multiple views of an image explanation.

In ProtoPNet, each class is associated with a fixed number of class prototypes. These can be regarded as capturing, and thus searching for, different features in each input image. Consequently, if there are artifacts present in a class during training, the PRP explanation maps for this class' prototypes will be able to reflect the contrast between artifact and non-artifact features learnt by the model. Therefore, interpreting the different prototype activations as various views of the same image, allows us to compare/cluster the prototype activations with multi-view clustering algorithms in order to detect global class-discriminative features in the data. Traditional multi-view clustering methods include learning a common representation from multiple views of data followed by clustering [31] or learning adaptive representations based on clustering [32]. Further, several multi-view clustering algorithms have been proposed that build on spectral clustering and consider a consensus Laplacian matrix among all the views [33,34]. In contrast, deep-learning based multi-view clustering methodologies learn a common encoding with the help of deep neural networks, which then can be leveraged by the clustering module [35]. Since a variation in clustering results can be observed using different multi-view clustering methodologies, in this work, we demonstrate the performance with a recent deep learning based clustering method [35] and a representative spectral multi-view clustering algorithm [33].

The deep multi-view clustering in [35] first transforms each input into its representation using view-specific encoders. The fused representation for all views is then computed using the fusion weights, which are also learned during the end-to-end training. This representation is then passed through a fully connected network to obtain the final cluster assignments. Deep divergence based clustering (DDC) [36] losses are incorporated to optimize the model. This approach is termed as Simple Multi-View Clustering (SiMVC). Trosten et al. [35] then introduces an auxiliary method which incorporates selective contrastive alignment of representations called Contrastive Multi-View Clustering (CoMVC) by adding a contrastive loss to the SiMVC framework. We provide the results with CoMVC in this work considering its additional advantage of aligning the representations at the sample level.

The spectral multi-view clustering methods work on the general principle of computing a consensus Laplacian matrix among all views. Co-regularized Multi-view Spectral Clustering (Co-Reg [33]) works by co-regularizing the clustering hypotheses. They obtain the combined Laplacian matrix by regularizing eigenvectors of the Laplacians through two schemes: 1) pairwise co-regularization, where they encourage the pairwise similarities across all views to be high and 2) centroid-based co-regularization, where they encourage each view to be closer to a common centroid.

5. Experiments & results

In this section, we first discuss the dataset and implementation details followed by detailed analysis of ProtoPNet and PRP heatmaps. Finally, we discuss in detail artifact suppression using multi-view clustering.

5.1. Dataset

In this work, we conduct experiments for both the Clever Hans and the Backdoor artifact using the LISA traffic sign dataset [28]. This dataset consists of video frames captured from a driving car. We follow the strategy of [5], where we extract the frames and resize them to 224x224 to be compatible with the original ProtoPNet architecture. The 47 classes in the dataset are partitioned into 5 high-level classes, as proposed by Chen et al. [5], consisting of restriction, speed limits, stop, warning, and yield signs (details provided in the appendix). In addition, we use the PASCAL VOC 2007 dataset [9] for evaluation as it naturally contains a Clever Hans artifact.[2]

5.1.1. Clever Hans

As artifact, we place a yellow post-it note, as shown in the input image in Fig. 5, in 100%, 50% and 20% of the stop sign images in the training data of the LISA traffic sign dataset to create the CH-100, CH-50 and CH-20 Clever Hans training datasets, respectively. We do not add Clever Hans artifacts to the PASCAL VOC 2007 dataset since it inherently includes a watermark tag of the photographer in about 15–20% of the images in the horse class [4].

5.1.2. Backdoor

According to the data manipulation scheme for backdoor attacks from [5] we insert the artifact, i.e., the yellow post-it, as shown in Fig. 5 (Input), in 15% of the stop sign images and assign them to the speed limit class. We refer to this corrupted training dataset as BD-15.

In order to create both, an artifact and a non-artifact i.e., a clean test dataset of the LISA traffic sign dataset, we insert the artifact in either 100% or 0% of the stop sign images, referred as Artifact Test and Clean Test data, respectively. Those test datasets are used for evaluating our experiments on the Clever Hans (CH-100, CH-50 and CH-20) as well as the Backdoor (BD-15) scenarios.

5.2. Implementation

We train ProtoPNet with ResNet34 as backbone architecture, fixing the number of prototypes to 10 for each class. Note that all training parameters have been set according to Chen et al. [8]. The network is trained for 1000 epochs, where a projection (push) of the prototypes is done every 10 epochs. After each push, the last layer is trained for 20 epochs. The learning rate is reduced by a factor of 0.1 every 5 epochs and the training is stopped when the training accuracy converges and the cluster loss becomes smaller than the separation loss on the training set [8]. While ProtoPNet uses bilinear interpolation for visualization, which takes 0.001 s on average, computed for 1000 images, PRP has an additional overhead of 0.71 s for one backward pass to generate the heatmaps. Note, given that heatmaps are produced only after training the model, this overhead can be considered negligible. The code is implemented using PyTorch and the experiments were run on 2 GeForce RTX 2080 Ti GPUs.[3]

5.3. PRP maps vs ProtoPNet heatmaps

In the following, we conduct an experiment, where we add a Clever Hans feature to the training dataset to investigate the difference between the heatmaps of ProtoPNet and the ones that PRP generates. Therefore, we add the Clever Hans artifact to 50% of the stop sign images in the training data (CH-50). The 10 prototypes for the stop sign class, learned by the ProtoPNet trained on the manipulated dataset, are shown in the first row of Fig. 5. Given a test image, shown at the very left of Fig. 5, the heatmaps of ProtoPNet and the PRP heatmaps for the image are shown in the middle and bottom row of Fig. 5. Corroborating our earlier observations, we again note here that the ProtoPNet heatmaps are coarse, highlighting wider areas in the test image, and that neighboring regions of the artifact are focused upon, rather than the precise location of the artifact. In contrast, from the PRP maps, we can clearly observe that all prototypes are focusing precisely on the Clever Hans feature, some more (prototypes 2, 3, 4, 5, 7, 9, 10) and some less (prototypes 1, 6, 8). It is shown later that prototypes 6 and 8 are in fact not learning any significant features and even react strongly to random noise. With the new insight into the model behavior gained through the PRP maps, we can shed new light on the hypothesis from Section 3.3. The idea was to remove the prototypes that had learned the Clever Hans, retrain the last layer and thus eliminate the Clever Hans effect. Given the original prototype explanation, this made sense, as only 2 of the 10 prototypes had learned the Clever Hans feature. With the PRP maps, however, we gain new knowledge and can see that all prototypes (some more, some less) take into account the Clever Hans feature.

We also note here that ProtoPNet heatmaps are highlighting all pixels in the image activated by different prototypes (before Max Pooling). If they were highlighting only the maximally activated region (after Max Pooling), they would only be able to depict connected regions in the image space, considering the naive upsampling heavily based on spatial location correspondence between the activation map and the input image. On the other hand, PRP maps represent the maximally activated pixels and are still able to highlight disjointed areas in the image, as can be seen in the PRP map for Prototype 5 in Fig. 5, where both the artifact and “ST” in the stop sign are indicated as relevant.

Fig. 6 illustrates the difference between PRP maps and ProtoPNet heatmaps for a stop sign image with no artifact. PRP maps, as shown in the bottom row, are of higher resolution and, as noticed in this case, tend to show more accurate information than the normal upsampled heatmaps from ProtoPNet. PRP maps also contain higher variability, as shown by explanations for Prototype 2 and 4 in Fig. 6, which therefore yields more information from the original prototypes to explain the test pattern.

In the following, we quantitatively evaluate the faithfulness of the PRP maps and ProtoPNet heatmaps regarding their ability to capture the most discriminative class-wise information. For this, we follow the strategy presented in [37], referred to as the Rel-

[2] Since in PASCAL VOC 2007, one image can belong to several classes, we deliberately remove the person class from this dataset to decrease ambiguity. The person images overlap to a large extent with the images of the other classes, leading to a lot of duplicate images in multiple classes.

[3] The source code is available at https://github.com/SrishtiGautam/PRP.

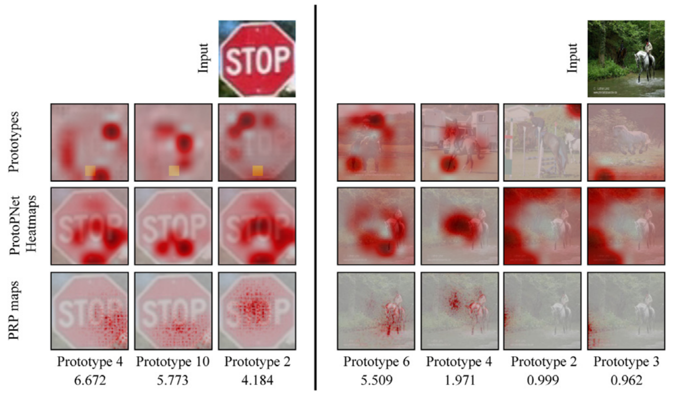


Fig. 6. PRP Maps vs Activation Map Upsampling for CH-50 (left) and PASCAL VOC 2007 (right). The top 3 activated prototypes for the stop sign class and the top 4 activated prototypes for the horse class for the respective input images are shown in the second row in descending order of similarity scores (last row). The third row shows the heatmaps generated by ProtoPNet and the last row shows the corresponding PRP maps.

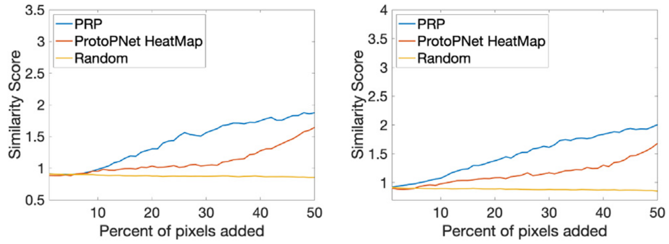


Fig. 7. CH-50: Quantitative evaluation of PRP Maps vs ProtoPNet Heatmaps via relevance ordering test. The results are shown as an average over all the prototypes and averaged over the same images without (left) and with artifact (right).

evance ordering test, where we start from a random image and monitor both the similarity scores as we gradually add the most relevant pixels to the image.

Primarily, we are interested in the trustworthiness of the ProtoPNet heatmaps and PRP maps with regard to their calculated pixel relevance for activating the prototypes. Therefore, first, for an input image, the PRP maps and the ProtoPNet heatmaps are computed, followed by sorting the pixels in descending order of their assigned relevance by PRP and ProtoPNet explanations, respectively. We then compute the similarity scores for different prototypes of the stop sign images while gradually adding the pixel with the next highest relevance to a random image. We compute this for 50 randomly chosen clean images from the stop sign class and compute the average across all images followed by an average over all prototypes. The same experiment is repeated with the same images, this time adding the Clever Hans artifact. The average results for all prototypes of the stop sign class are shown in Fig. 7. The x-axis represents the percentage of pixels that are replaced by the relevant pixels of the test image and the y-axis represents the corresponding similarity scores. As a baseline, we start from a random image and gradually replace a percentage of randomly chosen pixels by their test image pixel values and refer to this as the Random approach. From Fig. 7 we can observe that for both test case scenarios, i.e, the stop sign images with and without the artifact, adding the most relevant pixels, based on the PRP explanations, results in a significantly steeper slope (blue) than using the ProtoPNet heatmaps (orange). Therefore, conclusively, we can state that the relevance of the important discriminate features distributed by PRP is more accurate than by ProtoPNet explanations. These quantitative results also uncover ineffective prototypes which are not learning anything specific from the training images and are reacting very highly even to random noise, as shown in Fig. 8. This behavior is observed in both test scenarios of clean and artifact data, with the results depicted for artifact test images in Fig. 8 for prototypes 6 and 8.

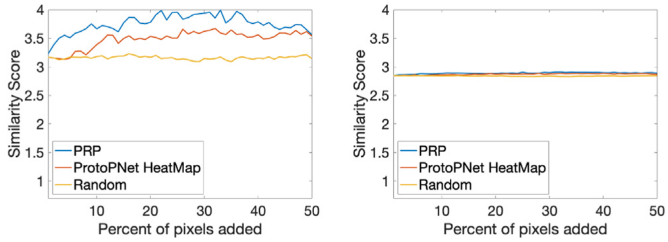


Fig. 8. CH-50: Relevance ordering test results shown for prototypes 6 and 8 of the stop sign class for the artifact test images. Both of these are not learning anything specific, therefore having high similarity with even random data.

5.4. Assessing the network behavior with PRP maps

So far, we have established the drawbacks of ProtoPNet, which are the lack of higher resolution and spatially precise explanations, which hinder the user in identifying the most relevant discriminative features. Accordingly, we proposed a method — PRP — to overcome this lack of precise explanations. Our proposed PRP maps provide a higher level of fine grained explanations while keeping the benefit of “this-looks-like-that” behavior of the ProtoPNet, as shown in Fig. 9 for both LISA (CH-50) and PASCAL VOC 2007 datasets. Therefore, we still have inherent interpretability, where each class is being represented by a fixed number of prototypes. This exponentially reduces the need for the manual laborious task of analysing individual ad-hoc explainability heatmaps for assessing deep neural networks. Additionally, this also reduces the need to use semi-automated methodologies like SpRAy [4] to find patterns in a model’s explanations with a huge number of explanation maps.

We can now directly visually identify the strategies learned by the network by only looking at a few representative prototypes for each class. For instance, we manually cluster the PRP maps of the stop sign class for the LISA dataset, as shown in Fig. 10. We can observe, that aside from learning the artifact, the network is also relying on the textual part of the stop signs as well as on the corner features. Note, that we have excluded prototypes 6 and 8 from the assessment since they did not capture any useful information (see Fig. 8).

Following this, we investigate the performance of PRP and ProtoPNet explanations on the PASCAL VOC 2007 dataset in order to uncover relevant features learned by the networks for predicting the class horse. First, we show a few prototypes (top 4 activated) that were learned by the model for the horse class along with their ProtoPNet heatmaps and PRP Maps, shown in Fig. 6 (right). Here, we can observe that PRP explanations capture the relevant features in a more fine grained manner and are able to identify a Clever Hans strategy used by the model where it tends to focus on the text in the watermark in prototype 3, rather than on the horse. In contrast, the information in ProtoPNet’s heatmaps in the second row of Fig. 6 is ambiguous since prototype 3 is allocating relevance to a broader background area. The strategies learnt by the network for recognizing a horse are grouped manually and visualized in Fig. 10. The four effective groups, disregarding the insignificant gray cluster, which focuses on the background features, represent the horse class in terms of a horse’s face, legs, presence of a rider, and finally the Clever Hans watermark.

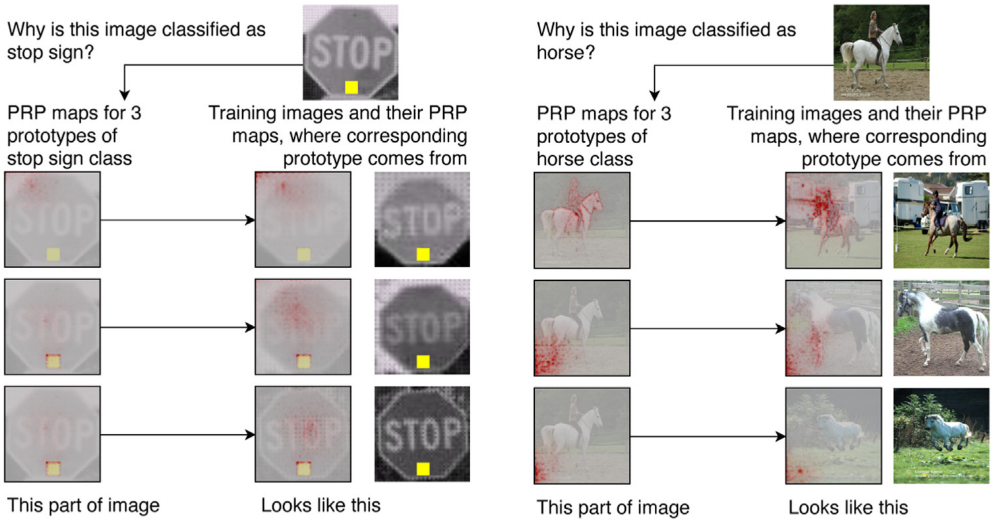

Why is this image classified as stop sign?
PRP maps for 3 prototypes of stop sign class
Training images and their PRP maps, where corresponding prototype comes from
This part of image
Looks like this
Why is this image classified as horse?
PRP maps for 3 prototypes of horse class
Training images and their PRP maps, where corresponding prototype comes from
This part of image
Looks like this

Fig. 9. *This* looks *more* like *that*: Enhanced ProtoPNet self-explainability with PRP for a LISA stop sign image from the CH-50 dataset (left) and a PASCAL VOC horse image (right).

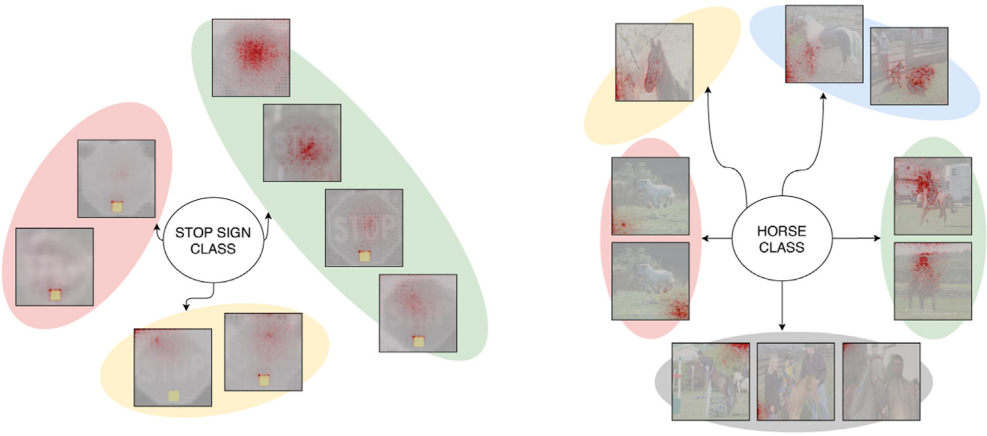


Fig. 10. Representing cluster of prototypes for the stop sign class (left) and Horse class (right). For the stop sign class, the red cluster predominantly highlights the artifact, the green cluster indicates the text, while the yellow cluster captures the corner features. For the Horse class, red cluster looks at the “Clever-Hans” i.e, the watermark in the images, the yellow cluster highlights the features of the horse’s mouth, the blue cluster indicates the presence of horse-type legs, the green cluster looks if there is a rider present, and the gray cluster captures the background features and is thus insignificant. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5.5. Multi-view clustering for suppressing artifacts

Artifacts in the data can be learned by the model, which subsequently might lead to the model exhibiting undesirable behavior, as shown in [4] and demonstrated above in case of the self-explaining network ProtoPNet. Consequently, it is essential to either remove the artifacts from the data, or to ensure that the model is not using those spurious attributes present in the data for prediction. We tried the latter in the introductory experiments on ProtoPNet — identifying and removing the artifact prototypes. However, as we observed, this is not possible since the artifact is not always perceivable by the ProtoPNet heatmaps even if the artifact was learned by a particular prototype. Using our suggested method, we are now able to find the prototypes that are activated by the artifact. It was further discovered using PRP in the previous sections, that almost all the prototypes incorporate the artifact features, thus suggesting the entanglement of the artifact information within the whole network. Therefore, instead of pruning the artifact prototypes, we propose to detect the samples in the training dataset that activate the artifact prototypes, which can be subsequently removed from the training data set before retraining the ProtoPNet on the cleansed dataset.

Using PRP, we obtain k PRP maps corresponding to the artifact-containing class for each image, where k corresponds to the number of learned prototypes for that class. We can consider these PRP maps as k different views of the same image and can thus build on existing multi-view clustering methodologies to automatically cluster the training images and thereby discover clusters corresponding to artifact-containing images. In this work, we cluster the images into 2 clusters, an artifact and a clean data cluster.

To demonstrate the efficiency of PRP in detecting artifacts in the data, we test different multi-view clustering methodologies on the LISA dataset with 50% and 20% Clever Hans features added to the stop sign images. We further use the same methodologies for backdoor detection thereby demonstrating PRP’s efficiency in multiple artifact scenarios. We also compare our clustering approach with SpRAy, which performs spectral clustering analysis on single view LRP maps, and demonstrate that our approach is able to capture better information in PRP maps, especially in the setting with multiple views.

5.5.1. Clever Hans type artifacts in 50% training data

The accuracy for CH-50 for the artifacts in the stop sign class in 100% (artifact test) and 0% (clean test) data is shown in Table 1.

Table 2
Accuracy (ACC) and F1-scores (F1) for different data scenarios with several multi-view clustering methodologies on PRP maps along with comparison with SpRAy on both PRP and LRP maps. Best and insignificantly different results, computed using *t*-test, are marked in bold.

	SpRAy-LRP[4]		**SpRAy-PRP**[4]		**CoMVC**[35]		**Co-Reg**[33]	
	ACC (%)	F1	ACC (%)	F1	ACC (%)	F1	ACC (%)	F1
CH-50	54.06±1.62	0.68±0.01	53.52±0.75	0.68±0.04	**99.99±0.00**	**0.99±0.00**	99.57 ± 0.00	0.99±0.00
CH-20	75.92±1.11	0.08±0.03	81.98±1.55	0.28±0.03	82.27±20.52	0.75±0.24	**94.54±0.00**	**0.86±0.00**
BD-15	83.18±5.76	0.21±0.24	85.72±3.87	0.30±0.15	66.85±6.91	0.76±0.06	**99.42±0.00**	**0.98±0.00**

As we can observe, the accuracy for the stop sign class drops from 100% to 94.6% when there is no artifact in the test data. From Fig. 5, prototypes 4 and 9 can be considered as “artifact” prototypes according to ProtoPNet heatmaps. But as can be seen in Table 1, there is no effect on the artifact test accuracy when removing those two prototypes. The same holds when we remove the prototypes followed by a retraining of the model. On the other hand, a decrease in the accuracy for the clean test data is observed. This additionally supports our assertion of imprecise and even misleading information provided by ProtoPNet’s heatmaps.

In order to obtain a clean data set, we aim to identify the samples that contain an artifact in the first place in order to remove them from the training set. Assuming that the information on whether an artifact is present in a data point is recognizable in the PRP maps, we cluster the PRP maps in two clusters. For comparison, we use a set of representative algorithms to cluster the data, including SpRAy [4], CoMVC [35] and Co-Reg [33]. We downsample the heatmaps to a size of 80 × 80, as this had negligible impact on the results and led to a reduced computation time.

The results for accuracy and F1-scores for the artifact cluster for different clustering methods are given in Table 2. We follow the experiments in [35] and train CoMVC for 100 epochs for 20 runs and report the results from the run resulting in the lowest unsupervised cost-function value. We repeat this 5 times and report mean and standard deviation.

As observed from Table 2, CoMVC is working very efficiently to separate the artifact images from the clean images. We also report the results for multi-view spectral clustering algorithm Co-Reg in Table 2. Although being more computationally expensive, Co-Reg is able to cluster the data effectively. Co-Reg always obtains an accuracy of above 94% in separating the artifact data, and thus prove to be highly successful in detecting the artifacts. CoMVC on the other hand performs with almost 100% accuracy when the artifact and non-artifact classes are balanced, i.e, in the current setting of CH-50.

To compare against the multi-view clustering approaches, we apply SpRAy [4], on the LRP maps for the true class (SpRAy-LRP) as well as PRP maps for the prototypes of the true class (SpRAy-PRP). For SpRAy-LRP, we compute LRP maps using the rules in Section 4.1, followed by LRP$_\epsilon$ for the last layer and a combination of relevance for all prototypes. More details are provided in the Appendix. Accordingly, we obtain one LRP map for each image, which is scaled down to 80 × 80 and flattened before applying SpRAy. For SpRAy-PRP, we combine the PRP map images by summing them across the channels and concatenating all 10 PRP maps for each image to get a 10 × 80 × 80 map. We then flatten it and apply SpRAy.

The results for both are shown in Table 2. As observed, SpRAy fails in clustering the artifacts in CH-50 data using both LRP and the concatenation of PRP maps. This behavior is expected since both SpRAy-LRP and SpRAy-PRP do not capture dependencies among multiple views of the same objects as opposed to other multi-view clustering methodologies.

Table 3
Accuracy on the artifact test (backdoor in 100% of the images in the stop sign class test data) and clean test data for BD-15, along with corresponding accuracies after removing the artifact prototype and retraining the last layer.

Test set	**BD-15**	**Remove prototype 4**	**Retraining last layer**
Artifact	1.0%	6.5%	2.5%
Clean	96.0%	96.0%	95.6%

5.5.2. Clever Hans type artifact in 20% training data

In the following, we want to capture the scenarios when less Clever Hans artifacts are included in the training data. Therefore, we evaluate the efficiency of multi-view clustering methodologies on the unbalanced dataset CH-20. The stop sign class accuracy for artifact and clean test data is 99.7% and 95.8%, respectively. This depicts that the stop sign class is still affected by the Clever Hans effect.

Applying the multi-view clustering methodologies to this scenario, we report the accuracy and F1-score in Table 2. Results show that SiMVC is performing best with 97.99% accuracy, with comparable performance by almost all the other multi-view clustering methods. SpRAy fails again with a very low F1-scores of 0.04 and 0.08 on LRP and PRP maps, clustering almost all images into one cluster.

5.5.3. Backdoor type artifact in 15% training data

Similar to the experiments above, we examine the backdoor setting, using the generated BD-15 dataset. The prototypes and their corresponding heatmaps for the speed limit class are shown in Fig. 11. The test accuracy for the case that the artifact is present in 100% of the stop sign test images is given in Table 3. Most of the stop sign images are now classified as speed limits and only 1% of the stop sign images are classified correctly.

The prototypes of the speed limit class, as learned by ProtoPNet, show that only one prototype has learned the backdoor artifact, while all the remaining 9 prototypes correspond to the speed limit class, as shown in Fig. 11. As per ProtoPNet’s explanations, removing prototype 4 of the speed limit class should solve the problem of backdoor attacks. We remove the prototype and retrain the last layer and report the accuracies in Table 3.

We can observe that removing the backdoor prototype has only a minor effect on the accuracy of the stop sign class, which increased from 1.0% to 6.5%. However, after retraining the last layer it again drops to only 2.5%. This behaviour of the network thus emphasizes the inherent learning of the backdoor artifact by the network, which is not limited to only learning a specific backdoor prototype, as incorrectly suggested by ProtoPNet visualizations. Here, the PRP explanations decode the behavior of the model as well - they indicate that almost all prototypes are activated by the artifact, even if those prototypes refer to the speed limit signs.

We therefore use multi-view clustering to clean the data of the backdoor feature and report the results in Table 2. SiMVC and CoMVC are still performing better than SpRAy-PRP with F1-scores of 0.60 and 0.57 respectively, as opposed to 0.02 F1-score of

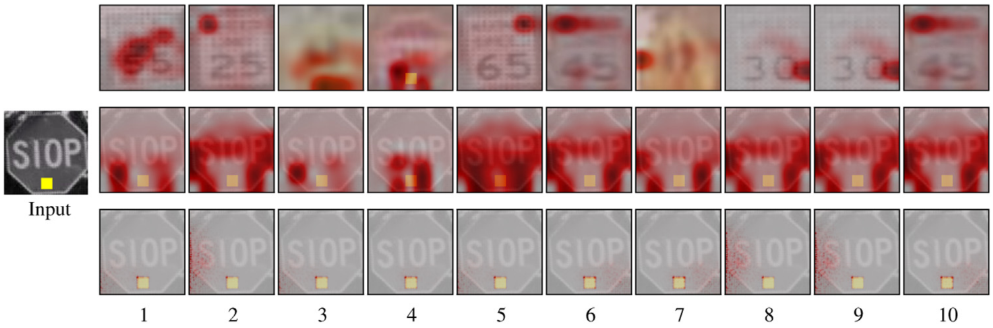


Fig. 11. BD-15: Top row depicts the learned prototypes 1 to 10 for the speed limit class with the Backdoor in 15% of the stop sign images (labeled as speed limit), the middle row depicts the ProtoPNet's heatmaps corresponding to the respective prototypes for the test image shown on the left and the bottom row shows the corresponding PRP maps for the prototypes, which capture more precise information.

SpRAy-PRP. Although, SpRAy-LRP is performing well in this setting with a F1-score of 0.91, this is due to the fact that LRP maps consist of negative relevances from the stop sign class in addition to the positive relevances from the speed limit class. This helps in accentuating the difference between speed limit and backdoor stop sign images. Furthermore, all the multi-view spectral clustering-based algorithms are able to separate these clusters efficiently, with the best being Co-Reg with an accuracy of 99.42% and a F1-score of 0.98.

6. Conclusion

Considering the success of machine learning algorithms in diverse safety-critical applications, it is instrumental to verify the behavior of these models. In this work, we assess the faithfulness of the explanations provided by a well known self-explainable network, ProtoPNet, which has subsequently been utilized as a baseline for a variety of works [23,38]. We provide an in-depth assessment of ProtoPNet's behavior in the presence of a range of artifacts. Our results indicate that, despite the attractiveness of ProtoPNet owing to its self-explaining characteristic, it is still very far from achieving the required quality of explanations. Considering this, we propose a model-aware method, PRP, to generate more precise and higher resolution prototypical explanations. These enhanced explanations help in uncovering more credible decision strategies, while keeping the self-explainability intact. We further show that these explanations are able to uncover the spurious artifact features learned by the model, which are then efficiently identified and removed via our proposed multi-view clustering strategy.

While PRP has been analysed extensively in this work, it needs to be explored further for variations of datasets as well as artifacts. So far, a limitation is the requirement of the manual analysis of clusters to distinguish the model and data heuristics despite the effective clustering performed by the proposed methodology. The behavior of the clustering further needs to be analysed in the future work in the presence of multiple artifacts per class. The design of explainable approaches with the inherent capability to leverage artifactual data in addition to clean data without capturing the artifact features would be ideal instead of removing the data and is therefore a main focus of future work. Finally, the benefit of using PRP in combination with other prototypical self-explainable models will be explored further in the future work.

The insights obtained in this work highlight the importance of evaluating the quality of self-explaining machine learning approaches and will pave the way towards the development of more robust and precise models, thereby increasing their trustworthiness.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was financially supported by the Research Council of Norway (RCN), through its Centre for Research-based Innovation funding scheme (Visual Intelligence, grant no. 309439), and Consortium Partners. The work was further partially funded by RCN FRIPRO grant no. 315029 and RCN IKTPLUSS grant no. 303514. Moreover, the work was partly supported by the German Ministry for Education and Research through the third-party funding project Explaining 4.0 (ref. 01IS20055).

ProtoPNet: Cost function

The overall cost function for ProtoPNet is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{clst}}\mathcal{L}_{\text{clst}} + \lambda_{\text{sep}}\mathcal{L}_{\text{sep}} \tag{A.1}$$

$\mathcal{L}_{\text{CE}}$ is the cross entropy (CrsEnt) loss, $\mathcal{L}_{\text{clst}}$ is the cluster loss and $\mathcal{L}_{\text{sep}}$ is the separation loss, defined as:

$$\mathcal{L}_{\text{CE}} = \min_{W} \frac{1}{N}\sum_{i=1}^{N} \text{CrsEnt}(\hat{\mathbf{y}}_i, \mathbf{y}_i) \tag{A.2}$$

$$\mathcal{L}_{\text{clst}} = \frac{1}{N}\sum_{i=1}^{N} \min_{m:\mathbf{p}_m \in \mathbf{P}_{\mathbf{y}_i}} \min_{\tilde{\mathbf{z}}} ||\tilde{\mathbf{z}} - \mathbf{p}_m||_2^2 \tag{A.3}$$

$$\mathcal{L}_{\text{sep}} = -\frac{1}{N}\sum_{i=1}^{N} \min_{m:\mathbf{p}_m \notin \mathbf{P}_{\mathbf{y}_i}} \min_{\tilde{\mathbf{z}}} ||\tilde{\mathbf{z}} - \mathbf{p}_m||_2^2 \tag{A.4}$$

where N are the total number of training images, $\mathbf{y}_i$ is the true label for image i, $\hat{\mathbf{y}}_i$ is the predicted label, W represents the learnable parameters of the whole network, $\mathbf{P}_{\mathbf{y}_i}$ are all the prototypes belonging to class $\mathbf{y}_i$ and $\tilde{\mathbf{z}}$ are the patches of the convolutional output which are of the same size as the prototypes.

SpRAy-LRP

For SpRAy based on LRP maps, we first backpropagate the output relevances i.e, class scores to the similarity score layer. We follow the $\mathbf{LRP}_{CMP}$ rule and use the LRP$_\epsilon$ rule [29]:

$$\mathbf{R}_{i\leftarrow j}^{(l,l+1)} = \frac{z_{ij}}{z_j + \epsilon \ . \ \text{sign}(z_j)}\mathbf{R}_j^{(l+1)} \tag{A.5}$$

Table A.1
Combination of classes from LISA dataset for 5-class CH-100, CH-50, CH-20 and BD-15 datasets.

Restriction signs	noRightTurn, keepRight, thruMergeLeft, thruMergeRight, thruTrafficMergeLeft, doNotPass, noLeftTurn, doNotEnter, rightLaneMustTurn
Speed limits	speedLimit40, speedLimit25, speedLimit35, speedLimit50, speedLimit45, truckSpeedLimit55, speedLimit65, speedLimit55, speedLimit30, speedLimit15, schoolSpeedLimit25
Stop signs	stopAhead, stop
Warning signs	turnLeft, signalAhead, zoneAhead25, school, curveLeft, pedestrianCrossing, curveRight, rampSpeedAdvisory50, rampSpeedAdvisoryUrdbl, dip, rampSpeedAdvisory40, merge, turnRight, slow, roundabout, speedLimitUrdbl, zoneAhead45, intersection, laneEnds, rampSpeedAdvisory45, rampSpeedAdvisory20, rampSpeedAdvisory35, addedLane
Yield signs	yield, yieldAhead

For the rest of the network, the rules for PRP are used. Considering that we are now computing relevance corresponding to all the prototypes, we combine them to get the relevance at *CONV* layer as:

$$\mathbf{R}_{ijc}^{(CONV,AM)} = \sum_{m=1}^{n} \mathbf{R}_{mijc}^{(CONV,AM)} \tag{A.6}$$

LISA 5 class dataset

An overview of the classes that were combined in the LISA dataset can be found in Table A.1.

References

[1] C. Barata, M.E. Celebi, J.S. Marques, Explainable skin lesion diagnosis using taxonomies, Pattern Recognit. 110 (2021) 107413.

[2] A.I. Aviles-Rivero, P. Sellars, C.-B. Schönlieb, N. Papadakis, GraphXCOVID: explainable deep graph diffusion pseudo-labelling for identifying COVID-19 on chest X-rays, Pattern Recognit. 122 (2022) 108274.

[3] E. Tjoa, C. Guan, A survey on explainable artificial intelligence (XAI): toward medical XAI, IEEE TNNLS 32 (11) (2021) 4793–4813.

[4] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking Clever Hans predictors and assessing what machines really learn, Nat. Commun. 10 (1) (2019).

[5] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, B. Srivastava, Detecting backdoor attacks on deep neural networks by activation clustering, SafeAI@AAAI, 2019.

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (7) (2015) 1–46.

[7] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (5) (2019) 206–215.

[8] C. Chen, O. Li, A. Barnett, J. Su, C. Rudin, This looks like that: deep learning for interpretable image recognition, NeurIPS, 2019.

[9] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes challenge 2007 (VOC2007) results, (http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html).

[10] M. Ivanovs, R. Kadikis, K. Ozols, Perturbation-based methods for explaining deep neural networks: a survey, Pattern Recognit. Lett. 150 (2021) 228–234.

[11] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should i trust you?": explaining the predictions of any classifier, in: KDD '16, 2016, pp. 1135–1144.

[12] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: NIPS '17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 4768–4777.

[13] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: visual explanations from deep networks via gradient-based localization, in: ICCV 2017, pp. 618–626.

[14] J. Sun, S. Lapuschkin, W. Samek, Y. Zhao, N. Cheung, A. Binder, Explanation-guided training for cross-domain few-shot classification, ICPR, 2021.

[15] X. Li, X. Song, T. Wu, AOGNets: compositional grammatical architectures for deep learning, CVPR 2019, 2019.

[16] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, in: NeurIPS, 2014, pp. 2204–2212.

[17] Q. Zhang, Y.N. Wu, S. Zhu, Interpretable convolutional neural networks, in: CVPR, 2018, pp. 8827–8836.

[18] Z. Chen, Y. Bei, C. Rudin, Concept whitening for interpretable image recognition, Nat. Mach. Intell. 2 (12) (2020) 772–782.

[19] J. Parekh, P. Mozharovskyi, F. d'Alché-Buc, A framework to learn with interpretation, in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 24273–24285.

[20] I. Rio-Torto, K. Fernandes, L.F. Teixeira, Understanding the decisions of CNNs: an in-model approach, Pattern Recognit. Lett. 133 (2020) 373–380.

[21] D. Alvarez Melis, T. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), NeurIPS, 2018.

[22] S. Saralajew, L. Holdijk, M. Rees, E. Asan, T. Villmann, Classification-by-components: Probabilistic modeling of reasoning over a set of components, NeurIPS, 2019.

[23] E. Kim, S. Kim, M. Seo, S. Yoon, XProtoNet: diagnosis in chest radiography with global and local explanations, in: CVPR, 2021, pp. 15714–15723.

[24] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: CVPR, 2018, pp. 1625–1634.

[25] L. Sixt, M. Granz, T. Landgraf, When explanations lie: why many modified bp attributions fail, ICML, 2020.

[26] J. Luo, J. Zhao, B. Wen, Y. Zhang, Explaining the semantics capturing capability of scene graph generation models, Pattern Recognit. 110 (2021) 107427.

[27] H. Zheng, E. Fernandes, A. Prakash, Analyzing the interpretability robustness of self-explaining models, arXiv:1905.12429(2019).

[28] A. Mogelmose, M.M. Trivedi, T.B. Moeslund, Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey, IEEE Trans. Intell. Transp. Syst. (2012) 1484–1497.

[29] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, S. Lapuschkin, Towards best practice in explaining neural network decisions with LRP, in: IJCNN, 2020, pp. 1–7.

[30] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognit. 65 (2017) 211–222.

[31] K. Chaudhuri, S.M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: ICML, 2009, pp. 129–136.

[32] M. Cheng, L. Jing, M.K. Ng, Tensor-based low-dimensional representation learning for multi-view clustering, IEEE Trans. Image Process. 28 (5) (2019) 2399–2414.

[33] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K.Q. Weinberger (Eds.), NeurIPS, 2011.

[34] L. Zong, X. Zhang, X. Liu, H. Yu, Weighted multi-view spectral clustering based on spectral perturbation, AAAI, 2018.

[35] D.J. Trosten, S. Lokse, R. Jenssen, M. Kampffmeyer, Reconsidering representation alignment for multi-view clustering, in: CVPR, 2021, pp. 1255–1265.

[36] M. Kampffmeyer, S. Løkse, F.M. Bianchi, L. Livi, A.-B. Salberg, R. Jenssen, Deep divergence-based approach to clustering, Neural Netw. (2019) 91–101.

[37] S. Kolek, D.A. Nguyen, R. Levie, J. Bruna, G. Kutyniok, A rate-distortion framework for explaining black-box model decisions, in: xxAI - Beyond Explainable AI (2020), pp. 91–115.

[38] P. Hase, C. Chen, O. Li, C. Rudin, Interpretable image recognition with hierarchical prototypes, in: HCOMP, 2019, pp. 32–40.

Srishti Gautam is currently a PhD student in the Machine Learning group at UiT The Arctic University of Norway. She previously received her MS by research degree from IIT Mandi, India in 2017. Her research interests include development of algorithms focusing on explainable AI and their application to medical and other real world data.

Marina M.-C. Höhne (née Vidovic) is a Professor at the University of Potsdam and is leading the department for Data Science at the Leibniz Institute for Agricultural Engineering and Bioeconomy (ATB). She received her PhD from TU Berlin in 2017 and continued working there as a PostDoc. She is the head of the research group UMI lab dealing with explainable artificial intelligence. Since 2021 she has a secondary employment as Associate Professor in the Machine Learning group at UiT The Arctic University of Norway. Furthermore, she is a member of the Berlin Institute for the Foundations of Learning and Data (BIFOLD) and in the ELLIS Society.

Stine Hansen received her M.Sc. degree from UiT The Arctic University of Norway in 2018. She is currently a PhD student in the Machine Learning group at UiT. Her research interests include medical image analysis and computer vision.

Robert Jenssen directs SFI Visual Intelligence (visual-intelligence.no). He is a Professor in the Machine Learning Group (machine-learning.uit.no) at UiT The Arctic University of Norway and an Adjunct Professor at the University of Copenhagen and at the Norwegian Computing Center.

Michael Kampffmeyer is an Associate Professor in the Machine Learning Group at UiT The Arctic University of Norway and a Senior Researcher at the Norwegian Computing Center. Research interests include the development of deep learning algorithms that learn from limited labeled data and their interpretability.

Paper II

Demonstrating the Risk of Imbalanced Datasets in Chest X-Ray Image-Based Diagnostics by Prototypical Relevance Propagation

Srishti Gautam, Marina M. -C. Höhne, Stine Hansen, Robert Jenssen and Michael Kampffmeyer

IEEE 19th International Symposium on Biomedical Imaging (ISBI), 2022

Paper III

ProtoVAE: A Trustworthy Self-Explainable Prototypical Variational Model

Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba A. Salahuddin, Robert Jenssen, Marina M. -C. Höhne, and Michael Kampffmeyer

Advances in Neural Information Processing Systems, 2022

ProtoVAE: A Trustworthy Self-Explainable Prototypical Variational Model

Srishti Gautam[1], **Ahcene Boubekki**[1], **Stine Hansen**[1], **Suaiba Amina Salahuddin**[1]
Robert Jenssen[1], **Marina MC Höhne**[2,1], **Michael Kampffmeyer**[1]
[1]UiT The Arctic University of Norway
[2]Technical University of Berlin

Abstract

The need for interpretable models has fostered the development of self-explainable classifiers. Prior approaches are either based on multi-stage optimization schemes, impacting the predictive performance of the model, or produce explanations that are not transparent, trustworthy or do not capture the diversity of the data. To address these shortcomings, we propose ProtoVAE, a variational autoencoder-based framework that learns class-specific prototypes in an end-to-end manner and enforces *trustworthiness* and *diversity* by regularizing the representation space and introducing an orthonormality constraint. Finally, the model is designed to be *transparent* by directly incorporating the prototypes into the decision process. Extensive comparisons with previous self-explainable approaches demonstrate the superiority of ProtoVAE, highlighting its ability to generate trustworthy and diverse explanations, while not degrading predictive performance.

1 Introduction

Despite the substantial performance of deep learning models in solving various automated real-world problems, lack of transparency still remains a crucial point of concern. The black-box nature of these high-accuracy achieving models is a roadblock in critical domains such as healthcare [1, 2], law [3], or autonomous driving [4]. This has led to the emergence of the field of explainable artificial intelligence (XAI) which aims to justify or explain a model's prediction in order to increase trustworthiness, fairness, and safeness in the application of the complex models henceforward.

Consequently, two lines of research have emerged within XAI. On the one hand, there are general methodologies explaining a posteriori black-box models, so-called post-hoc explanation methods [5, 6, 7]. While on the other hand, there are models developed to provide explanations along with their predictions [8, 9, 10]. The latter class of models, also known as self-explainable models (SEMs), are the focus of this work. Recently, many methods have been developed for quantifying post-hoc explanations [11]. However, there is still a lack of a concise definition of what SEMs should encompass, thus a lack of comparability of recent methods [12].

Methodologically, a large number of SEMs follow the approach of concept learning, analogous to prototype or basis feature learning, where a set of class representative features are learned [8, 9]. In this paper we gauge SEMs through the prism of three properties. First and foremost, the prototypes should be visualizable in the input space, and these transparent concepts should directly be employed by a glass-box classification model. Many of the existing approaches try to imitate prototype transparency by using nearest training samples to visualize the prototypes [8, 13], while some flatly use training images as prototypes preventing an end-to-end optimization and limiting the flexibility of the model [9, 14]. Secondly, the prototypes should exhibit both inter-class and intra-class diversity. Methods failing to ensure this property [9] are prone to prototype collapse into a single point which necessarily undermines their performance. Finally, SEMs should perform comparable to their black-

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

box counterparts while producing robust and faithful explanations. Previous approaches have a tendency to achieve self-explanability by sacrificing the predictive performance [9, 14, 13].

To address the aforementioned shortcomings of current SEMs, we introduce ProtoVAE, a prototypical self-explainable model based on a variational autoencoder (VAE) backbone. The architecture and the loss function are designed to produce *transparent*, *diverse*, and *trustworthy* predictions, as well as explanations, while relying on an end-to-end optimization. The predictions are linear combinations of distance-based similarity scores with respect to the prototypes in the feature space. The encoder and decoder are trained as a mixture of VAEs sharing the same network but each with its own Gaussian prior centered on one of the prototypes. The latter are enjoined to capture diverse characteristics of the data through a class-wise orthonormality constrain. Consequently, our learned prototypes are truly transparent global explanations that can be decoded and visualized in the input space. Further, we are able to generate local pixel-wise explanations by back-propagating relevances from the similarity scores. Empirically, our model corroborates trustworthiness both in terms of performance as well as the quality of its explanations.

Our main contributions can be summarized as follows:

- We define three properties for SEMs, based on which we present a novel prototypical self-explainable model with a variational auto-encoder backbone, equipped with a fully *transparent* prototypical space.
- We are able to learn faithful and *diverse* global explanations easily visualizable in the input space.
- We provide an extensive qualitative and quantitative analysis on five image classification datasets, demonstrating the efficiency and *trustworthiness* of our proposed method.

2 Predicates for a self-explainable model

For the benefit of an efficient and comprehensible formalization of SEMs, we here define three properties that we consider as prerequisites for SEMs.

Definition 1 *An SEM is **transparent** if:*

(i) its concepts are utilized to perform the downstream task without leveraging a complex black-box model;

(ii) its concepts are visualizable in input space.

Definition 2 *An SEM is **diverse** if its concepts represent non-overlapping information in the latent space.*

Definition 3 *An SEM is **trustworthy** if:*

(i) the performance matches to that of the closest black-box counterpart;

(ii) the explanations are robust, i.e., similar images yield similar explanations.

(iii) the explanations represent the real contribution of the input features to the prediction.

Note that these definitions echo properties and axioms found in other works. However, the view of such properties is diverse across the literature which leads to failure of encompassing the wide research of SEMs in general. For example, *transparency* is known as 'completeness' in [15] and 'local accuracy' in [16]. In the next section, we provide a comparison of existing SEMs based on the fulfillment of the proposed predicates.

3 Categorization of related self-explainable works

Self-explainable models optimize for both explainability and prediction, making the network inherently interpretable. As our main contribution is a prototypical model, we review and categorise existing prototypical SEMs according to the above-mentioned properties.

SENN [8] introduces a general self-explainable neural network designed in stages to behave locally like a linear model. The model generates interpretable concepts, to which sample similarities are directly aggregated to produce predictions. This generalized approach has been followed by most of the prototypical and concept-based self-explainable methods, and is also mirrored by our approach. SENN, however uses training data to provide interpretation of learned concepts, therefore approximating transparency, unlike our model which by-design has a decoder to visualize prototypes.

Table 1: Summary of the SEM properties satisfied by the baselines. The optimization scheme is also indicated. The symbol $\sim$ indicates that the concepts cannot be directly visualized in the input space and that the nearest training data serve as ersatz.

	Transparency	Diversity	Trustworthiness	Optimization
SENN[8]	$\sim$	✓	✓	End-to-end
ProtoPNet[9]	✓			Alternating
TesNET[14]	✓	✓		Alternating
SITE[17]		✓	✓	End-to-end
FLINT[13]	$\sim$	✓		End-to-end
ProtoVAE	✓	✓	✓	End-to-end

ProtoPNet [9] is a representative of a line of works [9, 14, 18, 19, 20], where a prototypical layer is introduced before the final classification layer. For maintaining interpretability, the prototypes are set as the projection of closest training image patches after every few iterations during training. Our method is closely related to ProtoPNet with the distinction of decode-able learned prototypes yielding a smooth and regularized prototypical space, thus allowing more flexibility in the model. TesNet [14] extends ProtoPNet and improves diversity at class level using five loss terms. Similarly to our approach, they distribute the base concepts among the classes and include an orthonormal constraint. However, the basis concepts are still projections from the nearest image patches, which leads to loss in predictive performance, similar to ProtoPNet. SITE [17] generates class prototypes from the input and introduces a transformation-equivariant model by constraining the interpretations before and after transformation. Since the prototypes are dynamic and generated for each test image, this method only provides local interpretations and lacks global interpretations. FLINT [13] introduces an interpreter model (FLINT-g) in addition to the original predictor model (FLINT-f). Although FLINT-f has been introduced by the authors as a framework that learns in parallel to the interpretations, it is not an SEM on its own. Therefore, we focus on FLINT-g, henceforward referred to as FLINT. FLINT takes as input features of several hidden layers of the predictor to learn a dictionary of attributes. However, the interpreter is not able to approximate the predictor model perfectly, therefore losing *trustworthiness*. Unlike prior approaches, ProtoVAE is designed to fulfill all three SEM properties. We summarize the discussed methods and their categorization in Table 1.

4 ProtoVAE

In this section, we introduce ProtoVAE, which is designed to obey the aforementioned SEM properties. Specifically, *transparency* is in-built in the architecture and further enforced along with *diversity* through the loss function. Also, we describe how our choices ensure the *trustworthiness* of our method.

4.1 Transparent architecture

In a *transparent* self-explainable model, the predictions are interpretable functions of concepts visualizable in the input space. To satisfy this property, we rely on an autoencoder-based architecture as backbone and a linear classifier. In order to have consistent, *robust*, and *diverse* global explanations, we consider prototypes in a greater number than classes. Unlike previous prototypical methods [9, 14], which update the prototypes every few iterations with the embeddings of the closest training images, ProtoVAE is trained *end-to-end* to learn both the prototypes in the feature space and the projection back to the input space. This gives ProtoVAE the flexibility to capture more general class characteristics. To further alleviate situations where some of the optimized prototypes are positioned far from the training data in the feature space, possibly causing poor reconstructions and interpretations, we leverage a variational autoencoder (VAE). VAEs are known to learn more robust embeddings and thus generate better reconstructions from out-of-distribution samples than simple autoencoders [21]. A schematic representation of the network is depicted in Fig. 1.

Details of the operations The downstream task at hand is the classification into $K > 0$ classes of the image dataset $\mathcal{X} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is an image and the one-hot vector $\boldsymbol{y}_i \in \{0, 1\}^K$

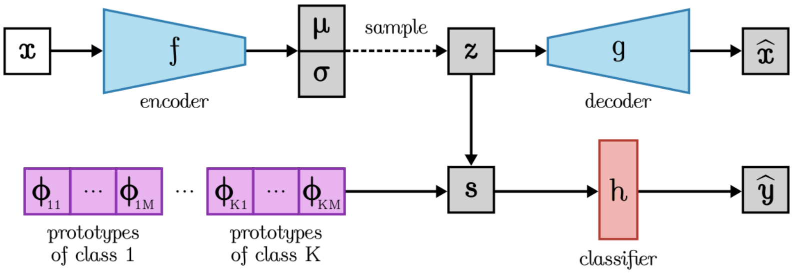


Figure 1: Schematic representation of the architecture of ProtoVAE. The input image $\boldsymbol{x}$ is encoded by f into a tuple $(\boldsymbol{\mu}, \boldsymbol{\sigma})$. A vector $\boldsymbol{z}$ is sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$ which, on one side, is decoded by g into the reconstructed input $\hat{\boldsymbol{x}}$ and, on the other side, is compared to the prototypes ϕ_{kj} resulting in the similarity scores s. The latter are passed through the classifier h to get the final prediction $\hat{\boldsymbol{y}}$.

encodes its label. The network consists of an encoder $f : \mathbb{R}^p \to \mathbb{R}^d \times \mathbb{R}^d$, a decoder $g : \mathbb{R}^d \to \mathbb{R}^p$ ($d < p$), M prototypes per class $\boldsymbol{\Phi} = \{\phi_{kj}\}_{j=1..M}^{k=1..K}$, a similarity function $\text{sim} : \mathbb{R}^d \to \mathbb{R}^M$ and a glass-box linear classifier $h : \mathbb{R}^M \to [0\,, 1]^K$. An image $\boldsymbol{x}_i$ is first transformed by f into a tuple $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) = f(\boldsymbol{x}_i)$ which, in the VAE realm, are the parameters of the posterior distribution. A feature vector $\boldsymbol{z}_i$ is then sampled from the normal distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ and is used twice. On the one hand, it is decoded as $\hat{\boldsymbol{x}}_i = g(\boldsymbol{z}_i)$. On the other hand, it is compared to the prototypes. We use the same similarity function as in [9] and obtain the resulting vector $\boldsymbol{s}_i \in \mathbb{R}^{K \times M}$ as:

$$\boldsymbol{s}_i(k, j) = \text{sim}(\boldsymbol{z}_i, \boldsymbol{\phi}_{kj}) = \log\left(\frac{||\boldsymbol{z}_i - \boldsymbol{\phi}_{kj}||^2 + 1}{||\boldsymbol{z}_i - \boldsymbol{\phi}_{kj}||^2 + \epsilon}\right), \tag{1}$$

with $0 < \epsilon < 1$. Finally, $\boldsymbol{s}_i$ is used to compute the predictions: $\hat{\boldsymbol{y}}_i = h(\boldsymbol{s}_i)$. Moreover, the similarity vector $\boldsymbol{s}_i$ captures the distance to the prototypes but also indicates the influence of each prototype on the prediction.

4.2 Diversity and trustworthiness

Unlike transparency, diversity cannot be achieved solely through the architectural choices. It needs to be further enforced during the optimization. Our architecture implies two loss terms: a classification loss and a VAE-loss. Without further regulation, our model is left vulnerable to the curse of prototype collapse [14, 22] which would undermine the SEM *diversity* property. We prevent such a situation with a third loss term enforcing orthonormality between prototypes of the same class. The loss function of ProtoVAE can thus be stated as follows:

$$\mathcal{L}_{\text{ProtoVAE}} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{orth}} + \mathcal{L}_{\text{VAE}}. \tag{2}$$

We detail now each term and discuss how they favor *diversity* and *trustworthiness*.

Inter-class diversity through classification Although the prototypes are assigned to a class, the classifier is blind to that information. Thus, the prediction problem is a classic classification that we solve using the cross-entropy loss.

$$\mathcal{L}_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{CE}(h(\boldsymbol{s}_i); \boldsymbol{y}_i). \tag{3}$$

Since h is linear, the loss pushes the embedding of each class to be linearly separable, yielding a greater *inter-class diversity* of the prototypes.

Intra-class diversity through orthonormalization The inter-class diversity is guaranteed by the previous terms. However, without further regularization, the prototypes might collapse to the center of the class, obviating the possibilities offered by the extra prototypes. To prevent such a situation and foster intra-class diversity, we enforce the prototype of each class to be orthonormal to each other as follows:

$$\mathcal{L}_{\text{orth}} = \sum_{k=1}^{K} ||\bar{\boldsymbol{\Phi}}_k^T \bar{\boldsymbol{\Phi}}_k - \boldsymbol{I}_M||_F^2, \tag{4}$$

where $\boldsymbol{I}_M$ is the identity matrix of dimension $M \times M$ and the column-vectors of matrix $\bar{\boldsymbol{\Phi}}_k$ are the prototypes assigned to class k minus their mean, i.e., $\bar{\boldsymbol{\Phi}}_k = \{\boldsymbol{\phi}_{kj} - \bar{\boldsymbol{\phi}}_k,\ j = 1\ ..\ M\}$ with $\bar{\boldsymbol{\phi}}_k = \sum_{l=1}^{M} \boldsymbol{\phi}_{kl}$. Beyond regularizing the Frobenius norm $||.||_F$ of the prototype, this term favors the disentanglement of the captured concepts within each class, which is one way to obtain *intra-class diversity*.

Robust classification and reconstruction through VAE The VAE architecture ensures the robustness of the embedding and of the decoder. In its original form, the VAE loss considers a single standard normal distribution as a prior and is trained to minimize:

$$||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 + D_{\mathrm{KL}}\big(p_f(\boldsymbol{z}|\boldsymbol{x})||\mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)\big), \tag{5}$$

where $\boldsymbol{I}_d$ is the identity matrix of dimension $d \times d$. Such an objective enjoins the embedding to organize as if generated by a single Gaussian distribution, thus making it difficult to split it with the linear classifier h. To help the classifier, we consider instead a mixture of VAEs sharing the same network each with a Gaussian prior centered on one of the prototypes. Since, each prototype has a label, only data-points sharing that label are involved in the training of the associated VAE. The loss function of our mixture of VAEs is (derivation in the supplementary material Sec. S2):

$$\mathcal{L}_{\mathrm{VAE}} = \frac{1}{N}\sum_{i=1}^{N} ||\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i||^2 + \sum_{k=1}^{K}\sum_{j=1}^{M} \boldsymbol{y}_i(k) \frac{\boldsymbol{s}_i(k,j)}{\sum_{l=1}^{M} \boldsymbol{s}_i(k,l)} D_{\mathrm{KL}}\big(\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)||\mathcal{N}(\boldsymbol{\phi}_{kj}, \mathbf{I}_d)\big). \tag{6}$$

In addition to training the decoder, this loss enjoins the embedding to gather closely around their class prototypes.

4.3 Visualization of explanations

ProtoVAE is designed to have the inherent capability to reconstruct prototypes via the decoder, which is trained to approximate the input distribution. Additionally, to generate faithful pixel-wise local explanation maps, we build upon the concepts of Layer-wise relevance propagation (LRP) [23] which is a model-aware XAI method computing relevances based on the contribution of a neuron to the prediction. Following [12], we generate explanation maps, where for each prototype, the similarity of an input to the prototype is backpropagated to the input image according to the LRP rules. For an input image $\boldsymbol{x}_i$, the point-wise similarity between the transformed mean vector $\boldsymbol{\mu}_i$ with a prototype $\boldsymbol{\phi}_{kj}$ is first calculated as:

$$\gamma_{ikj} = \frac{1}{\boldsymbol{d}_{ikj} + \eta} \quad \text{with} \quad \boldsymbol{d}_{ikj} = (\boldsymbol{\mu}_i - \boldsymbol{\phi}_{kj}) * (\boldsymbol{\mu}_i - \boldsymbol{\phi}_{kj}), \tag{7}$$

where $*$ is the Hadamard element-wise product and $\eta > 0$. The similarity γ_{ikj} is then backpropagated through the encoder following LRP composite rule, which is known as best practice [24] to compute local explanation maps. Following this, the LRP$_{\alpha\beta}$ rule is applied to the convolutional layers and the Deep Taylor Decomposition based rule DTD$_{z^B}$ [25] is applied to the input features.

5 Experiments

In this section, we conduct extensive experiments to evaluate ProtoVAE's trustworthiness, transparency, and ability to capture the diversity in the data. More specifically, we demonstrate the trustworthiness of our model in terms of predictive performance in Sec. 5.1. Qualitative evaluations are then conducted in Sec. 5.2 to verify the diversity and transparency properties, followed by a quantitative evaluation of the explanations corroborating its trustworthiness. Additionally, we provide an ablation study for the terms in Eq. 2 and further study the effect of the L2 norm in Eq. 6 on the prototype reconstructions in the supplementary material in Sec. S6.1 and Sec. S6.9, respectively.

Datasets and implementation: We evaluate ProtoVAE on 5 datasets, MNIST [26], FashionMNIST [27] (fMNIST), CIFAR-10, [28], a subset of QuickDraw [29] and SVHN [30]. We use small encoder networks with 4 convolution layers for MNIST, fMNIST and CIFAR-10, 3 for QuickDraw and 8 for SVHN. These convolution layers are followed by 2 linear layers which gives us the tuple $(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$ for each image i. The decoder mirrors the encoder's architecture. Similar to [9], we fix

Table 2: Performance results of ProtoVAE compared to other state-of-the-art methods (measured in accuracy (in %)). The reported numbers are means and standard deviations over 4 runs. Best and statistically non-significantly different results are marked in bold. *Results for SITE are taken from the original paper and thus based on more complex architectures.

	Black-box encoder	FLINT [13]	SENN [8]	*SITE [17]	ProtoPNet [9]	ProtoVAE
MNIST	99.2±0.1	**99.4±0.1**	98.8±0.7	98.8	94.7±0.6	**99.4±0.1**
fMNIST	91.5±0.2	91.5±0.2	88.3±0.3	-	85.4±0.6	**91.9±0.2**
CIFAR-10	83.9±0.1	79.6±0.6	76.3±0.2	84.0	67.8±0.9	**84.6±0.1**
QuickDraw	86.7±0.4	82.6±1.4	79.3±0.3	-	58.7±0.0	**87.5±0.1**
SVHN	**92.3±0.3**	90.8±0.4	91.5±0.4	-	88.6±0.3	**92.2±0.3**

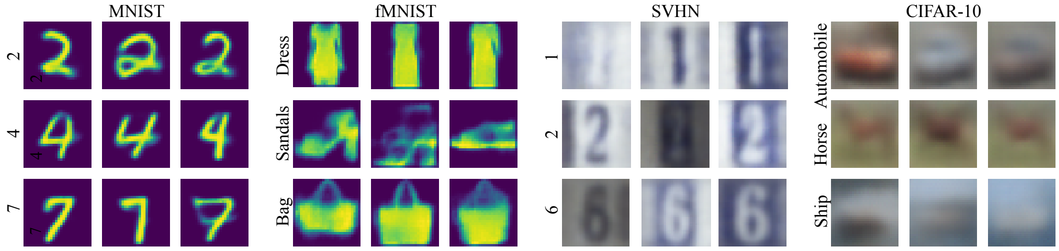


Figure 2: Visualization of learned prototypes for different classes for MNIST, fMNIST, SVHN and CIFAR-10.

the prototypes per class, M, to 5 for MNIST and SVHN and 10 for the other datasets. Further details about the datasets and additional implementation details, such as the detailed architecture and hyperparameters, are provided in the supplementary material Sec. S3 and S4. Our code is available at https://github.com/SrishtiGautam/ProtoVAE.

Baselines To ensure a fair comparison, we modified the publicly available code of ProtoPNet, FLINT and SENN to use the same backbone network as ProtoVAE and when relevant the same number of prototypes per class as used for ProtoVAE. We also provide the results for the predictive performance of SITE as reported in [17], since the code is not publicly available. We also report the performance of our model with a ResNet-18 backbone in the supplementary material Sec. S6.2. Further, we compare ProtoVAE using FLINT's encoders as provided in [13] for both FLINT and SENN in Sec. S6.3 in the supplementary material. Finally, we also compare with the black-box counterpart of our model, i.e, a classical feed-forward CNN based on the same encoder as ProtoVAE but followed by a linear classifier and trained end-to-end with the cross-entropy loss. This black-box encoder model is thus free from all regularization necessary for self-explainability.

5.1 Evaluation of predictive performance

In the Table 2, we can observe that ProtoVAE surpasses all other SEMs in terms of predictive performance on all five datasets, which is based on its increased flexibility in the architecture. For ProtoPNet, we observe a gap in performance, which is due to the low number of optimal class-representatives in the actual training data. This creates a huge bottleneck at the prototype layer and therefore limits its performance. Further, and more importantly, when compared to the true black-box counterpart, ProtoVAE achieves no loss in accuracy and is even able to perform better on all the datasets. We believe this is due to an efficient over-clustering of the latent space with the flexible prototypes, as well as the natural regularizations achieved through the VAE model. These results strengthens the *trustworthiness* of ProtoVAE in terms of the predictive performance.

5.2 Evaluation of explanations

Qualitative evaluation The demonstrated results in this section strengthen the fulfilment of the *transparency* property by providing human-understandable explanations for ProtoVAE. We visualize the decoded prototypes for different datasets, which act as global explanations for the corresponding

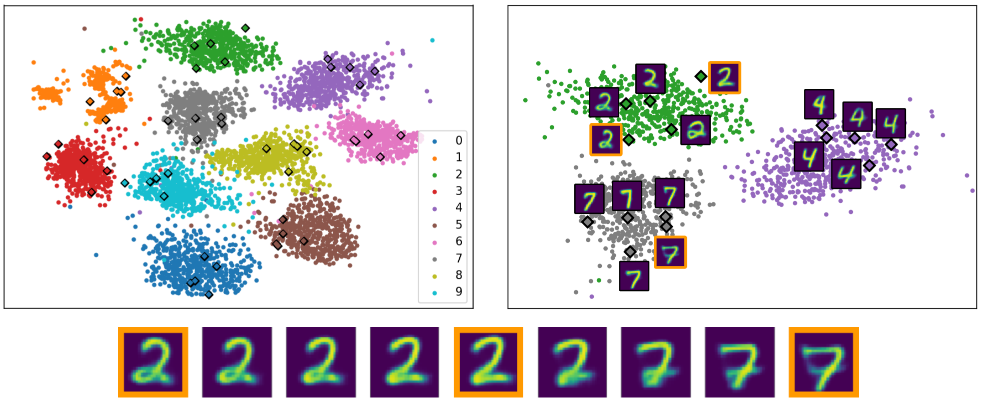


Figure 3: UMAP representations for the prototypical space for MNIST (left), decoded prototypes overlayed for classes 2, 4 and 7 (right), and interpolation between prototypes of the same class (2) and between prototypes of different classes (2-7) (bottom).

classes, in Fig. 2 [1]. The prototypes for MNIST demonstrate that class 2 consist of 2's with flat bottom line or with rounded bottom lines. For fMNIST, the sandals class consists of both heels and flats. The class prototypes thus directly help visualizing the components of the classes by looking at a fixed number of prototypes per class instead of all the training data. Interestingly, although SVHN often contains multiple digits of different classes in the same training image, our prototypes efficiently capture only one digit representing its class. Moreover, a blurring effect is observed in our prototypes which captures more variability and therefore suggests efficient representation of the true "mean" of a subset of a given class, as opposed to other methods [9, 14, 13] which show the closest training images and are therefore sharper. This behavior supports our claim of more flexibility in the network, therefore enhancing predictive capability along with the ability to provide more faithful explanations. This blurring effect is observed to be more prominent in CIFAR-10, which is due to the high complexity in each class in the dataset and can be reduced by using a larger number of prototypes per class.[2] Additionally, to provide more clear visualization of the learned transparent prototypical space, we show UMAP representations of the prototypes and the training data for MNIST in Fig. 3. This visualization further illustrates the inter-class as well as the intra-class *diversity* of the prototypes. Moreover, due to the regularized prototypical space, we are efficiently able to interpolate between prototypes both within a class and between classes, therefore making the latent space fully *transparent*. In Fig. 3, we interpolate between 2 different prototypes of class '2' and from a prototype of class '2' to a class '7' prototype.

The local explanability maps for a test image according to the three closest, i.e. most similar, prototypes for both ProtoVAE and ProtoPNet are shown in Fig. 4, along with the corresponding similarity scores. As seen, different prototypes of the same class activate different parts of the same test image, which therefore helps in achieving better performance. The ProtoPNet maps are extremely coarse which therefore makes them challenging to interpret. Therefore, we overlay the heatmaps over the input image for ProtoPNet. As observed, the most activated prototypes do not belong to the same class as the test image. This might happen because of ProtoPNet focusing on patches in prototypes, therefore losing contextual information. The 3 closest prototypes shown for the image 'apple' belong to class 'lion'. Further, an uninformative training image, which is not seen in the ProtoVAE prototypes, has been selected by ProtoPNet to represent 5 out of 10 prototypes for class 'lion'. The remaining 5 prototypes are represented by 1 other same training image. This effect is seen predominantly in ProtoPNet where the prototypes of the same class collapse to one point and are thus represented by the same training image, therefore dissatisfying the *diversity* property, as opposed to ProtoVAE. The prototypes for class 'lion' for both the models are included in the supplementary material in Sec. S6.4.

[1] As a reference to gauge the quality and sharpness of the pictures of Fig. 2, reconstructions of test images are provided in Sec. S6.11.

[2] We demonstrate this behavior in Sec. S6.8 and show in Sec. S6.7 how local explainability maps can be used to gather additional information about pixel-wise relevances thereby counterbalancing blurry prototypes.

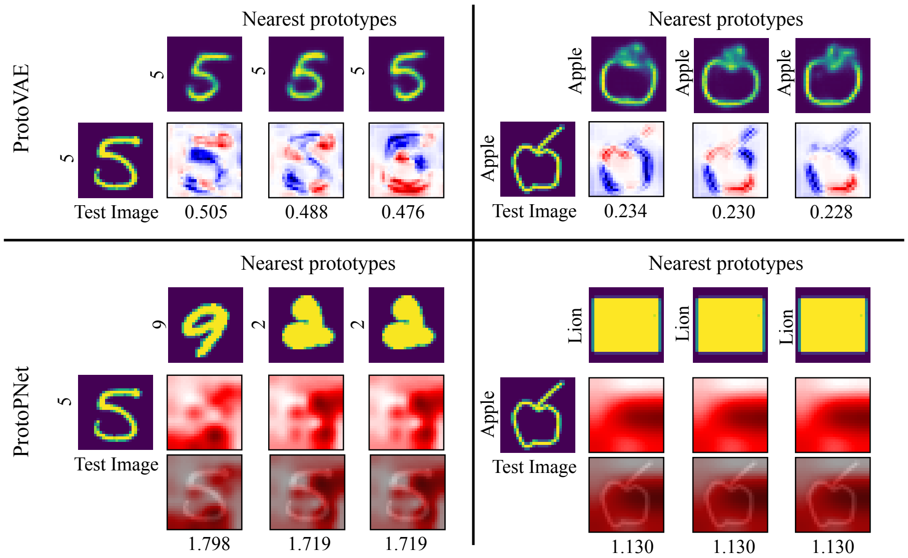


Figure 4: Three maximally activated prototypes, the corresponding prototypical activations, and corresponding similarity scores for a test image of class 5 (for MNIST) and apple (for QuickDraw), for both ProtoVAE and ProtoPNet models.

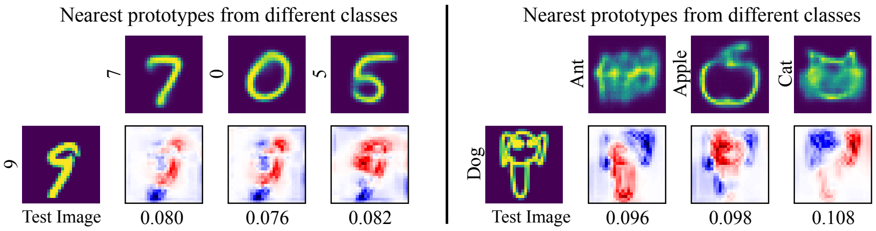


Figure 5: Maximally activated prototypes from three random classes, along with the prototypical explanations for MNIST (left) and QuickDraw (right) datasets.

We also show the closest prototypes from 3 different random classes and their corresponding explainability maps to demonstrate the behavior of explanations for different class prototypes in Fig. 5. Interestingly, the 'dog' image from the QuickDraw dataset resembles an 'ant' prototype for the legs, an 'apple' prototype for the face and a 'cat' prototype for the ears. This information provided by the local explainability maps thus aligns well with human-understandable concepts.

To compare the efficacy of the mapping to the input space learned by our decoder, to methodologies with training-data projection of prototypes [9, 14], we show prototypes along with the 3 closest training images for different datasets in Fig. 6. The prototypes are observed to be the representative of a subset of the respective class. For example, the prototype shown for class '4' of MNIST is representing the subset of '4' with an extended bar, while the 'banana' prototype represents the left facing 'banana' subset, and the 'dog' prototype represents the subset of white dogs on a darker background.

Finally, in order to demonstrate the scalability of ProtoVAE and its applicability on complex higher-resolution real world datasets, we provide an analysis on the CelebA dataset [31] in Sec. S6.10. Note that the less important and fairly diverse features (such as background) appear blurry, while the more important features (skin color, hair color, hair style or age) are crisp and clearly visible.

Quantitative evaluation To quantify the *trustworthiness* of the explanations provided by the proposed model, we calculate the Average Drop (AD) and Average Increase (AI) with respect to local explanation maps and similarity scores for all prototypes [32, 2]. The AD measures the decrease in similarity scores with respect to each prototype when the 50% least important pixels are removed from the images, while AI estimates the ratio of increasing similarity scores. A low AD and a high

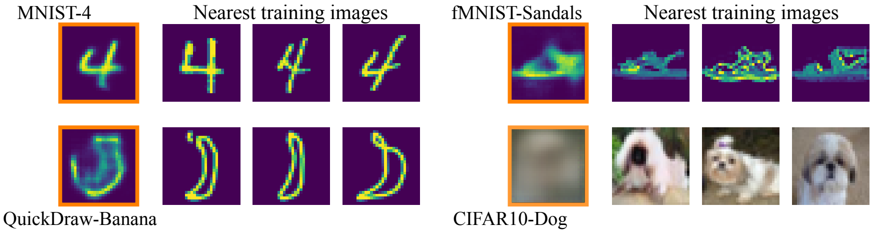


Figure 6: The three closest training images to the learned prototypes for MNIST (class '4'), QuickDraw (class 'Banana'), fMNIST (class 'Sandals') and CIFAR-10 (class 'Dog'), proving our prototypes representing a "real mean" of subset of classes.

Table 3: AD and AI for quantitative evaluation of explanations of ProtoVAE and ProtoPNet. The reported numbers are means and standard deviations over 5 runs. Best and statistically non-significantly different results are marked in bold.

	MNIST		fMNIST		CIFAR-10		QuickDraw		SVHN	
	AD	AI	AD	AI	AD	AI	AD	AI	AD	AI
ProtoPNet	3.4±0.3	**0.6±0.0**	7.2±0.4	0.5±0.0	11.6±0.2	0.5±0.0	2.6±0.1	0.7±0.0	**5.4±0.0**	**0.7±0.0**
ProtoVAE	**0.4±0.0**	**0.6±0.0**	**5.1±0.0**	**0.8±0.0**	**6.6±0.0**	**0.7±0.0**	**0.1±0.0**	**0.9±0.0**	6.1±0.1	**0.7±0.0**

AI suggest better performance. These scores are computed as follows:

$$\text{AD} = \frac{100}{NKM}\sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{M}\frac{\max\left(0, \boldsymbol{s}_i(k,j) - \boldsymbol{s}_i^{50\%}(k,j)\right)}{\boldsymbol{s}_i(k,j)}, \quad \text{AI} = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{M}\frac{[[\boldsymbol{s}_i(k,j) < \boldsymbol{s}_i^{50\%}(k,j)]]}{NKM},$$

where $s_i(k,j)$ is the similarity score of an image i with prototype j of class k (see Eq.1) and $\boldsymbol{s}_i^{50\%}(k,j)$ is the similarity score after masking the 50% least activated pixels according to the prototypical explanation map of prototype j. Also, $[[\cdot]]$ are the Iverson brackets which take the value 1 if the statement they contain is satisfied and 0 otherwise.

We report the mean and standard deviation for AD and AI computed over 5 random subsets of 1000 test images for ProtoVAE and ProtoPNet in Table 3. For the grayscale datasets, MNIST and fMNIST, the masked pixels are replaced by 0. For CIFAR-10 and SVHN, they are replaced by random uniformly sampled values. ProtoVAE achieves considerably lower AD and higher or comparable AI for all the datasets. For SVHN, ProtoPNet performs well which we believe is due to the abundance of representative patches in the dataset, thereby improving its explanations.

Finally, we perform a relevance ordering test [33, 12], where we start from a random image and monitor the predicted class probabilities while gradually adding a percentage of the most relevant pixels to the random image according to the local explanation maps. We take 100 random test images and report the average results of change in predicted class probability for all the prototypes in the model. The rate distortion graphs are shown in Fig. 7 for MNIST, QuickDraw and SVHN. We also include two baselines, Random-ProtoPNet and Random-ProtoVAE, where the pixel relevances are ordered randomly. Larger area under the curve indicates better performance. As shown, ProtoVAE's local explanations are able to capture more relevant information than ProtoPNet for all three datasets. Further, for MNIST, ProtoPNet is performing even worse than Random-ProtoPNet, highlighting the lack of trustworthiness in ProtoPNet's explanations.

6 Conclusion and Discussion

In this work, we define three properties that act as prerequisites for efficient development of SEMs, namely, *transparency*, *diversity*, and *trustworthiness*. We then introduce ProtoVAE, a prototypical self-explainable method, based on a variational auto-encoder backbone, which addresses these three properties. ProtoVAE incorporates a transparent model and enforces diversity and trustworthiness through the loss functions. In addition to providing faithful explanations, ProtoVAE is able to achieve better predictive performance than its counterpart black-box models.

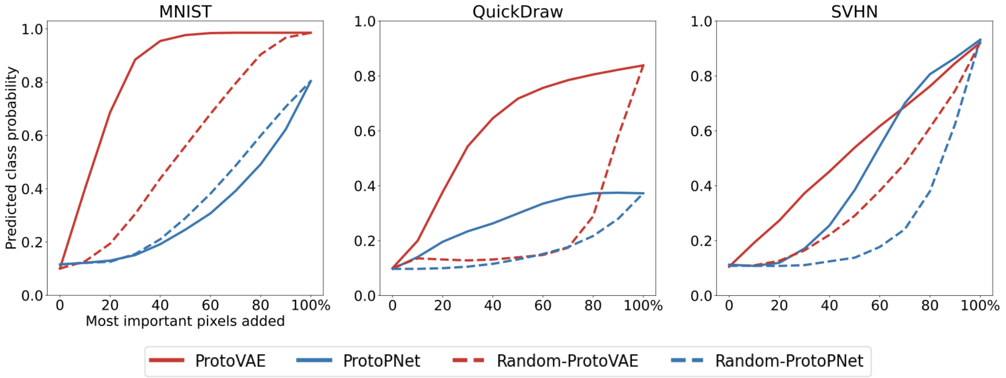


Figure 7: Relevance ordering test for ProtoPNet and ProtoVAE, along with the respective random baselines (Random-ProtoPNet and Random-ProtoVAE). Higher curve suggests better performance of ProtoVAE for all 3 datasets of MNIST, QuickDraw and SVHN.

The main limitation of ProtoVAE is the fixed number of prototypes. This means that the model has to grasp simple as well as more complex classes with the same number of prototypes. For example, in MNIST, there are more variations to be captured by the prototypes in the class ‘4’ than in class ‘1’. A simple but effective solution is a distance-based pruning procedure, which will be explored in future works. Another approach in sight is to use a prior on the distribution of the prototypical similarities and prioritize some prototypes by controlling the frequency with which each prototype is used in the predictions. Finally, since our global explanations can only be as good as the decoder, one more promising research direction is to leverage more expressive generative models, such as "Very Deep VAEs" [34] and normalizing flows [35] to further improve the scalability of the method to more complex datasets.

Acknowledgments and Disclosure of Funding

This work was financially supported by the Research Council of Norway (RCN), through its Centre for Research-based Innovation funding scheme (Visual Intelligence, grant no. 309439), and Consortium Partners. The work was further partially funded by RCN FRIPRO grant no. 315029 and RCN IKTPLUSS grant no. 303514. Moreover, the work was partly supported by the German Ministry for Education and Research through the third-party funding project Explaining 4.0 (ref. 01IS20055).

References

[1] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

[2] Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2022.

[3] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

[4] Daniel Omeiza, Helena Webb, Marina Jirotka, and Lars Kunze. Explanations in autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–21, 2021.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. KDD ’16, page 1135–1144, 2016.

[6] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[7] Jason Yosinski, Jeff Clune, Anh M Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *ICML Deep Learning Workshop*, 2015.

[8] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[9] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: Deep learning for interpretable image recognition. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[10] Sascha Saralajew, Lars Holdijk, Maike Rees, Ebubekir Asan, and Thomas Villmann. Classification-by-components: Probabilistic modeling of reasoning over a set of components. In *NeurIPS*, 2019.

[11] Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. 2022.

[12] Srishti Gautam, Marina M. C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *arXiv*, 2021.

[13] Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché-Buc. A framework to learn with interpretation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24273–24285. Curran Associates, Inc., 2021.

[14] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 875–884, 2021.

[15] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

[16] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[17] Yipei Wang and Xiaoqian Wang. Self-interpretable model with transformation equivariant interpretation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2359–2372. Curran Associates, Inc., 2021.

[18] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: Diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15719–15728, June 2021.

[19] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining*, KDD '21, page 1420–1430, New York, NY, USA, 2021. Association for Computing Machinery.

[20] Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14933–14943, June 2021.

[21] Alexander Camuto, Matthew Willetts, Stephen Roberts, Chris Holmes, and Tom Rainforth. Towards a theoretical understanding of the robustness of variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 3565–3573. PMLR, 2021.

[22] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[23] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

[24] M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin. Towards best practice in explaining neural network decisions with lrp. In *IJCNN*, pages 1–7, 2020.

[25] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[29] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[32] Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. Relevance-cam: Your model already knows where to look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14944–14953, June 2021.

[33] Jan MacDonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. A rate-distortion framework for explaining neural network decisions. *ArXiv*, abs/1905.11092, 2019.

[34] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

[35] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Discussed in section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Included in the supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Provided in the supplementary material.

    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Details reported in the supplementary material.
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Mean and standard deviation for accuracy are reported.
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Included in the supplementary material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [Yes]
    (b) Did you mention the license of the assets? [Yes] Included in the supplementary material.
    (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Provided in the supplementary material.
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] All the datasets used in this work are open-source.
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets used are open-source and do not contain personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Paper IV

Prototypical Self-Explainable Models Without Re-training

Srishti Gautam, Ahcene Boubekki, Marina M. -C. Höhne, and Michael Kampffmeyer

Under Review

Prototypical Self-Explainable Models Without Re-training

Srishti Gautam *srishti.gautam@uit.no*
Department of Physics and Technology
UiT The Arctic University of Norway, Norway

Ahcene Boubekki *ahcene.boubekki@ptb.de*
Machine Learning and Uncertainty
Physikalisch-Technische Bundesanstalt, Germany

Marina M. C. Höhne *marina.hoehne@uni-potsdam.de*
Data Science in Bioeconomy
University of Potsdam, Germany

Michael C. Kampffmeyer *michael.c.kampffmeyer@uit.no*
Department of Physics and Technology
UiT The Arctic University of Norway, Norway

Abstract

Explainable AI (XAI) has unfolded in two distinct research directions with, on the one hand, post-hoc methods that explain the predictions of a pre-trained black-box model and, on the other hand, self-explainable models (SEMs) which are trained directly to provide explanations alongside their predictions. While the latter is preferred in most safety-critical scenarios, post-hoc approaches have received the majority of attention until now, owing to their simplicity and ability to explain base models without retraining. Current SEMs instead, require complex architectures and heavily regularized loss functions, thus necessitating specific and costly training. To address this shortcoming and facilitate wider use of SEMs, we propose a simple yet efficient universal method called KMEx (K-Means Explainer), which can convert any existing pre-trained model into a prototypical SEM. The motivation behind KMEx is to push towards more transparent deep learning-based decision-making via class-prototype-based explanations that are guaranteed to be diverse and trustworthy without retraining the base model. We compare models obtained from KMEx to state-of-the-art SEMs using an extensive qualitative evaluation to highlight the strengths and weaknesses of each model, further paving the way toward a more reliable and objective evaluation of SEMs.

1 Introduction

XAI has become a key research area with the primary objective of enhancing the reliability of deep learning models (Yosinski et al., 2015; Tjoa & Guan, 2021). This domain has notably evolved along two parallel trajectories in recent years. One focuses on post-hoc methods (Ribeiro et al., 2016; Selvaraju et al., 2017), where the algorithms aim to explain the behavior of the black-box models *after* they have been trained. The other promising branch focuses on SEMs (Rudin, 2019), where the models are strategically designed and trained to generate explanations *along with* their predictions.

The easily employable post-hoc techniques have become widely adopted in recent works due to their ability to offer insights into any black-box models without retraining (Bodria et al., 2023). Nevertheless, the need for inherently interpretable models has taken some momentum fueled by the unreliability and high variability of these post-hoc methods which inhibits their usability for safety-critical applications (Rudin, 2019). SEMs offer explanations that align with the actual computations of the model, thus proving to be more dependable which is crucial in domains such as criminal justice, healthcare, and finance (Rudin, 2019). However, existing

SEMs rely on complex designs based on large deep-learning backbones and require intricate training strategies. The associated computational and time costs limit their accessibility and sustainability.

We tackle this limitation by introducing a simple but efficient method called KMEx (K-Means Explainer), which is the first approach that aims to convert a trained black-box model into a prototypical self-explainable model (PSEM). PSEMs provide inherent explanations in the form of class-representative concepts, also called prototypes, in the latent space that can be visualized in the human-understandable input-space (Kim et al., 2021). These prototypes serve as global explanations of the model (*this looks like that* (Chen et al., 2019b)), and their visualization provides knowledge about their neighborhood in the learned embedding. KMEx keeps the trained encoder intact, learns prototypes via clustering in the embedding space, and replaces the classifier with a transparent one. This results in an SEM with similar local explanations and performance to the original black-box model and such enables the reuse of existing trained models.

Comparing models obtained using KMEx to existing PSEMs requires a comprehensive evaluation strategy which, for this fairly new field, is still lacking. Differing from conventional black-box classifiers, PSEMs yield global (prototypes' visualization) and/or local (activation of individual prototypes by input images) explanation maps, alongside the predicted class probabilities. Yet, the assessment of SEMs until now has been limited to comparing the predictive performance to the black-box counterpart with the same backbone architecture as the SEM, followed by quantifying the robustness of local explanation maps and qualitative evaluation of global explanations Wang & Wang (2021); Parekh et al. (2021). We argue that this approach overlooks crucial facets of SEM explainability, failing to establish a standardized framework for thorough analysis and comparison of existing models. For example, we observe that most of the prototypes learned by recent SEMs might never be used by their classifier, which challenges the rationale of a transparent model. Further, the diversity captured by different prototypes in the embedding space, while being a driving force behind the development of several SEMs (Wang et al., 2021), has traditionally only been validated by highly subjective visual inspection of the prototypes.

We, therefore, present a novel quantitative and objective evaluation framework based on the three properties that arose as predicates for SEMs (Gautam et al., 2022): transparency, diversity, and trustworthiness. The rationale is not to rank models but to highlight the consequences of modeling choices. Indeed, in some applications, having robust local explanations might be more valuable than diverse prototypes. Yet, this behavior needs to be quantified in order to support practitioners in choosing the best model for their use case.

Our main contributions are thus as follows:

- We propose a simple yet efficient method, KMEx, which converts any existing black-box model into a PSEM, thus enabling wider applicability of SEMs.
- We propose a novel quantitative evaluation framework for PSEMs, grounded in the validation of SEM's predicates (Gautam et al., 2022), which allows for an objective and comprehensive comparison.

Our key findings are as follows:

- Experiments on various datasets confirm that KMEx matches the performance of the black-box model while offering inherent interpretability without altering the embedding, making it an efficient benchmark for SEMs.
- Most existing PSEMs tend to *ghost* the prototypes, i.e., never utilize them for prediction, which gives a false sense of needed concepts but also undermines the rationale formalized by the predicates, especially transparency.
- Unlike KMEx, the large variations in the design and regularizations of other SEMs lead to drastically different learned representation spaces and local explanations.
- While many SEMs incorporate measures to obtain diverse prototypes, these efforts are not necessarily reflected in terms of captured input data attributes. We illustrate how KMEx can be leveraged, without the need for retraining, to improve the prototype positioning on the SEM's embeddings and to better cover the attributes and their correlations.

Table 1: Design strategies used by state-of-the-art SEMs.

	Similarity Measure	**Classifier**	**Prototypes**	**Diversity Loss**
ProtoPNet	Distance based	Linear Layer	Projected from training data	Min/max intra/inter-class distance
FLINT	Linear Layer	Linear Layer	Weight of the network	Min/max similarity entropy
ProtoVAE	Distance based	Linear Layer	Learned ad-hoc parameters	Orthonormality + KL Divergence
KMEx	Distance based	Nearest Neighbor	k-means	Clustering

2 Prototypical self-explainable models

In this section, we review the recent literature on PSEMs for the task of image classification, which is the focus of this work, emphasizing their design considerations as well as evaluation approaches.

PSEMs for image classification typically consist of four common components: an encoder, a set of prototypes, a similarity function, and a transparent classifier. The encoder is typically sourced from a black-box model, thereby making the latter the *closest* (to the SEM architecture) natural baseline for comparison until now. Prototypes are class-concepts that live in the embedding space and serve as global class explanations, i.e., representative vectors, that eliminate the necessity to examine the entire dataset for explaining the learning of the model. The similarity function compares features extracted from the input to those embodied in the prototypes. Ultimately, a transparent classifier transforms the similarity scores into class predictions. The fact that the final classification revolves around the prototypes makes them a critical component of SEMs. In addition to these, other modules have also been utilized in the literature to facilitate the learning of prototypes, such as a decoder to align the embedding space to the input space (Parekh et al., 2021; Gautam et al., 2022), or a companion encoder to learn the prototypical space (Parekh et al., 2021).

2.1 Predicates for SEMs

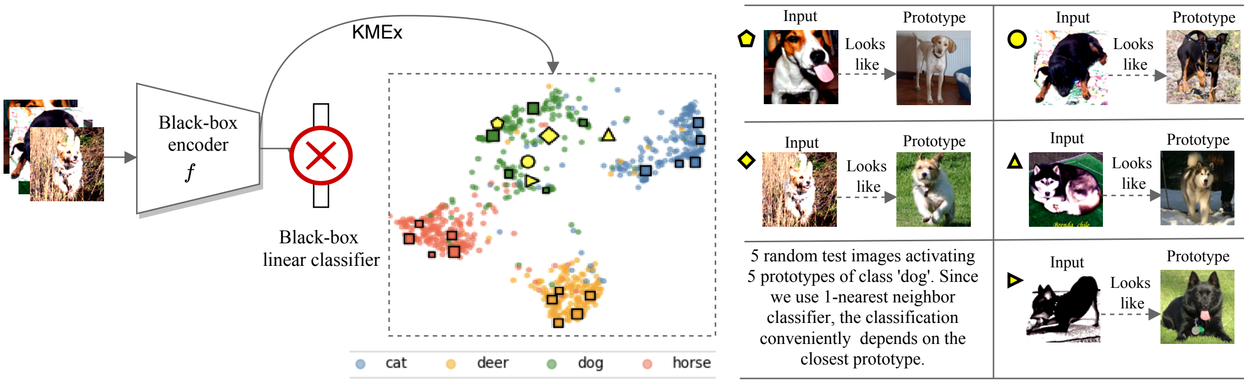
PSEMs are designed to learn inherently interpretable global class concepts. Three principles arise from the literature to form a framework for their construction: transparency, diversity, and trustworthiness (Gautam et al., 2022).

- A model is said to be *transparent* if the downstream task involves solely human-interpretable concepts and operations.
- The learned concepts are *diverse* if they capture non-overlapping information in the embedding space and, therefore, in the input space.
- *Trustworthiness* comes in several dimensions. An SEM is deemed *faithful* if its classification accuracy and explanations match its black-box counterpart. In addition, local and global explanations should be *robust* (similar inputs yield similar explanations) and truly reflect the important features of the input with respect to the downstream task.

2.2 Related work

The first general framework to compute interpretable concepts was SENN (Alvarez Melis & Jaakkola, 2018), which relies on a complex architecture and loss function to ensure interpretability. Following this, several SEMs have emerged, one of the most popular being ProtoPNet (Chen et al., 2019b). The latter introduces a learnable prototype similarity layer with a fixed number of prototypes per class. Several methods have followed to address the limitations of ProtoPNet. For example, ProtoPShare (Rymarczyk et al., 2021), ProtoTree (Nauta et al., 2021) and ProtoPool (Rymarczyk et al., 2022) proposed learning of shareable prototypes across classes, (Donnelly et al., 2021) proposed adaptive prototypes which change their spatial location based on the input image and TesNet (Wang et al., 2021) introduced a plug-in embedding space spanned by basis concepts constructed on the Grassman manifold, thereby inducing diversity among prototypes.

In parallel to ProtoPNet and its extensions, several other SEMs have been proposed. FLINT (Parekh et al., 2021) introduces an interpreter network with a learnable attribute dictionary in addition to the predictor. SITE (Wang & Wang, 2021) introduces regularizers for obtaining a transformation-equivariant SEM. ProtoVAE (Gautam et al., 2022) learns a transparent prototypical space thanks to a backbone based



5 random test images activating 5 prototypes of class 'dog'. Since we use 1-nearest neighbor classifier, the classification conveniently depends on the closest prototype.

Figure 1: Schematic representation of KMEx: The black-box classifier is removed and replaced by a nearest neighbor classifier based on prototypes learned using k-means in the embedding space. The UMAP (McInnes et al., 2018) representation is the projection of the learned embedding space for STL-10, along with prototypes, depicted by squares weighted by their respective importances $(1 - \mathcal{D}_{\text{tsp}})$. The prototypes are visualized in the input space using the closest training images.

on a variational autoencoder, thereby having the capability to reconstruct prototypical explanations using the decoder.

While all the existing SEMs have demonstrated effective generation of explanations alongside comparable accuracies, they invariably demand significant architectural modifications and integration of multiple loss functions. This often introduces several additional hyperparameters to achieve satisfactory performance. For instance, in the case of ProtoPNet, a three-step training process involves encoder training, prototypes projection for explainability, followed by last-layer training. Furthermore, as highlighted in FLINT (Parekh et al., 2021), a simultaneous introduction of all losses can lead to suboptimal optimization. Their workaround strategy involves distinct loss combinations for fixed epochs. These intricate training strategies, combined with the challenge of training large deep learning architectures, complicate the accessibility of SEMs, thereby emphasizing the demand for more resource-efficient alternatives. KMEx, a universally applicable method that necessitates no re-training, no additional loss terms for training the backbone, and minimal architectural adjustments for learning the prototypes, presents an efficient solution to this challenge. Considering our general contributions to SEMs, we use ProtoPNet, a representative approach encompassing all its extensions, as a baseline in this work. Additionally, we also consider FLINT and ProtoVAE, which cover the diversity of the SEM's literature in terms of backbones, similarity, and loss functions. A summary of these baselines is given in Table 1, along with KMEx, which is presented in the following section.

3 KMEx: a universal explainer

In this section, we introduce our resource-efficient and universal method, KMEx, which transforms a black-box model into an SEM, fulfilling all the aforementioned predicates. Note that to enhance legibility, KMEx may refer in the following to both the method and the transformed model.

3.1 KMEx

Let us consider a trained model made of an encoder and a classifier. It can be converted into a self-explainable model using the following procedure:

1. Learn prototypes for each class using k-means on the embedding of the training data.
2. The classifier returns now the class of the closest prototype using as similarity measure:

$$\mathfrak{s}(z, p_k) = \log\left((||z - p_k||^2 + 1)/(||z - p_k||^2 + \epsilon)\right).$$

The resulting model is referred to as the K-Means Explainer (KMEx) of the original model. A schematic representation of the operations is depicted in Figure 1. Note that the KMEx conversion is not a post-hoc explainability method. Although a trained encoder is re-used, the predictions are computed differently. Additionally, given the central role of the prototypes, the inherent nature of the KMEx model is now interpretable.

We further highlight that k-means is computed per class and on the embedding space, which usually has a reasonable number of dimensions (512 for ResNet34). Hence, the computational cost is limited and manageable by classic implementations, irrespective of the complexity of the data. For very large datasets, it can be approximated by computing k-means on a subset of the training set or by using other efficient implementations (Johnson et al., 2019).

3.2 KMEx is an SEM

Visualisation of explanations The explanations for a PSEM are two-fold. *Global* explanations involve visualizations of prototypes in the input space, providing insights into the model's acquired knowledge. *Local* explanations, on the other hand, entail pixel-level explanations for input images, revealing which portions of an image are activated by each prototype. For KMEx, we provide global explanations by visualizing the training images that are closest to the corresponding prototypes in the embedding space. This approximation is justified by the problem solved by k-means, which makes it unlikely for a prototype to be out of distribution. For local explanations, we adhere to previous works and employ Prototypical Relevance Propagation (PRP), a technique demonstrated to be efficient and accurate for ProtoPNet (Gautam et al., 2023).

Transparency The nearest prototype classifier of KMEx allows backtracking of the influence of a prototype on the predictions, which relates to a distance in the embedding space, thus embodying transparency.

Trustworthiness of the predictions If the original trained model learned to separate well the classes in the embedding, there should be enough inter-class distance for the linear partition of k-means to yield KMEx prototypes that also correctly separate the classes and thus achieve classification performance akin to that of the trained model.

Trustworthiness of the explanations The only difference between a black-box and its KMEx is how the predictions are derived from the embedding. Therefore, considering identical weights in both models' encoders, most of the operations involved in the generation of local explanation maps are common to both, thus similar explanations are expected, regardless of the technique chosen to generate local explanations.

Diversity The purpose of the prototypes is to serve as representatives of their neighborhood in the embedding space. The diversity predicate implicitly requires that they also spread over the embedding. To satisfy this predicate without compromising their function, we aim to position the prototypes on the accumulation points of the embedding. These are captured as the modes of a Gaussian density estimate. Computing such a model for a high dimensional and sparse dataset is costly, hence we approximate it using k-means. Finally, given that k-means employs a uniform prior on the cluster probabilities, this method has the advantage of covering as much of the data in the embedding space as possible, thus fostering diversity.

4 Evaluations

As stated earlier, existing SEMs build upon three shared predicates but adopt varied strategies to ensure their fulfillment. *Transparency* is assumed based on architectural choices and, at best, confirmed through visualization of prototypes using different strategies, such as upsampling (Chen et al., 2019b), activation maximization (Parekh et al., 2021; Mahendran & Vedaldi, 2016) and PRP (Gautam et al., 2023), accompanied with similarity scores. The *trustworthiness* predicate is the most quantifiable one. The faithfulness of the performance with respect to the "closest" black-box is often reduced to a comparison of accuracies, and the robustness of the explanations is evaluated via recent measures such as Average Increase (AI), Average Drop (AD), and Relevance Ordering (RO) test (Lee et al., 2021; MacDonald et al., 2019; Hedström et al., 2022). Nonetheless, the quantification of disparities between local explanations generated by an SEM and its nearest black-box model has been largely disregarded. We emphasize that this aspect grows in significance, particularly as we transition to techniques that transform existing black-box models into interpretable ones without re-training, a domain where KMEx stands as the first approach. Finally, prototypical *diversity*

Table 2: Evaluation strategies for the predicates used by state-of-the-art SEMs. Proposed evaluation framework is *italicised.*

	Transparency	**Trustworthiness**			**Diversity**
		Baseline	**Faithfulness**	**Robustness**	
ProtoPNet	Visualization	Black-box	Accuracy	-	-
FLINT	Visualization	Black-box	Accuracy	-	-
ProtoVAE	Visualization	Black-box/SEM	Accuracy	AI/AD/RO	Reconstruction visualization
Proposed Evaluation	*Ghosting*	Black-box/SEM/*KMEx*	Accuracy/*KL Divergence*	AI/AD/RO	*Inter-prototype similarity*

Table 3: Prediction accuracy for SEMs demonstrating the effectiveness of KMEx as an SEM baseline. Reported numbers are averages over 5 runs along with standard deviations.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
ResNet34	$99.4^{\pm0.0}$	$92.4^{\pm0.1}$	$92.6^{\pm0.2}$	$85.6^{\pm0.1}$	$91.8^{\pm0.1}$	$86.5^{\pm0.1}$	$98.5^{\pm0.0}$
R34+KMEx	$99.4^{\pm0.0}$	$92.3^{\pm0.1}$	$92.4^{\pm0.1}$	$85.3^{\pm0.1}$	$91.9^{\pm0.2}$	$86.6^{\pm0.2}$	$98.3^{\pm0.0}$
FLINT	$99.2^{\pm0.1}$	$91.8^{\pm0.5}$	$91.1^{\pm0.7}$	$82.2^{\pm1.1}$	$87.5^{\pm0.6}$	$87.3^{\pm0.2}$	$97.2^{\pm0.3}$
ProtoPNet	$99.4^{\pm0.1}$	$92.4^{\pm0.2}$	$94.4^{\pm0.1}$	$84.9^{\pm0.2}$	$88.1^{\pm0.6}$	$87.8^{\pm0.2}$	$98.1^{\pm0.0}$
ProtoVAE	$99.4^{\pm0.0}$	$92.7^{\pm0.5}$	$93.8^{\pm0.6}$	$83.0^{\pm0.2}$	$85.6^{\pm1.1}$	$85.1^{\pm0.8}$	$98.6^{\pm0.0}$

has been largely overlooked in prior research, with evaluations, if conducted, being primarily qualitative in nature (Gautam et al., 2022).

In this section, we first evaluate KMEx following the evaluation protocols used in the original papers of the selected baselines, which are summarized in Table 2. Following this, we propose our full quantitative evaluation framework based on the predicates for SEMs, highlighting the gaps in the evaluation of SEMs existing until now, also summarized in Table 2. Additionally, we present a quantitative study of the diversity and subclass representation captured by the prototypes learned by existing SEMs and their KMEx counterparts.

Datasets, implementation and baselines We evaluate all methods on 7 datasets, MNIST (Lecun et al., 1998), FashionMNIST (Xiao et al., 2017) (fMNIST), SVHN (Netzer et al., 2011), CIFAR-10 (Krizhevsky, 2009), STL-10 (Coates et al., 2011), a subset of QuickDraw (Parekh et al., 2021) and binary classification for male and female for the CelebA dataset (Liu et al., 2015). We use a vanilla ResNet34 (He et al., 2016) as the encoder for all the models and fix the number of prototypes per class as 20 for CelebA and 5 for all other datasets. Further implementation details are provided in Appendix A.2. For baselines, we train ProtoPNet (Chen et al., 2019b), FLINT (Parekh et al., 2021), and ProtoVAE (Gautam et al., 2022) for learning image-level prototypes. For ProtoPNet, we use average pooling to generate image-level prototypes. For FLINT, we use the interpreter network FLINT-g.

4.1 Traditional evaluation of KMEx

In this section, we evaluate KMEx following previous lines of works (Wang & Wang, 2021; Gautam et al., 2022). We start with comparing the predictive performance of KMEx, which is then followed by an evaluation of explanations consisting of visualization of prototypes and evaluating the robustness of explanations.

4.1.1 Predictive performance

We report the accuracy achieved by KMEx, as well as selected baselines in Table 3. As can be observed, KMEx performs on par with its corresponding ResNet34 black-box base model, thereby validating the change of classifier. On the other hand, other SEMs, i.e., FLINT, ProtoPNet, and ProtoVAE, suffer some loss of accuracy for some datasets when compared to the black-box.

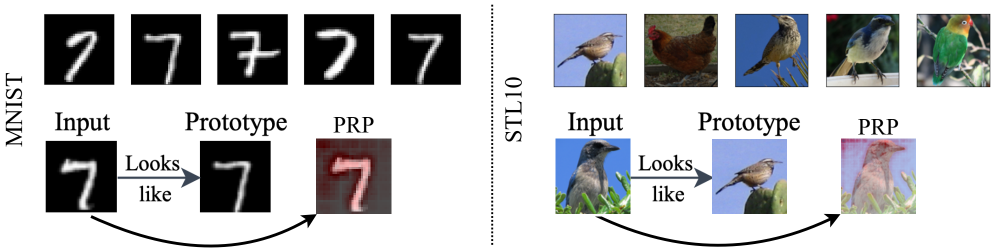


Figure 2: Qualitative evaluation of KMEx: Prototypes learned by KMEx for MNIST for class ‘7’ (left) and STL-10 for class ‘bird’ (right) are shown at the top, demonstrating global explainability. *This* looks like *that* behavior for test images are shown at the bottom, along with PRP maps demonstrating the regions activated by closest prototypes for the test images, exhibiting local explainability.

4.1.2 Evaluation of explanations

In previous works, the evaluation of explanations is in two folds: 1) Qualitative evaluation of prototypes and 2) Evaluation of robustness of prototypical explanations. For qualitative evaluation, we visualize prototypes learned by KMEx for MNIST and STL-10 datasets in Figure 2 (top row). We further demonstrate the “*this* looks like *that*" behavior exhibited by KMEx for test images in the bottom row, along with their corresponding PRP maps, demonstrating the regions activated in the test images by their closest prototypes. As observed, for the MNIST dataset, the activations are in response to the shape of the digit in the prototype. Similarly, for STL-10, the closest prototype has emphasized key features of a bird, such as the head, beak, and eyes, as well as a portion of the sky in the background.

We evaluate the robustness of the local explanations using the AD, AI of the similarity scores, as well as RO test (Lee et al., 2021; Gautam et al., 2023). AD estimates the average decrease in similarity scores with respect to each prototype when the 50% least important pixels are set to zero for black and white images and to random noise for colored datasets, respectively. AI corresponds to the frequency with which the similarly disturbed input increases the similarity. A low AD and a high AI suggest robustness. We report in Table 4 average AI and AD scores and standard deviation over 1000 test images and five runs. The RO test (MacDonald et al., 2019; Gautam et al., 2022), where the most important pixels from the PRP maps are added gradually in an image to measure the change in predicted class probability. The RO curves are shown in Figure 3 for MNIST, CIFAR-10, and CelebA, along with the respective random baselines (MacDonald et al., 2019). A larger area-under-the-curve suggests more robustness. The curves are computed as mean for 1000 test images selected at random, averaged over 5 runs. We employ PRP maps for generating local explanation maps for all baselines and tests, thus ensuring equitable and consistent analysis across all SEMs. FLINT is excluded here because of the lack of clear PRP rules for such an architecture.

First, except for CelebA, none of the results of Table 4 are statistically different. Overall, ProtoPNet returns, on average, the lowest AD scores (except for STL-10). As for AI scores, the highest averages alternate between ProtoPNet and ProtoVAE. Although KMEx’s AD scores remain worse than that of the black-box ResNet34, the AI scores of both models are overall very similar, with an average difference of about 0.030%. This behavior is also visible in Figure 3, where the curve of ResNet34+KMEx (orange) stays very close to the black-box baseline (blue), while other SEMs show different behaviors. These results suggest that KMEx does not produce more robust explanations on its own. This is anticipated as KMEx aims to facilitate the interpretation of a learned latent representation but does not enforce robustness or stability of the prototypes during the training process, unlike other SEMs.

4.2 Quantitative evaluation of SEMs

Having demonstrated the traditional evaluation of the proposed KMEx, we now address the lack of a comprehensive evaluation framework for SEMs that quantitatively evaluates the predicates. First, we expose for the first time how transparency is often undermined by unused prototypes (ghosting) and measure the phenomenon. Next, we objectively quantify the faithfulness of local explanations and the diversity of the

Table 4: AI and AD scores for robustness of explanations.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
AD similarity (lower is better)							
ResNet34	$0.076^{\pm 0.142}$	$0.121^{\pm 0.155}$	$0.136^{\pm 0.158}$	$0.130^{\pm 0.153}$	$0.096^{\pm 0.129}$	$0.085^{\pm 0.134}$	$0.399^{\pm 0.155}$
R34+KMEx	$0.121^{\pm 0.234}$	$0.156^{\pm 0.207}$	$0.178^{\pm 0.213}$	$0.167^{\pm 0.189}$	$0.117^{\pm 0.172}$	$0.150^{\pm 0.201}$	$0.662^{\pm 0.158}$
ProtoPNet	$0.045^{\pm 0.113}$	$0.095^{\pm 0.138}$	$0.025^{\pm 0.083}$	$0.010^{\pm 0.038}$	$0.125^{\pm 0.276}$	$0.036^{\pm 0.114}$	$0.099^{\pm 0.101}$
ProtoVAE	$0.157^{\pm 0.352}$	$0.146^{\pm 0.297}$	$0.083^{\pm 0.232}$	$0.102^{\pm 0.221}$	$0.051^{\pm 0.138}$	$0.054^{\pm 0.157}$	$0.573^{\pm 0.454}$
AI similarity (higher is better)							
ResNet34	$0.650^{\pm 0.434}$	$0.466^{\pm 0.492}$	$0.416^{\pm 0.474}$	$0.386^{\pm 0.480}$	$0.478^{\pm 0.4946}$	$0.592^{\pm 0.426}$	$0.032^{\pm 0.168}$
R34+KMEx	$0.638^{\pm 0.420}$	$0.461^{\pm 0.495}$	$0.434^{\pm 0.474}$	$0.348^{\pm 0.469}$	$0.509^{\pm 0.4957}$	$0.514^{\pm 0.418}$	$0.003^{\pm 0.026}$
ProtoPNet	$0.765^{\pm 0.422}$	$0.530^{\pm 0.480}$	$0.834^{\pm 0.367}$	$0.889^{\pm 0.312}$	$0.742^{\pm 0.4309}$	$0.834^{\pm 0.371}$	$0.363^{\pm 0.479}$
ProtoVAE	$0.774^{\pm 0.417}$	$0.684^{\pm 0.461}$	$0.761^{\pm 0.422}$	$0.710^{\pm 0.450}$	$0.669^{\pm 0.4690}$	$0.851^{\pm 0.343}$	$0.375^{\pm 0.484}$

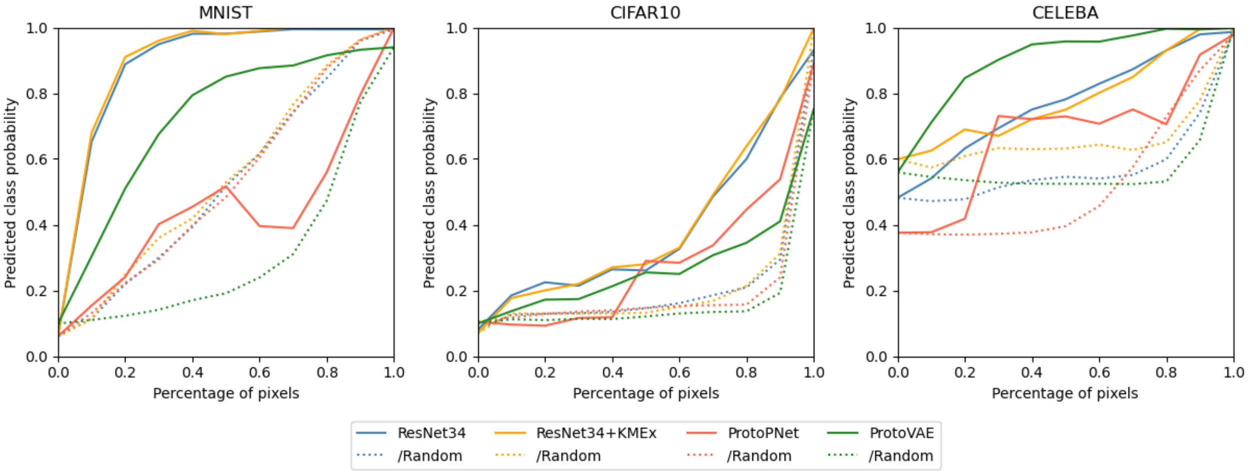


Figure 3: Relevance Ordering curves computed on different datasets and with different architectures, along with the respective random baselines (dashed).

prototypes without resorting to visual inspection. Again, the rationale is to propose a framework to assess objectively each model's strengths and weaknesses.

4.2.1 Notations

Let us consider an image dataset $\mathcal{X} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ made of $N > 0$ images split into $C > 0$ classes, where $\boldsymbol{x}_i \in \mathbb{R}^{W \times H \times J}$ is an image of width $W > 0$, height $H > 0$, and with $J > 0$ channels, and $\boldsymbol{y}_i \in [1 \dots C]$ encodes its label. We consider a set of $K > 0$ prototypes $\{p_1 \dots p_K\} \subset \mathbb{R}^D$ that are vectors of the embedding space $\mathbb{R}^D$. Any model in the following contains both an encoder f such that $z_i = f(x_i) \in \mathbb{R}^D$ and a similarity measure $\mathfrak{s}$ between vectors of the embedding space that returns larger values to pairs of vectors deemed similar.

4.2.2 Transparency and concept ghosting

The transparency predicate allows the user to backtrack the influence of the learned concepts on the predictions and is usually enforced through architecture design. However, we observe for state-of-the-art SEMs that, in practice, some learned prototypes are not reachable from the predictions. More specifically, they are never *activated* by any training data point of their class, i.e., they are never the most similar prototype of any training data. This so-called *ghosting* of the prototypes not only gives a false sense of needed concepts but

Table 5: *Transparency*: frequency of ghosted prototypes by SEMs.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
ProtoPNet	$0.580^{\pm 0.060}$	$0.528^{\pm 0.027}$	$0.300^{\pm 0.076}$	$0.164^{\pm 0.056}$	$0.232^{\pm 0.083}$	$0.156^{\pm 0.050}$	$0.670^{\pm 0.141}$
FLINT	$0.160^{\pm 0.101}$	$\mathbf{0.188^{\pm 0.254}}$	$0.060^{\pm 0.025}$	$0.112^{\pm 0.076}$	$0.240^{\pm 0.077}$	$\mathbf{0.228^{\pm 0.409}}$	$\mathbf{0.215^{\pm 0.411}}$
ProtoVAE	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.004^{\pm 0.009}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$0.760^{\pm 0.037}$	$0.552^{\pm 0.070}$	$0.155^{\pm 0.060}$
KMEx	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$	$\mathbf{0.0^{\pm 0.0}}$

also undermines the notion of transparency itself, as the link between prototypes and prediction can not be fully trusted.

In the case of a distance-based similarity measure (Gautam et al., 2022; Chen et al., 2019b), if $k = \text{argmax}_l\, \mathfrak{s}(z_i, p_l)$, then prototype p_k is the closest to z_i. If several points activate p_k, this indicates that the data embedding aggregates around a concentration point close to p_k. In contrast, if p_k never maximizes any $\mathfrak{s}(z_i, \cdot)$, then there might be no data in its neighborhood. The prototype is either out-of-distribution or lies in an area of low density. In the case of a dot-product-based similarity measure (Parekh et al., 2021), if p_k is activated by z_i, then z_i and p_k are aligned. Assuming that z_i is not the only maximizer of $\mathfrak{s}(\cdot, p_k)$, then p_k carries a direction along which the data accumulates. In contrast, if p_k never maximizes any $\mathfrak{s}(z_i, \cdot)$, then the data does spread along its direction and may represent either a variation within a class or, in the worst case, noise.

We propose to quantify this *ghosting* phenomenon based on the average activation frequencies over the prototypes on the training set:

$$\mathcal{D}_{\text{tsp}} = 1 - \sum_{k=1}^{K} \frac{\#\{i, \text{argmax}_{1 \leq l \leq K} \Big(\mathfrak{s}(z_i, p_l)\Big) = k\}}{KN}, \tag{1}$$

where # stands for the cardinal of the set. The values of $\mathcal{D}_{\text{tsp}}$ range between 0 and 1, with lower values indicating less ghosting.

In Table 5, we report average $\mathcal{D}_{\text{tsp}}$ scores (Equation 1) with standard deviation over five runs. We observe that ghosting affects all models but, unsurprisingly, not KMEx. Indeed, for such a low number of prototypes, relative to the size of the data, k-means is unlikely to create an empty cluster. Interestingly, ProtoVAE almost never ghosts any prototype on four out of the seven datasets, suggesting that SEMs with geometrical constraints are more robust to ghosting.

4.2.3 Trustworthiness and faithfulness

According to its definition, the trustworthiness predicate encompasses two major axes. The first is the *faithfulness of the predictions*, which we have quantified in terms of accuracies. The second aspect concerns the *robustness of the explanations*, which we measured using AI, AD, and RO (Lee et al., 2021; Gautam et al., 2022). An often overlooked aspect of the trustworthiness predicate is the *faithfulness of the explanations*. Indeed, SEMs differ from black-box models in their architecture and training and, therefore, also in their local explanations. However, as we move towards methods that convert black-box models into self-explainable, it becomes crucial to quantitatively evaluate this discrepancy. We propose to use the Kullback-Leibler (KL) divergence (D_{KL}) between the Layer-wise Relevance Propagation (LRP) maps (Bach et al., 2015) for the prediction probabilities produced by the SEM and the black-box baseline. Since the divergence acts on distributions, the relevance maps need to be normalized. The use of LRP aligns with the previous utilization of PRP. Other methods could be used, yet our intention here is not to evaluate the SEMs with respect to these methods but rather with respect to the predicates.

Let us denote the output of the local explanation method for an input x as $e(x) \in \mathbb{R}^{W \times H \times C}$. The corresponding normalized relevance $e_n(x) \in \mathbb{R}^{W \times H}$ is defined as:

$$e_n(x) = \frac{\max_{j=1 \ldots J} |e(x)|(\cdot, \cdot, j)}{\sum_{w=1 \ldots W} \sum_{h=1 \ldots H} \max_{j=1 \ldots J} |e(x)|(w, h, j)} \tag{2}$$

Table 6: *Faithfulness of explanations*: divergence of LRP explanation maps from the black-box.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
ProtoPNet	$0.438^{\pm 0.173}$	$\mathbf{0.279^{\pm 0.182}}$	$\mathbf{0.316^{\pm 0.183}}$	$\mathbf{0.225^{\pm 0.146}}$	$\mathbf{0.455^{\pm 0.168}}$	$0.689^{\pm 0.441}$	$\mathbf{0.800^{\pm 0.318}}$
ProtoVAE	$0.829^{\pm 0.239}$	$0.678^{\pm 0.221}$	$0.838^{\pm 0.033}$	$1.106^{\pm 0.420}$	$\mathbf{0.361^{\pm 0.086}}$	$0.953^{\pm 0.649}$	$\mathbf{0.828^{\pm 0.721}}$
KMEx	$\mathbf{0.086^{\pm 0.141}}$	$\mathbf{0.169^{\pm 0.163}}$	$\mathbf{0.199^{\pm 0.159}}$	$\mathbf{0.159^{\pm 0.163}}$	$\mathbf{0.354^{\pm 0.109}}$	$\mathbf{0.082^{\pm 0.141}}$	$\mathbf{0.554^{\pm 0.219}}$

The divergence of $e_n(x)$ with respect to the normalized local explanation maps produced by the black-box backbone $e_n^{\text{bbox}}(x)$ is measured by $\mathcal{D}_{\text{fdl}}$ defined as follows:

$$\mathcal{D}_{\text{fdl}} = \sum_{w=1}^{W} \sum_{h=1}^{H} e_n(x)(w,h) \log \left(\frac{e_n(x)(w,h)}{e_n^{\text{bbox}}(x)(w,h)} \right) \tag{3}$$

The KL divergence is zero if, and only if, the distributions are equal. Consequently, $\mathcal{D}_{\text{fdl}}$ can be null if, and only if, the SEM and the black-box models always produce the same explanations.

In Table 6, we report the average $\mathcal{D}_{\text{fdl}}$ based on LRP and standard deviation over five runs for each SEM on 1000 images of each dataset. FLINT is excluded here because of the lack of clear LRP rules for such an architecture. Despite the quite large standard deviations, KMEx produces the most faithful feature importance maps. This is expected since most of the operations happen in the encoder, which originates from the black-box model. It is closely followed by ProtoPNet. On the other hand, ProtoVAE, which has the most different architecture, also yields the most different local explanations.

4.2.4 Interpreted diversity

The abundance of existing strategies to guarantee the *diversity* of an SEM reflects the subjectivity of the notion. Thus, it is not obvious how to evaluate this predicate in the input space, especially given that very few public image datasets provide attributes describing the image. Thus, the evaluation has to be done in the embedding space. However, using a metric based on distances in the embedding space would disadvantage methods relying on a dot-product-based similarity measure and the other way around. We therefore propose to evaluate SEMs on the basis of their own interpretation of diversity and to base our *diversity* metric on the models' own similarity function. In other words, the idea is to assess the extent to which models achieve diversity on the basis of their own model choices.

The overarching objective of existing approaches for diversity is to prevent prototype collapse. In such a case, the information captured by the prototypes highly overlaps, yielding inter-prototype similarities ($\mathfrak{s}(p_k, p_l)$ with $k \neq l$) as high as prototypical self-similarities ($\mathfrak{s}(p_k, p_k)$). On the other hand, if prototype collapse is well alleviated, the inter-prototype similarities are low, while the self-similarities remain high. This observation motivates the use of the entropy function.

Accordingly, we quantify the diversity of a set of class concepts using $\mathcal{D}_{\text{dvs}}$ defined as the class average of the normalized entropy of the similarities between each prototype of the class. The computation is done per class and without discarding the ghosted prototypes, as they may indicate a collapse.

$$\mathcal{D}_{\text{dvs}} = \frac{1}{K} \sum_{k=1}^{K} \frac{\mathbf{H}\left(\text{Softmax}\left(\mathfrak{s}(p_k, p_\cdot)\right)\right)}{\log(K)}, \tag{4}$$

where $\mathbf{H}$ is the entropy function. The $\log(K)$-normalization restricts the measure to $[0,1]$ and allows comparisons between different runs, number of prototypes, as well as models. Large values indicate more similarity between the clusters and, thereby, less diversity.

In Table 7, we report the average $\mathcal{D}_{\text{dvs}}$ (Equation 4) score with standard deviations over five runs for each SEM on each dataset. Recall that $\mathcal{D}_{\text{dvs}}$ estimates how well a model satisfies its own interpretation of diversity. Following this, ProtoVAE and FLINT are, respectively, the most and the least satisfying models. We emphasize here that a low diversity doesn't reflect the caliber of the learned embedding space and

Table 7: *Interpreted diversity*: quantitative evaluation of diversity for different SEMs.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
ProtoPNet	$0.691^{\pm0.067}$	$0.717^{\pm0.020}$	$0.708^{\pm0.031}$	$0.768^{\pm0.081}$	$0.548^{\pm0.052}$	$0.687^{\pm0.102}$	$0.741^{\pm0.158}$
FLINT	$0.941^{\pm0.010}$	$0.943^{\pm0.013}$	$0.911^{\pm0.025}$	$0.930^{\pm0.020}$	$0.989^{\pm0.002}$	$0.901^{\pm0.047}$	$0.918^{\pm0.023}$
ProtoVAE	$\mathbf{0.367}^{\pm\mathbf{0.050}}$	$\mathbf{0.301}^{\pm\mathbf{0.044}}$	$\mathbf{0.349}^{\pm\mathbf{0.101}}$	$\mathbf{0.186}^{\pm\mathbf{0.012}}$	$\mathbf{0.215}^{\pm\mathbf{0.042}}$	$\mathbf{0.147}^{\pm\mathbf{0.025}}$	$\mathbf{0.445}^{\pm\mathbf{0.081}}$
KMEx	$0.453^{\pm0.067}$	$0.402^{\pm0.083}$	$\mathbf{0.389}^{\pm\mathbf{0.087}}$	$0.399^{\pm0.058}$	$0.373^{\pm0.071}$	$0.374^{\pm0.085}$	$\mathbf{0.443}^{\pm\mathbf{0.068}}$

only suggests an important overlap of information between the representative prototypes learned for the embeddings.

4.2.5 Summary

In Figure 4, we summarize the results of Tables 3, 4 and 5 to 7 using an average radar plot for each model. Axes are inverted when necessary such that a larger polygonal area is better. This visualization makes it easy to identify the strengths and weaknesses of each SEM and thus determine the most suitable model according to the problem statement at hand.

KMEx suffers the least of ghosting (good transparency) and is the most faithful model with respect to the original black-box both in terms of accuracy and explanation. ProtoPNet performs well in terms of predictions and robustness of the local explanations, but it underperforms in terms of diversity. This is due to the lack of an inter-class diversity constraint in the ProtoPNet's design. On the other hand, ProtoVAE leads in terms of diversity, but its explanations resemble the black-box base model explanations the least. This is due to the utilization of a VAE backbone, which deviates a lot from the architecture of the black-box baseline. FLINT, for which local explanations could not be evaluated, is satisfactory in terms of ghosting and fidelity of its predictions. On the other hand, despite having an entropy constraint for promoting diversity of the attributes, it obtains the worst results in terms of measured diversity.

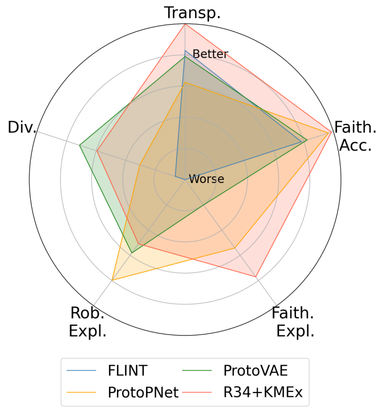


Figure 4: Summary of each model's strengths and weaknesses.

4.3 Diversity and embedding

In this section, we show that for the same embedding learned by an SEM, the KMEx paradigm for prototypes may also be used to improve both the measured and qualitative diversity without retraining or altering the embedding.

KMEx improves measured diversity We evaluate first how changing the paradigm of an SEM to KMEx may improve the quantified diversity ($\mathcal{D}_{\text{dvs}}$). We report in Table 8 average scores and standard deviations over five runs for the KMEx of each SEM baseline and the average difference with Table 7. We observe that KMEx almost always improves $\mathcal{D}_{\text{dvs}}$ scores (negative Diff.). The most significant gain is for FLINT, which, after transformation, returns the lowest scores for several data sets. Recall that $\mathcal{D}_{\text{dvs}}$ can only serve as an internal evaluation, therefore any further analysis of the prototypes requires an external criterion.

KMEx improves minority subclass representation We further study here the diversity of the prototypes in light of the representation of the attributes they capture. We interpret the notion of a fair subclass representation for SEMs as whether prototypes are able to capture the information about the underrepresented subclasses. For this experiment, we trained ResNet34+KMEx, ProtoPNet, ProtoVAE, FLINT, and their KMEx on the CelebA dataset for male and female classification with varying numbers of prototypes. Prototypes are represented in the input space by their nearest training images, which come with 40 binary attributes as annotations.

Table 8: Diversity with KMEx: Quantitative evaluation of diversity by applying KMEx to learned SEM embeddings.

	MNIST	fMNIST	SVHN	CIFAR-10	STL-10	QuickDraw	CelebA
ProtoPNet+KMEx	$0.698^{\pm 0.073}$	$0.649^{\pm 0.024}$	$0.640^{\pm 0.040}$	$0.688^{\pm 0.030}$	$0.401^{\pm 0.018}$	$0.641^{\pm 0.068}$	$0.742^{\pm 0.062}$
Diff.	0.007	-0.068	-0.068	-0.080	-0.147	-0.046	0.001
FLINT+KMEx	$0.205^{\pm 0.039}$	$0.161^{\pm 0.014}$	$0.109^{\pm 0.015}$	$0.138^{\pm 0.029}$	$0.313^{\pm 0.077}$	$0.180^{\pm 0.051}$	$0.228^{\pm 0.019}$
Diff.	-0.737	-0.782	-0.803	-0.793	-0.675	-0.721	-0.691
ProtoVAE+KMEx	$0.331^{\pm 0.052}$	$0.244^{\pm 0.069}$	$0.208^{\pm 0.072}$	$0.127^{\pm 0.017}$	$0.520^{\pm 0.067}$	$0.067^{\pm 0.007}$	$0.434^{\pm 0.075}$
Diff.	-0.037	-0.057	-0.141	-0.059	0.305	-0.080	-0.012

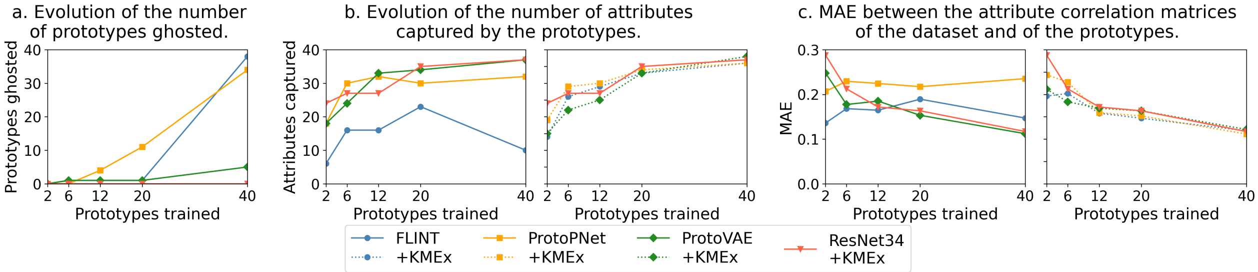


Figure 5: Analysis of the attributes captured by SEMs for different numbers of prototypes for CelebA.

To put the observations in the other figures into perspective., we plot first in Figure 5.a the number of prototypes ghosted against the number of prototypes trained. We see here clearly the depth of the issue for FLINT and ProtoPNet. The two following plots (Figure 5.b) depict the number of captured attributes given a number of trained prototypes, including ghosted ones. The left plot shows the results for the baselines and the right one for their KMEx. FLINT starts and ends with fewer captured attributes, and it seems unstable with a large number of prototypes. As for ProtoPNet, it caps at 32 attributes when trained with 12 or more prototypes. On the other hand, the KMEx of any method (right plot), including ResNet34+KMEx (red), always captures more attributes as the number of prototypes increases. The last experiment aims to evaluate how many combinations of attributes are captured using the mean absolute error (MAE) between the attribute correlation matrices computed from the training set and the prototypes. (Figure 5.c). The correlations based on ProtoPNet's prototypes are the most divergent, whereas ProtoVAE and ResNet34+KMEx consistently come closer to the ground truth as the number of prototypes increases. Again, the attribute correlation computed for any KMEx consistently improves as more prototypes are available.

Overall, KMEx of FLINT improves the most its original model in both criteria: the number of captured attributes and the faithfulness of the attributes correlations. This observation reinforces the intuition that FLINT learns an embedding with much more potential in terms of global explanations than it is able to leverage through the prototypes it learns.

5 Conclusion

In this paper, we introduce KMEx, the first approach for making any black-box model self-explainable. KMEx is a universally applicable, simple, and resource-efficient method that, unlike existing methodologies, does not require re-training of the black-box model. Furthermore, we reconsider the subjective evaluation practices for SEMs by introducing a quantitative evaluation framework that facilitates objective comparisons among SEM approaches. The proposed framework adopts a set of novel metrics to quantify how well SEMs adhere to the established predicates. An extensive evaluation with the help of the proposed framework highlights the strengths and weaknesses of existing SEM approaches when compared to the models obtained from KMEx. This work, therefore, additionally serves as a foundational step towards an objective, comprehensive, and resource-efficient advancement of the SEM field.

One notable limitation of the proposed KMEx is its reliance on selecting a priori the number of prototypes, a characteristic it shares with current state-of-the-art SEMs (Parekh et al., 2021; Gautam et al., 2022; Chen et al., 2019b). Additionally, note that the proposed detailed quantitative evaluation framework is meant to provide an additional perspective and not replace qualitative evaluations of SEMs, which are still required due to the subjective nature of explanations.

References

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

SeungHwan An, Hosik Choi, and Jong-June Jeon. Exon: Explainable encoder network. arXiv, 2021.

Christopher J. Anders, Talmaj Marinc, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Analyzing imagenet with spectral relevance analysis: Towards imagenet un-hans'ed. *ArXiv*, abs/1912.11425, 2019.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and Survey of Explanation Methods for Black Box Models. *Data Min Knowl Disc*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Alexander Camuto, Matthew Willetts, Stephen Roberts, Chris Holmes, and Tom Rainforth. Towards a theoretical understanding of the robustness of variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 3565–3573. PMLR, 2021.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *SafeAI@AAAI*, 2019a.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019b.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Alex J DeGrave, Joseph D Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021. ISSN 2522-5839.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Jonathan Donnelly, Alina Jade Barnett, and Chaofan Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10255–10265, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Xavier Ferrer, Tom van Nuenen, Jose M. Such, Mark Coté, and Natalia Criado. Bias and discrimination in ai: A cross-disciplinary perspective. *IEEE Technology and Society Magazine*, 40(2):72–80, 2021.

Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *ArXiv*, abs/2007.10760, 2020.

Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in Neural Information Processing Systems*, 35:17940–17952, 2022a.

Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2022b.

Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, 2023.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

Anna Hedström, Leander Weber, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M.-C. Höhne. Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations. 2022.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

Younghyun Jo, Sejong Yang, and Seon Joo Kim. Srflow-da: Super-resolution using normalizing flow with deep convolutional block. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 364–372, 2021.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. Xprotonet: Diagnosis in chest radiography with global and local explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15719–15728, June 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

M. Kohlbrenner, A. Bauer, S. Nakajima, A. Binder, W. Samek, and S. Lapuschkin. Towards best practice in explaining neural network decisions with lrp. In *IJCNN*, pp. 1–7, 2020.

Zhifeng Kong and Kamalika Chaudhuri. Understanding instance-based interpretability of variational autoencoders. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 2400–2412. Curran Associates, Inc., 2021.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998a.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998b.

Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. Relevance-cam: Your model already knows where to look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14944–14953, June 2021.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Jan MacDonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. A rate-distortion framework for explaining neural network decisions. *ArXiv*, abs/1905.11092, 2019.

Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120:233–255, 2016.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

Gaurav Menghani. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Comput. Surv.*, 55(12), mar 2023. ISSN 0360-0300.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65: 211–222, 2017.

Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14933–14943, June 2021.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Sajad Norouzi, David J Fleet, and Mohammad Norouzi. Exemplar vae: Linking generative models, nearest neighbor retrieval, and data augmentation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8753–8764. Curran Associates, Inc., 2020.

Dario Augusto Borges Oliveira and Laura Elena Cue La Rosa. Prototypical variational autoencoders, 2022.

Daniel Omeiza, Helena Webb, Marina Jirotka, and Lars Kunze. Explanations in autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–21, 2021.

Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché Buc. A framework to learn with interpretation. *Advances in Neural Information Processing Systems*, 34:24273–24285, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* Curran Associates Inc., Red Hook, NY, USA, 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. KDD '16, pp. 1135–1144, 2016.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, pp. 1420–1430, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325.

Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. Interpretable image classification with differentiable prototypes assignment. In *European Conference on Computer Vision*, pp. 351–368. Springer, 2022.

Sascha Saralajew, Lars Holdijk, Maike Rees, Ebubekir Asan, and Thomas Villmann. Classification-by-components: Probabilistic modeling of reasoning over a set of components. In *NeurIPS*, 2019.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

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

Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. Interpretable image recognition by constructing transparent embedding space. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 875–884, 2021.

Yipei Wang and Xiaoqian Wang. Self-interpretable model with transformation equivariant interpretation. *Advances in Neural Information Processing Systems*, 34:2359–2372, 2021.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Jason Yosinski, Jeff Clune, Anh M Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *ICML Deep Learning Workshop*, 2015.

A Appendix

In this section, we provide additional details for the proposed evaluation framework. First, we provide additional dataset and implementation details. We then provide qualitative results for the *faithfulness* of explanations for completeness. Further, we provide preliminary results on a more complex dataset CUB200 (Welinder et al., 2010), followed by additional qualitative results for KMEx.

A.1 Dataset details

All datasets used in this work are open-source. For all datasets, we use the official training and testing splits, except for QuickDraw (Ha & Eck, 2018) for which we use a subset of 10 classes that was created by (Parekh et al., 2021). This subset consists of the following 10 classes: *Ant, Apple, Banana, Carrot, Cat, Cow, Dog, Frog, Grapes, Lion.* Each of the classes contains 1000 images of size 28×28 out of which 80% are used for training and the remaining 20% for testing. The MNIST (Lecun et al., 1998), fMNIST (Xiao et al., 2017), CIFAR-10 (Krizhevsky, 2009) datasets consist of 60,000 training images and 10,000 test images of size 28×28, 28×28 and 32×32, respectively. The MNIST, fMNIST and QuickDraw images are resized to 32×32 to obtain a consistent latent feature size. SVHN (Netzer et al., 2011) consists of 73,257 training images and 26,032 images for testing of size 32×32. STL-10 (Coates et al., 2011) consists of 5000 images for training and 8000 for testing of size 96×96. All datasets have 10 classes, except for CelebA (Liu et al., 2015) for which we perform binary classification of male vs female. Number of training and testing images for CelebA are 162,770 and 19,962, respectively, of size 224×224. The licenses for the datasets are provided in Table 9.

For preprocessing, every dataset's respective mean and standard deviation for training data is used for normalization. For MNIST, fMNIST, SVHN and QuickDraw, no augmentations were performed. For STL-10, CIFAR-10 and CelebA, we apply a horizontal flip with a probability of 0.5 followed by random cropping after zero-padding with size 2 was applied for augmentation.

Table 9: Licenses for datasets used in this work. N-C is used to denote that the data is free for non-commercial use.

	MNIST	**fMNIST**	**SVHN**	**CIFAR-10**
License	CC BY-SA 3.0	MIT	CC0 1.0	MIT

	STL-10	**QuickDraw**	**CelebA**
License	N-C	CC BY 4.0	N-C

A.2 Implementation details

The experiments in this work were conducted on an NVIDIA A100 GPU. The backbone network used for all models as well as all datasets consists of an ImageNet (Deng et al., 2009) pretrained ResNet34 (He et al., 2016). The size of the latent vector is 512 and the batch size is 128 for all datasets as well as models. Stochastic gradient descent (SGD) is used as the optimizer for training ResNet34 with momentum 0.9 for CelebA and 0.5 for all other datasets. For ProtoVAE and FLINT, an Adam (Kingma & Ba, 2015) optimizer is used. Other hyperparameters including learning rate, number of epochs and number of prototypes are mentioned in Table 10. Note that unlike other SEMs, KMEx requires tuning of only one additional hyperparameter i.e., the number of prototypes per class, compared to the closest black-box model.

Table 10: Hyperparameter values for KMEx, ProtoPNet, FLINT and ProtoVAE for all the datasets.

		MNIST	**fMNIST**	**SVHN**	**CIFAR-10**	**STL-10**	**QuickDraw**	**CelebA**
KMEx	No. of prototypes per class	5	5	5	5	5	5	20
	No. of epochs	10	10	10	30	30	30	10
	Learning rate	0.001	0.001	0.001	0.001	0.001	0.001	0.001
ProtoPNet	No. of prototypes per class	5	5	5	5	5	5	20
	No. of epochs							
	•warm	5	5	5	5	5	5	5
	•train	15	15	15	15	15	15	15
	•push interval	5	5	5	5	5	5	5
	Learning rates							
	•joint, warm, last layer & prototypes	0.001	0.001	0.001	0.001	0.001	0.001	0.001
	Loss weights							
	•Cross entropy	1	1	1	1	1	1	1
	•Clustering	0.8	0.8	0.8	0.8	0.8	0.8	0.8
	•Separation	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08	-0.08
	•$l1$	0.004	0.004	0.004	0.004	0.004	0.004	0.004
FLINT	No. of prototypes per class	5	5	5	5	5	5	20
	No. of epochs	10	10	10	30	30	30	10
	Loss weights							
	•Cross entropy	0.8	0.8	0.8	0.8	0.8	0.8	0.8
	•Input fidelity	0.8	0.8	0.8	0.8	0.8	0.8	0.8
	•Output fidelity	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	•Conciseness	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	•Entropy	0.2	0.2	0.2	0.2	0.2	0.2	0.2
	•Diversity	0.2	0.2	0.2	0.2	0.2	0.2	0.2
ProtoVAE	No. of prototypes per class	5	5	5	5	5	5	20
	No. of epochs	20	20	20	60	60	60	20
	Loss weights							
	•Cross Entropy	1	1	1	1	1	1	1
	•Reconstruction	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	•KL Divergence	1	100	100	100	100	100	100
	•Orthogonality	0.1	0.1	0.1	0.1	0.1	0.1	0.1

A.3 Faithfulness of explanations

In this section, we qualitatively evaluate the faithfulness of explanations generated by an SEM to the closest black-box. For this, as mentioned in the main text, we compute Layer-wise Relevance Propagation (LRP) maps (Bach et al., 2015) for prediction probabilities. In Figure 6, we visualize the LRP maps for random images from the CIFAR-10, CelebA and MNIST datasets. The LRP maps for the black-box ResNet34 and SEMs ProtoPNet, ProtoVAE and KMEx are shown. For MNIST, we also show LRP maps for a CNN backbone. The CNN architecture used is from (Gautam et al., 2022). As observed, instead of producing non robust explanations, KMEx remains most faithful to the black-box. This makes KMEx the SEM closest to the corresponding black-box, thereby proving to be an efficient baseline.

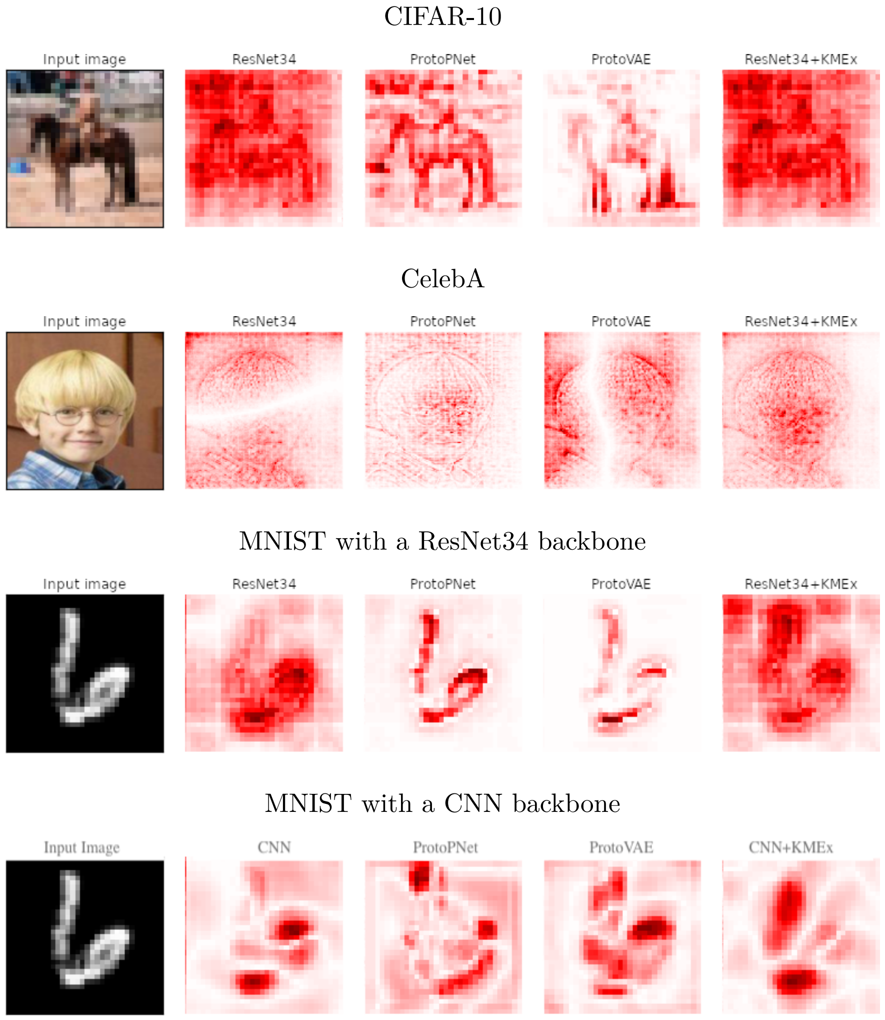


Figure 6: Normalized LRP maps computed on different datasets and with different architectures.

A.4 Preliminary results for CUB200

We report here preliminary results for the CUB200 (Welinder et al., 2010) dataset. The data consists of 6000 images of 200 classes of birds. We also present a naive extension of KMEx at the patch level. The idea is to compute the patch prototypes right before the final average pool ($7 \times 7 = 49$ patches per image). The class prediction for an image are then derived as the majority vote of the KMEx predictions for each patch.

We report accuracy in percentage in Table 11 for a ResNet34 and its KMEx based on the full images and patches both with 10 prototypes per class. Similarly to (Chen et al., 2019b), we show in Figure 7 the patch prototypes for 10 classes as red rectangles in the closest training image. Note, some prototypes capture background regions, indicating that the model has learned to exploit background cues.

The drop in accuracy when using patches is not surprising, since the task is more complex. Yet, the results are encouraging and highlight the versatility of KMEx.

Table 11: Classification performance on CUB200 dataset.

	ResNet34	ResNet34+KMEx	
		Full images	Patches
Accuracy	78.6	78.4	70.0

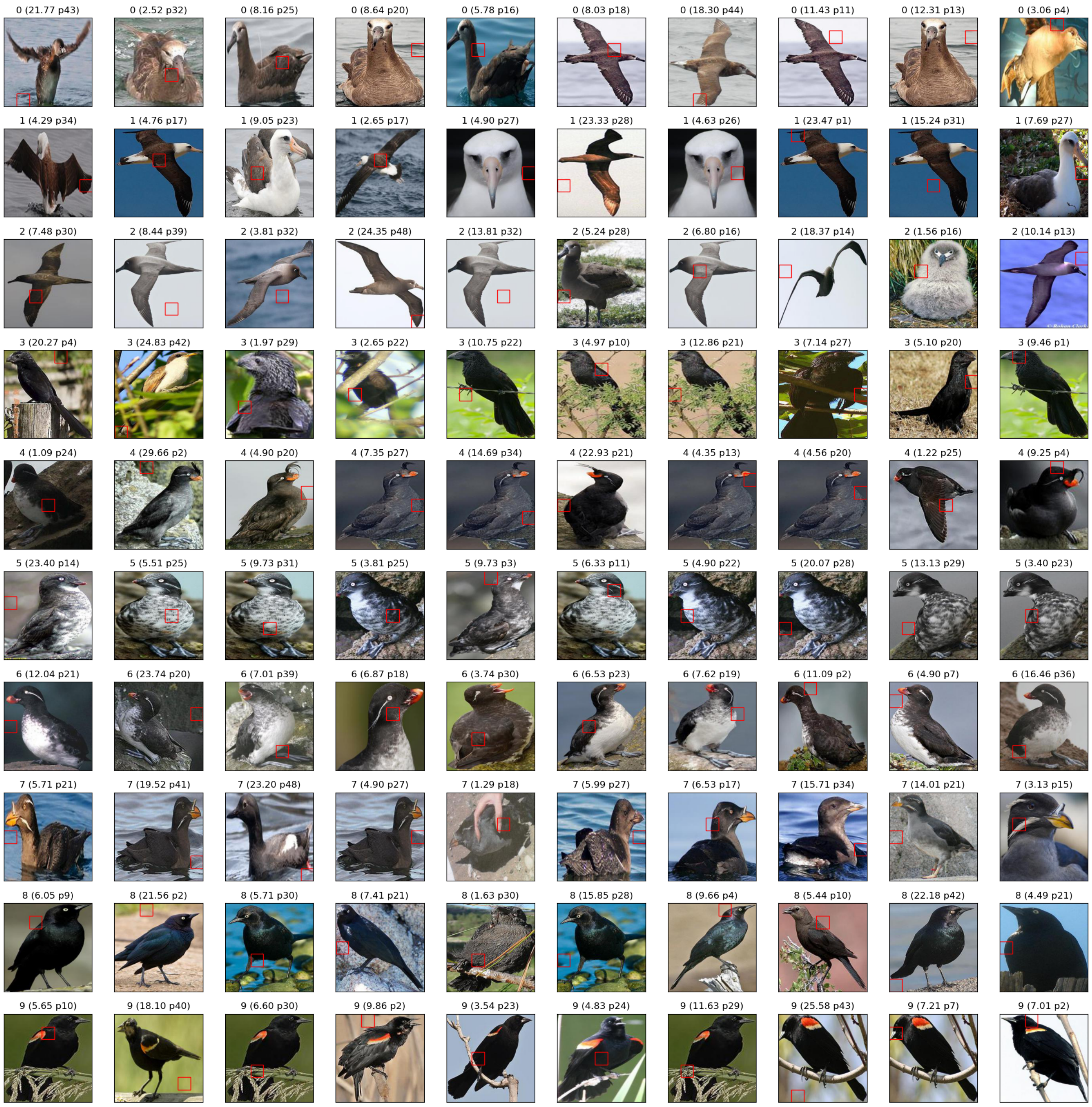


Figure 7: Patch prototypes labeled with class id, importance and patch id.

A.5 Additional qualitative results

As mentioned in the main text, quantitative evaluation is not meant to replace the qualitative evaluation of SEMs. Therefore, in this section, we provide qualitative results including prototype visualizations for KMEx. We also show visual classification strategy used by KMEx using the prototypes for different test examples, thereby exhibiting *this* (image) looks like *that* (prototype). We show this for both correctly and incorrectly test examples to further understand the decision making process of our SEM. Additionally, we also qualitatively compare the diversity of prototypes for different SEMs.

A.6 Prototypes learned by KMEx

We visualize the prototypes of KMEx as the images in the training set that have the closest embedding in the latent space to the prototypes. The prototypes are shown for MNIST, fMNIST, SVHN and STL-10 in Figure 8. It can be observed that the prototypes are very diverse and therefore efficiently represent different subgroups of classes.

A.6.1 KMEx: *This* looks like *that*

We now visually demonstrate the decision making process of the proposed SEM. In Figure 9, for random test examples, we show the closest prototype for the MNIST, fMNIST, CelebA, SVHN, STL-10 and CIFAR-10 datasets. It can be observed that the images look very similar to the closest prototypes, which illustrates that representative prototypes are learned. Additionally, we also demonstrate this behavior for misclassified examples (marked by a red rectangle) in Figure 9. As can be seen, the misclassified test images look very similar to prototypes from different classes. Therefore, the simple *this* looks like *that* behavior exhibited by KMEx is able to provide meaningful and transparent decisions.

A.6.2 Diversity of prototypes: qualitative evaluation

We now compare the prototypes of different SEMs, thereby qualitatively comparing the diversity of the prototypes. For consistency and fair comparison, we visualize the closest training images for all the models. In Figure 11, we visualize the prototypes learned by KMEx and ProtoPNet for the MNIST dataset. As observed, ProtoPNet's prototypes lack diversity, which is especially visible for classes 1, 4 5 and 7. Applying KMEx on ProtoPNet's embeddings drastically improves the diversity, as shown in Figure 11 (right). Similarly, we compare the prototypes for KMEx and ProtoVAE in Figure 12 for the STL-10 dataset and for KMEx and FLINT in Figure 10. In all the cases, KMEx efficiently improves the diversity of the prototypes.

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. Protovae: A trustworthy self-explainable prototypical variational model. *Advances in Neural Information Processing Systems*, 35:17940–17952, 2022.

Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *Pattern Recognition*, 136:109172, 2023.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018.

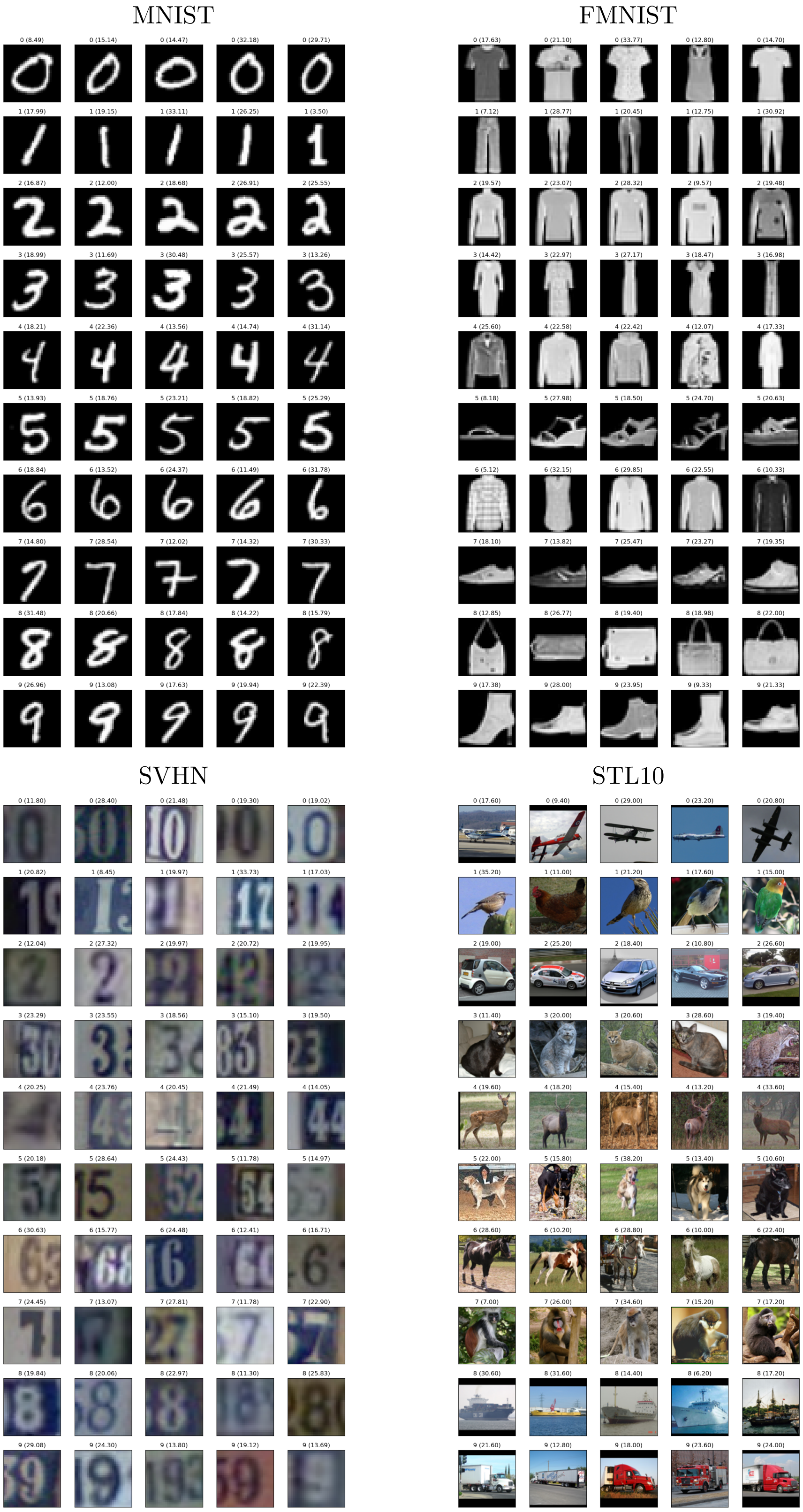


Figure 8: Prototypes learned by KMEx for several datasets. The class label is written on the top of each prototype image along with its importance in the brackets.

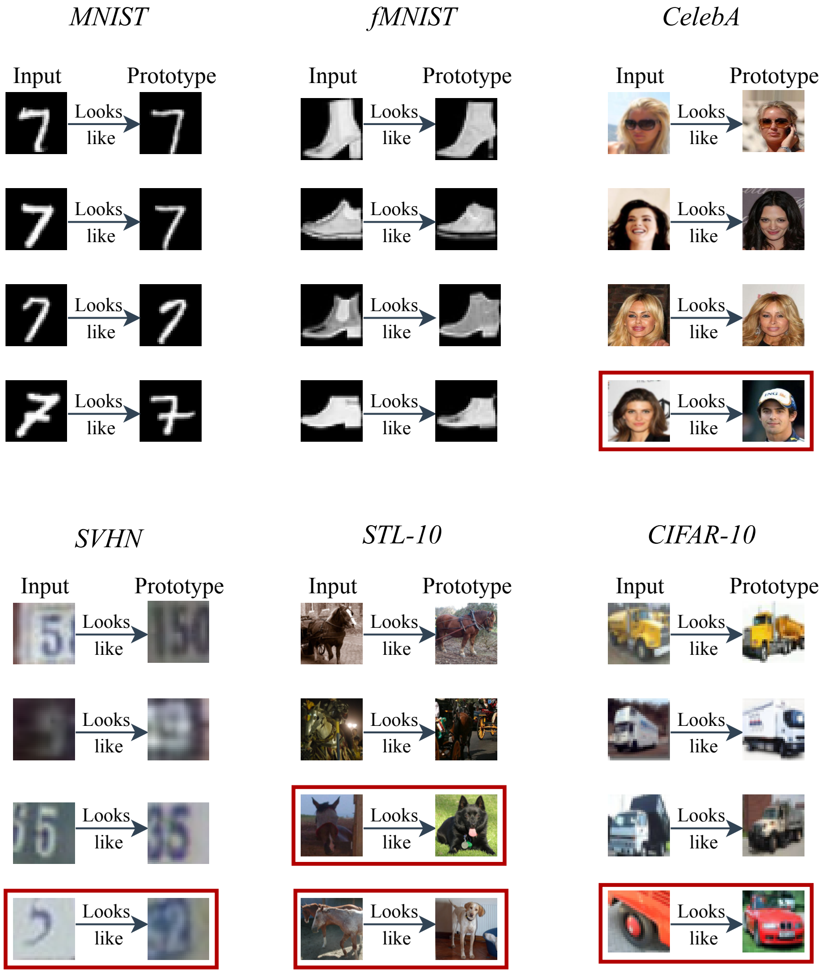


Figure 9: *This* looks like *that* behavior exhibited by KMEx for MNIST, fMNIST, QuickDraw, SVHN, STL-10 and CIFAR-10 datasets. The classification is based on 1 nearest neighnor, therefore only the closest prototype for each input image is required as the explanation. Misclassified examples are marked in red.

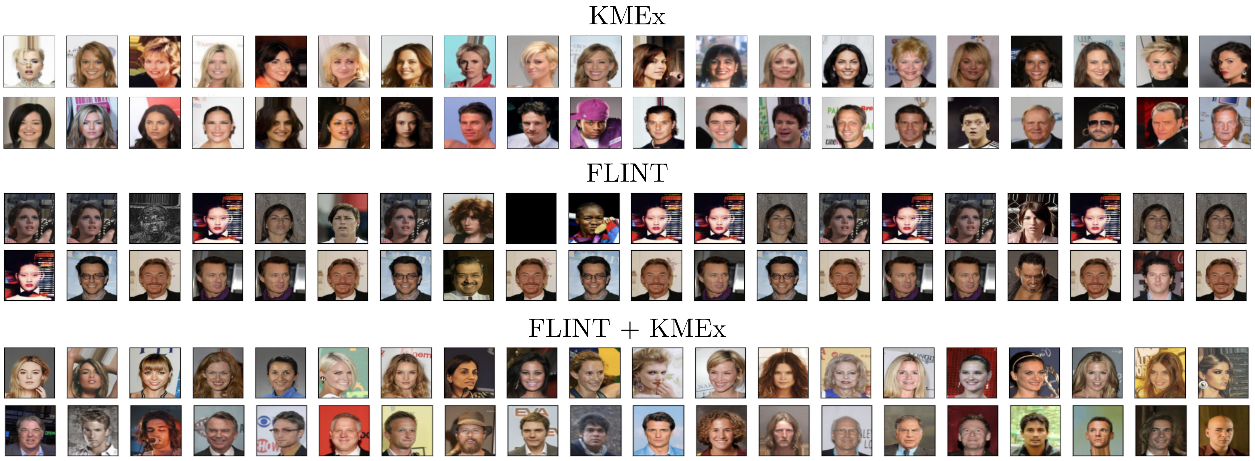


Figure 10: Prototypes learned by KMEx (top) and FLINT (middle) and FLINT-KMEx (bottom) for the CelebA dataset. KMEx generates more diverse prototypes and is again additionally able to improve the prototypes learned over FLINT's embeddings.

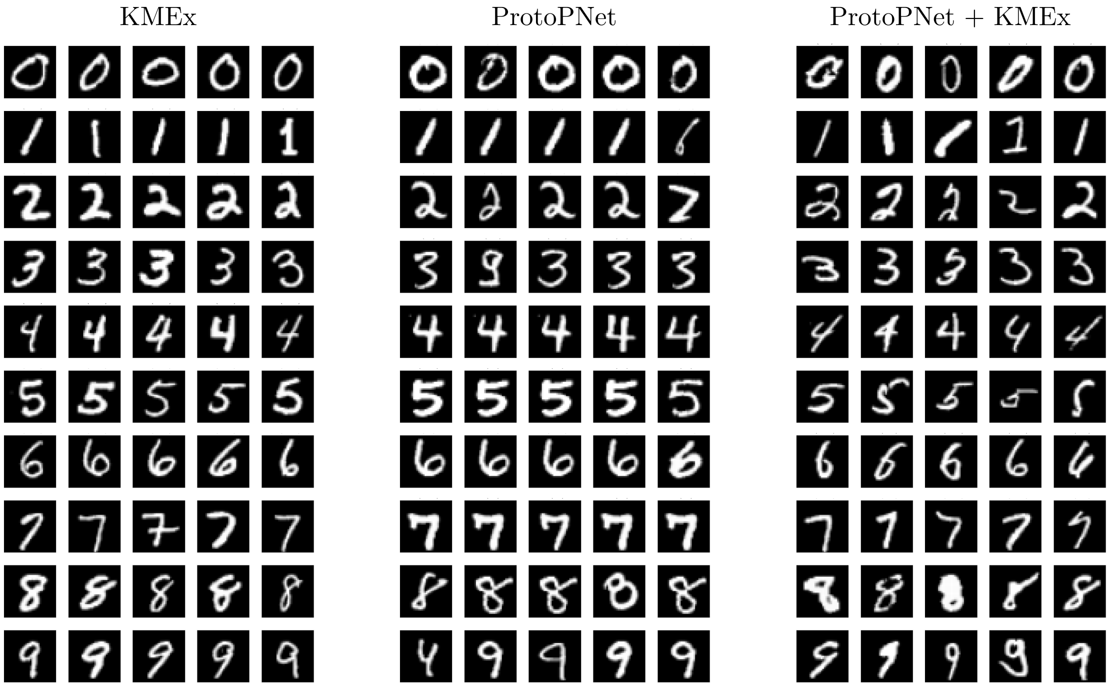


Figure 11: Prototypes learned by KMEx (left) and ProtoPNet (middle) and ProtoPNet-KMEx (right) for the MNIST dataset. KMEx generates more diverse prototypes and is additionally able to improve the prototypes learned over ProtoPNet's embeddings.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

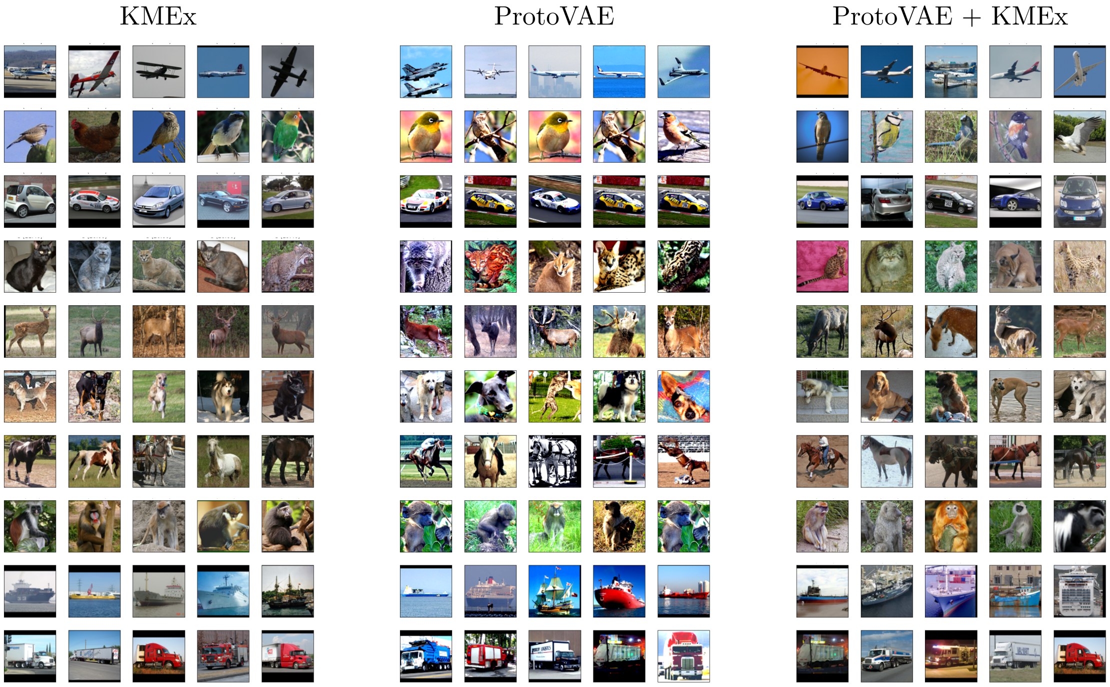


Figure 12: Prototypes learned by KMEx (left) and ProtoVAE (middle) and ProtoVAE-KMEx (right) for the STL-10 dataset. KMEx generates more diverse prototypes and is again additionally able to improve the prototypes learned over ProtoVAE's embeddings.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang. Relevance-cam: Your model already knows where to look. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14944–14953, June 2021.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Jan MacDonald, Stephan Wäldchen, Sascha Hauch, and Gitta Kutyniok. A rate-distortion framework for explaining neural network decisions. *ArXiv*, abs/1905.11092, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Jayneel Parekh, Pavlo Mozharovskyi, and Florence d'Alché Buc. A framework to learn with interpretation. *Advances in Neural Information Processing Systems*, 34:24273–24285, 2021.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Paper V

Investigating the Fairness of Large Language Models for Predictions on Tabular Data

Yanchen Liu, Srishti Gautam, Jiaqi Ma, and, Himabindu Lakkaraju

Under Review

Investigating the Fairness of Large Language Models for Predictions on Tabular Data

Yanchen Liu[1], **Srishti Gautam**[2], **Jiaqi Ma**[3], **Himabindu Lakkaraju**[1]
[1]Harvard University, [2]UiT The Arctic University of Norway,
[3]University of Illinois Urbana-Champaign

Abstract

Recent literature has suggested the potential of using large language models (LLMs) to make predictions for tabular tasks. However, LLMs have been shown to exhibit harmful social biases that reflect the stereotypes and inequalities present in the society. To this end, as well as the widespread use of tabular data in many high-stake applications, it is imperative to explore the following questions: what sources of information do LLMs draw upon when making predictions for tabular tasks; whether and to what extent are LLM predictions for tabular tasks influenced by social biases and stereotypes; and what are the consequential implications for fairness? Through a series of experiments, we delve into these questions and show that LLMs tend to inherit social biases from their training data which significantly impact their fairness in tabular prediction tasks. Furthermore, our investigations show that in the context of bias mitigation, though in-context learning and fine-tuning have a moderate effect, the fairness metric gap between different subgroups is still larger than that in traditional machine learning models, such as Random Forest and shallow Neural Networks. This observation emphasizes that the social biases are inherent within the LLMs themselves and inherited from their pre-training corpus, not only from the downstream task datasets. Besides, we demonstrate that label-flipping of in-context examples can significantly reduce biases, further highlighting the presence of inherent bias within LLMs.

1 Introduction

Many recent works propose to use large language models (LLMs) for tabular prediction (Slack & Singh, 2023; Hegselmann et al., 2023), where the tabular data is serialized as natural language and provided to LLMs with a short description of the task to solicit predictions. Despite the comprehensive examination of fairness considerations within conventional machine learning approaches applied to tabular tasks (Bellamy et al., 2018), the exploration of fairness-related issues in the context of employing LLMs for tabular predictions remains a relatively underexplored domain.

Previous research has shown that LLMs, such as GPT-3 (Brown et al., 2020), GPT-3.5, GPT-4 (OpenAI, 2023) can exhibit harmful social biases (Abid et al., 2021a; Basta et al., 2019), which may even worsen as the models become larger in size (Askell et al., 2021; Ganguli et al., 2022). These biases are a result of the models being trained on text generated by humans that presumably includes many examples of humans exhibiting harmful stereotypes and discrimination and reflects the biases and inequalities present in society (Bolukbasi et al., 2016; Zhao et al., 2017), which can lead to perpetuation of discrimination and stereotype (Abid et al., 2021a; Bender et al., 2021).

Considering that tabular data finds extensive use in high-stakes domains (Grinsztajn et al., 2022) where information is typically structured in tabular formats as a natural byproduct of relational databases (Borisov et al., 2022), it is of paramount importance to thoroughly examine the fairness implications of utilizing LLMs for predictions on tabular data. In this paper, we conduct a series of investigation centered around this critical aspect, with the goal of discerning the underlying information sources upon which LLMs rely when making tabular predictions. Through this exploration, our investigation aims to ascertain whether, and to what degree, LLMs are susceptible to being influenced by social biases and stereotypes in the context of tabular data predictions.

Through experiments using GPT-3.5 to make predictions for tabular data in a zero-shot setting, we demonstrate that LLMs exhibit significant social biases (Section 4). This evidence confirms that LLMs inherit social biases from their training corpus and tend to rely on these biases when making predictions for tabular data.

Furthermore, we demonstrate that providing LLMs with few-shot examples (in-context learning) or fine-tuning them on the entire training dataset both exhibit moderate effect on bias mitigation (Sections 5.1 and 6.1). Nevertheless, the achieved fairness levels remain below what is typically attained with traditional machine learning methods, including Random Forests and shallow Neural Networks, once again underscoring the presence of inherent bias in LLMs. Additionally, our investigation further reveals that flipping the labels of the in-context examples significantly narrows the gap in fairness metrics across different subgroups, but comes at the expected cost of a reduction in predictive performance. This finding, in turn, further emphasizes and reaffirms the indication of inherent bias present in LLMs (Section 5.2). Additionally, we further show that while resampling the training set is a known and effective method for reducing biases in traditional machine learning methods like Random Forests and shallow Neural Networks, it proves to be less effective when applied to LLMs (Section 6.2).

These collective findings underscore the significant influence of social biases on LLMs' performance in tabular predictions. These biases significantly undermines the fairness and poses substantial potential risks for using LLMs on tabular data, especially considering that tabular data is extensively used in high-stakes domains, highlighting the need for more advanced and tailored strategies to address these biases effectively. Straightforward methods like in-context learning and data resampling may not be sufficient in this context.

2 Related work

Fairness and Social Biases in LLMs Fairness is highly desirable for ensuring the credibility and trustworthiness of algorithms. It has been demonstrated that unfair algorithms can reflect societal biases in their decision-making processes (Bender et al., 2021; Bommasani, 2021), primarily stemming from the biases present in their training data (Caliskan et al., 2017; Zhao et al., 2017). LLMs, pre-trained on vast natural language datasets, are particularly susceptible to inheriting these social biases and have been shown to exhibit biases related to gender (Lucy & Bamman, 2021), religion (Abid et al., 2021b) and language variants (Ziems et al., 2023; Liu et al., 2023a). These social biases can lead to perpetuation of discrimination and stereotype (Abid et al., 2021a; Bender et al., 2021; Weidinger et al., 2021). While recent literature has made strides in addressing these issues, there still exists a significant gap in comprehensively assessing fairness in LLMs and its mitigation strategies for tabular data.

Tabular Tasks and LLM for Tabular Data Tabular data extensively exist in many domains (Shwartz-Ziv & Armon, 2021). Previous works propose to utilize self-supervised deep techniques for tabular tasks (Yin et al., 2020; Arik & Pfister, 2021), which, however, still underperform ensembles of gradient boosted trees in the fully supervised setting (Grinsztajn et al., 2022). This disparity in performance can be attributed to the locality, sparsity and mixed data types of tabular data. In recent times, LLMs have undergone intensive training using vast amounts of natural language data, which has enabled them to exhibit impressive performance across various downstream tasks (Brown et al., 2020; OpenAI, 2023), even with little or no labeled task data. Therefore, recent approaches by Hegselmann et al. (2023); Slack & Singh (2023) suggests serializing the tabular data as natural language, which is provided to LLM along with a short task description to generate predictions for tabular tasks.

However, tabular data plays a crucial role in numerous safety-critical and high-stakes domains (Borisov et al., 2022; Grinsztajn et al., 2022), which makes the fairness particularly crucial when employing LLMs for making predictions on tabular data, especially considering the inherent social biases present in LLMs. Despite the importance, this still remains largely unexplored. To the best of our knowledge, we regard our work as one of the most comprehensive investigations into the fairness issues arising when using LLMs for predictions on tabular data.

In-Context Learning Significant improvements for various tasks have been achieved by providing in-context examples to LLMs (Brown et al., 2020; Liu et al., 2022; 2023b). However, previous research by Min et al. (2022); Wei et al. (2023b); Lyu et al. (2023) illustrate that the effective performance of in-context learning largely hinges on semantic priors rather than learning the input-label mapping (Akyürek et al., 2022; Xie et al., 2022; Von Oswald et al., 2023) and the labels of the in-context examples might not play a crucial role in in-context learning, with flipped or random labels sometimes having minimal impact on performance. Despite these findings, the predominant focus of existing investigation of in-context learning remains on conventional natural language processing tasks (Zhao et al., 2021; Min et al., 2022; Wei et al., 2023a;b), largely overlooking the domain of tabular data. Furthermore, the fairness of in-context learning and the impact of flipped labels on this fairness is yet to be thoroughly investigated.

3 Experimental Setup

In this section, we outline the general setup of the experiments conducted in our work.

3.1 Models

In our work, we focus our experiments on GPT-3.5 (engine `GPT-3.5-turbo`) - an LLM released by OpenAI, trained with instruction tuning (Sanh et al., 2022; Wei et al., 2022) and reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), aligning LLMs with human preferences. Furthermore, we also compare its performance with conventional machine learning models in order to gain insight into the propagation of biases found within LLMs, which are likely mirrored in traditional models as well, consequently, offering valuable additional perspectives on the biases inherent in the training of LLMs. For this, we employ two widely used models for tabular data i.e, Random Forests (RF) and a shallow Neural Network (NN) of 3 layers. We provide additional implementation details for these two models in the Appendix B.

3.2 Datasets and Protected Attributes

To explore the fairness of LLMs in making predictions for tabular data, we utilize the following three widely used tabular datasets for assessing the fairness of traditional ML models: *Adult Income* (**Adult**) Dataset (Becker & Kohavi, 1996), **German Credit** Dataset (Dua & Graff, 2019), and *Correctional Offender Management Profiling for Alternative Sanctions* (**COMPAS**) Dataset (Larson et al., 2016). In this section, we introduce each dataset and discuss its associated protected attributes.

Adult The *Adult Income* dataset (Adult) is extracted from the 1994 U.S. Census Bureau database. The task is to predict whether a person earns more than $50,000 per year based on their profile data (*greater than 50K* or *less than or equal to 50K*). The original Adult Income Dataset contains 14 features. Following previous work (Slack & Singh, 2023), we retain only 10 features: *"workclass"*, *"hours per week"*, *"sex"*, *"age"*, *"occupation"*, *"capital loss""*, *"education"*, *"capital gain"*, *"marital status"*, and *"relationship"*. Our analysis on Adult primarily focuses on *sex* as the protected attribute, and *female* is acknowledged as a disadvantaged group.

German Credit The German Credit dataset is used to classify individuals based on their profile attributes as good or bad credit risks (*good* or *bad*). The raw dataset comprises 20 attributes. Consistent with previous work, we only retain the following features: *"age"*, *"sex"*, *"job"*, *"housing"*, *"saving accounts"*, *"checking account"*, *"credit amount"*, *"duration"*, and *"purpose"*. Same with Adult, *sex* is considered as a protected attribute in the German Credit dataset and *female* as the marginalized group.

COMPAS The COMPAS dataset comprises the outcomes from the *Correctional Offender Management Profiling for Alternative Sanctions* commercial algorithm, utilized to evaluate a convicted criminal's probability of reoffending. Known for its widespread use by judges and parole officers, COMPAS has gained notoriety for its bias against African-Americans. The raw COMPAS Recidivism dataset contains more than 50 attributes. Following the approach of Larson et al. (2016), we perform necessary preprocessing, group *“race”* into *African-American* and *Not African-American*, and only consider the features *“sex”*, *“race”*, *“age”*, *“charge degree”*, *“priors count”*, *“risk”* and *“two year recid”* (target). We frame the task as predicting whether an individual will recidivate in two years (*Did Not Reoffend* or *Reoffended*) based on their demographic and criminal history. For the COMPAS dataset, we consider *race* as the protected attribute.

A detailed description for each feature of the considered datasets is provided in Appendix A.

3.3 Serialization and Prompt Templates

To employ the LLM for making predictions on these tabular datasets, each data point is first serialized as text. Following previous works on LLM for tabular predictions (Hegselmann et al., 2023; Slack & Singh, 2023), we format the feature names and values into strings as “$f_1 : x_1, \ldots, f_d : x_d$”, and prompt to LLM along with a task description, as illustrated following:

```
You must predict if income exceeds $50K/yr. Answer with one of the
following: greater than 50K | less than or equal to 50K.
Example 1 -
workclass: Private
hours per week: 20
sex: Male
age: 17
occupation: Other-service
capital loss: 0
education: 10th
capital gain: 0
marital status: Never-married
relationship: Own-child
Answer: less than or equal to 50K
...

workclass: Private
hours per week: 40
sex: Female
age: 24
occupation: Sales
capital loss: 0
education: Some-college
capital gain: 0
marital status: Never-married
relationship: Own-child
Answer:
```

Figure 1: Prompt Template for **Adult** Dataset.

The example above is from Adult dataset, where text in blue represents the task description, text in green denotes optional few-shot examples (only used in in-context learning), and text in red indicates the test example. We provide the prompt templates for the other two datasets in Appendix C.

3.4 Evaluation Metrics

To assess fairness in the aforementioned datasets, we examine the disparity between different subgroups of protected attributes using the following common fairness metrics: accuracy, F1 score, statistical parity and equality of opportunity. Here, we briefly explain each evaluation metric.

Accuracy and F1 As the most basic metric, assessing accuracy among different subgroups ensures that the model delivers consistent performance across all groups, without undue favor to any particular subgroups. Considering that the evaluated datasets may be imbalanced, especially among different subgroups, the F1 Score computes the harmonic mean of precision and recall, offering a balanced perspective between these two metrics.

Statistical Parity Statistical parity is attained when *positive* decision outcomes (e.g., being predicted as a good credit risk) are independent of the protected attributes. This metric assesses whether different subgroups receive similar treatment from the model. For each subgroup z_i of each protected attribute Z, we calculate

$$P(\hat{Y} = 1 | Z = z_i).$$

Then we calculate the Statistical Parity Difference (SPD) of this protected attribute as

$$SPD = P(\hat{Y} = 1 | Z = z_1) - P(\hat{Y} = 1 | Z = z_2),$$

where z_1 is the minority group and z_2 is the majority.

Equality of Opportunity Equality of opportunity requires that qualified individuals have an equal chance of being correctly classified by the model, regardless of their membership in a protected group. This metric ensures equal *true positive* rates between different subgroups, providing equal opportunities for each subgroup. Similar as statistical parity, for equality of opportunity, we calculate the Equal Opportunity Difference (EOD) as

$$EOD = P(\hat{Y} = 1 | Y = 1, Z = z_1) - P(\hat{Y} = 1 | Y = 1, Z = z_2).$$

Each of these metrics offers a different perspective on fairness. For each subgroup from each protected attribute, we will compute every aforementioned metric. A model demonstrating good fairness should show minimal gaps in these fairness metrics between different subgroups. Considering them together can provide a more comprehensive evaluation of the model's fairness across different subgroups, ensuring that individuals are not unfairly disadvantaged based on their membership in a protected group.

4 Zero-Shot Prompting for Tabular Data

To explore the fairness of LLMs when making predictions on tabular data, we first conduct experiments in a zero-shot setting. We assess the fairness metrics of the outcomes and examine whether LLMs without any finetuning or few-shot examples would be influenced by social biases and stereotypes for tabular predictions. We run all the experiments 5 times and compute the mean and standard deviation.

In Tables 1-3, we present the evaluation of four fairness metrics, namely accuracy (ACC), F1 score (F1), statistical parity (SP), and equality of opportunity (EoO), for GPT-3.5 (engine `GPT-3.5-turbo`), RF and NN models on the **Adult**, **German Credit** and **COMPAS** datasets, respectively. For the Adult and German Credit datasets, the subgroups *female* and *male* are assessed regarding the protected attribute *sex*, identifying *female* as a disadvantaged group. In the COMPAS dataset, we evaluate *race* as protected attributes, recognizing African American (*AA*) as the disadvantaged group.

It is notable that when utilizing LLMs to make predictions for tabular data directly, without any fine-tuning or in-context learning, a significant fairness metric gap between the protected and non-protected groups is observed for GPT-3.5 (highlighted in red). For instance, the EoO difference between *male* and *female* on the *Adult* dataset reaches 0.483, indicating a substantial disadvantage for the *female* group. Additionally, when compared with traditional methods like RF and NN, the bias in zero-shot predictions made by GPT-3.5 is significantly larger for the Adult dataset. This observation suggests an inherent gender bias in GPT-3.5. For COMPAS dataset, the racial bias in zero-shot setting is comparatively lower than RF and NN but is still effectively high.

Exceptionally, GPT-3.5 is extremely biased for German Credit dataset where it classifies almost everything into *'good credit'* class in the zero-shot setting, thus rendering the difference in SP and EoO for both subgroups to be near 0. The accuracy for each subgroup is near to 50%, performing similar to random guessing. The possible reason might be that the German Credit dataset is too challenging for making tabular predictions with LLMs (especially, the features of German Credit

				ACC	F1	SP	EoO
GPT-3.5-turbo	Zero-Shot		*f*	0.898 $_{0.001}$	0.711 $_{0.002}$	0.065 $_{0.001}$	0.357 $_{0.000}$
			m	0.742 $_{0.002}$	0.727 $_{0.002}$	0.464 $_{0.003}$	0.840 $_{0.004}$
			d	0.157 $_{0.002}$	-0.016 $_{0.002}$	-0.399 $_{0.003}$	-0.483 $_{0.004}$
	Few-shot	Regular	*f*	0.899 $_{0.002}$	0.735 $_{0.003}$	0.082 $_{0.002}$	0.429 $_{0.000}$
			m	0.781 $_{0.003}$	0.749 $_{0.002}$	0.339 $_{0.003}$	0.700 $_{0.003}$
			d	0.118 $_{0.004}$	-0.014 $_{0.004}$	-0.257 $_{0.005}$ ↓	-0.271 $_{0.003}$ ↓
		Label-flipping	*f*	0.682 $_{0.004}$	0.590 $_{0.003}$	0.396 $_{0.006}$	0.800 $_{0.013}$
			m	0.614 $_{0.002}$	0.605 $_{0.002}$	0.545 $_{0.001}$	0.763 $_{0.003}$
			d	0.068 $_{0.004}$	-0.015 $_{0.004}$	-0.148 $_{0.006}$ ✓	0.037 $_{0.014}$ ✓
	Finetuning	Regular	*f*	0.915 $_{0.014}$	0.773 $_{0.036}$	0.079 $_{0.002}$	0.476 $_{0.048}$
			m	0.799 $_{0.005}$	0.754 $_{0.005}$	0.269 $_{0.036}$	0.613 $_{0.053}$
			d	0.116 $_{0.009}$	0.020 $_{0.039}$	-0.190 $_{0.035}$ ↓	-0.137 $_{0.098}$ ↓
		Oversampling	*f*	0.913 $_{0.016}$	0.770 $_{0.042}$	0.081 $_{0.004}$	0.476 $_{0.067}$
			m	0.813 $_{0.007}$	0.780 $_{0.003}$	0.310 $_{0.038}$	0.702 $_{0.048}$
			d	0.100 $_{0.013}$	-0.010 $_{0.041}$	-0.229 $_{0.030}$	-0.226 $_{0.077}$
		Undersampling	*f*	0.912 $_{0.015}$	0.770 $_{0.046}$	0.086 $_{0.006}$	0.488 $_{0.084}$
			m	0.794 $_{0.006}$	0.751 $_{0.001}$	0.285 $_{0.031}$	0.631 $_{0.044}$
			d	0.118 $_{0.021}$	0.018 $_{0.046}$	-0.200 $_{0.025}$	-0.143 $_{0.040}$
RF		Regular	*f*	0.914 $_{0.002}$	0.767 $_{0.006}$	0.075 $_{0.003}$	0.457 $_{0.010}$
			m	0.822 $_{0.005}$	0.783 $_{0.005}$	0.269 $_{0.004}$	0.652 $_{0.004}$
			d	0.092 $_{0.004}$	-0.015 $_{0.005}$	-0.195 $_{0.003}$	-0.195 $_{0.012}$
		Oversampling	*f*	0.912 $_{0.006}$	0.770 $_{0.011}$	0.084 $_{0.005}$	0.486 $_{0.012}$
			m	0.824 $_{0.002}$	0.785 $_{0.002}$	0.270 $_{0.003}$	0.656 $_{0.006}$
			d	0.087 $_{0.005}$	-0.015 $_{0.01}$	-0.185 $_{0.004}$	-0.170 $_{0.011}$
		Undersampling	*f*	0.917 $_{0.004}$	0.776 $_{0.011}$	0.075 $_{0.001}$	0.471 $_{0.018}$
			m	0.814 $_{0.003}$	0.771 $_{0.004}$	0.263 $_{0.002}$	0.627 $_{0.009}$
			d	0.103 $_{0.005}$	0.005 $_{0.011}$	-0.187 $_{0.001}$	-0.156 $_{0.018}$
NN		Regular	*f*	0.917 $_{0.003}$	0.778 $_{0.019}$	0.081 $_{0.016}$	0.490 $_{0.068}$
			m	0.819 $_{0.006}$	0.773 $_{0.015}$	0.250 $_{0.045}$	0.614 $_{0.079}$
			d	0.098 $_{0.005}$	0.006 $_{0.009}$	-0.169 $_{0.032}$	-0.123 $_{0.033}$
		Oversampling	*f*	0.916 $_{0.004}$	0.794 $_{0.013}$	0.100 $_{0.016}$	0.562 $_{0.058}$
			m	0.813 $_{0.012}$	0.774 $_{0.008}$	0.286 $_{0.044}$	0.663 $_{0.056}$
			d	0.103 $_{0.011}$	0.020 $_{0.018}$	-0.186 $_{0.030}$	-0.102 $_{0.038}$
		Undersampling	*f*	0.904 $_{0.005}$	0.748 $_{0.014}$	0.084 $_{0.007}$	0.452 $_{0.030}$
			m	0.813 $_{0.006}$	0.774 $_{0.005}$	0.283 $_{0.023}$	0.659 $_{0.031}$
			d	0.090 $_{0.006}$	-0.026 $_{0.014}$	-0.199 $_{0.018}$	-0.206 $_{0.031}$

Table 1: **Fairness evaluation for Adult dataset**. This table depicts the evaluation of accuracy (ACC), F1 score (F1), statistical parity (SP), and equality of opportunity (EoO) metrics for the subgroup - *female* (*f*) and *male* (*m*) as well as the difference (*d*) between them. We list the protected group first. The significant fairness disparities are highlighted in red. Both in-context learning and finetuning can lead to bias reduction (indicated by ↓), and label-flipped in-context learning can further minimize bias (indicated by ✓).

are ambiguous and vague). This also suggests that, when using LLM to make predictions on tabular data, a potential description of table feature names is favorable.

These findings demonstrate the tendency of LLMs to rely on social biases and stereotypes inherited from their training corpus when applied to tabular data. This implies that using LLMs for predictions on tabular data may incur significant fairness risks, including the potential to disproportionately disadvantage marginalized communities as well as exacerbate social biases and stereotypes present in society. This is particularly concerning given the widespread application of tabular data in high-stake contexts, further magnifying the potential for harm.

5 Few-Shot Prompting for Tabular Data

As demonstrated in Section 4, employing LLMs for predictions on tabular data reveals significant social biases in a zero-shot setting. Instead of directly utilizing LLMs for zero-shot tabular predictions, this section explores whether including few-shot examples during prompting will reduce or amplify these biases. To delve deeper into the influence of few-shot examples during in-context

				ACC	F1	SP	EoO
GPT-3.5-turbo	Zero-Shot		*f*	0.471 $_{0.011}$	0.359 $_{0.021}$	0.980 $_{0.011}$	1.000 $_{0.000}$
			m	0.556 $_{0.000}$	0.357 $_{0.000}$	0.984 $_{0.000}$	0.972 $_{0.000}$
			d	-0.084 $_{0.011}$	0.002 $_{0.021}$	-0.004 $_{0.011}$	0.028 $_{0.000}$
	Few-shot	Regular	*f*	0.610 $_{0.013}$	0.593 $_{0.013}$	0.348 $_{0.027}$	0.453 $_{0.029}$
			m	0.606 $_{0.007}$	0.603 $_{0.008}$	0.337 $_{0.007}$	0.450 $_{0.012}$
			d	0.003 $_{0.012}$	-0.010 $_{0.011}$	0.011 $_{0.027}$	0.003 $_{0.026}$
		Label-flipping	*f*	0.614 $_{0.011}$	0.606 $_{0.012}$	0.695 $_{0.011}$	0.842 $_{0.000}$
			m	0.559 $_{0.013}$	0.538 $_{0.011}$	0.638 $_{0.013}$	0.672 $_{0.023}$
			d	0.056 $_{0.021}$	0.067 $_{0.021}$	0.057 $_{0.012}$	0.170 $_{0.023}$
	Finetuning	Regular	*f*	0.571 $_{0.067}$	0.567 $_{0.062}$	0.619 $_{0.101}$	0.711 $_{0.186}$
			m	0.548 $_{0.011}$	0.539 $_{0.023}$	0.532 $_{0.123}$	0.569 $_{0.098}$
			d	0.024 $_{0.079}$	0.029 $_{0.085}$	0.087 $_{0.022}$	0.141 $_{0.088}$
		Oversampling	*f*	0.536 $_{0.017}$	0.532 $_{0.012}$	0.607 $_{0.084}$	0.658 $_{0.112}$
			m	0.532 $_{0.011}$	0.523 $_{0.020}$	0.548 $_{0.079}$	0.569 $_{0.059}$
			d	0.004 $_{0.028}$	0.009 $_{0.033}$	0.060 $_{0.006}$	0.088 $_{0.053}$
		Undersampling	*f*	0.548 $_{0.034}$	0.547 $_{0.033}$	0.571 $_{0.034}$	0.632 $_{0.074}$
			m	0.556 $_{0.000}$	0.555 $_{0.000}$	0.444 $_{0.000}$	0.500 $_{0.000}$
			d	-0.008 $_{0.034}$	-0.008 $_{0.033}$	0.127 $_{0.034}$	0.132 $_{0.074}$
RF		Regular	*f*	0.581 $_{0.024}$	0.580 $_{0.025}$	0.519 $_{0.028}$	0.611 $_{0.054}$
			m	0.600 $_{0.019}$	0.588 $_{0.020}$	0.597 $_{0.022}$	0.672 $_{0.021}$
			d	-0.019 $_{0.016}$	-0.008 $_{0.016}$	-0.078 $_{0.044}$	-0.062 $_{0.061}$
		Oversampling	*f*	0.576 $_{0.018}$	0.575 $_{0.018}$	0.505 $_{0.018}$	0.589 $_{0.021}$
			m	0.568 $_{0.032}$	0.552 $_{0.034}$	0.616 $_{0.025}$	0.661 $_{0.037}$
			d	0.008 $_{0.034}$	0.023 $_{0.035}$	-0.111 $_{0.013}$	-0.072 $_{0.041}$
		Undersampling	*f*	0.586 $_{0.024}$	0.585 $_{0.024}$	0.533 $_{0.024}$	0.632 $_{0.047}$
			m	0.575 $_{0.031}$	0.555 $_{0.037}$	0.635 $_{0.033}$	0.683 $_{0.022}$
			d	0.011 $_{0.024}$	0.031 $_{0.031}$	-0.102 $_{0.041}$	-0.052 $_{0.039}$
NN		Regular	*f*	0.533 $_{0.024}$	0.533 $_{0.024}$	0.519 $_{0.028}$	0.558 $_{0.026}$
			m	0.556 $_{0.017}$	0.544 $_{0.017}$	0.584 $_{0.012}$	0.622 $_{0.022}$
			d	-0.022 $_{0.037}$	-0.012 $_{0.036}$	-0.065 $_{0.031}$	-0.064 $_{0.026}$
		Oversampling	*f*	0.548 $_{0.040}$	0.547 $_{0.040}$	0.552 $_{0.028}$	0.611 $_{0.026}$
			m	0.562 $_{0.026}$	0.547 $_{0.024}$	0.603 $_{0.048}$	0.644 $_{0.057}$
			d	-0.014 $_{0.037}$	0.000 $_{0.035}$	-0.051 $_{0.061}$	-0.034 $_{0.065}$
		Undersampling	*f*	0.529 $_{0.049}$	0.524 $_{0.047}$	0.467 $_{0.051}$	0.495 $_{0.042}$
			m	0.495 $_{0.025}$	0.490 $_{0.023}$	0.524 $_{0.047}$	0.517 $_{0.054}$
			d	0.033 $_{0.063}$	0.035 $_{0.059}$	-0.057 $_{0.033}$	-0.022 $_{0.061}$

Table 2: **Fairness evaluation for German Credit dataset**. This table depicts the evaluation of accuracy (ACC), F1 score (F1), statistical parity (SP), and equality of opportunity (EoO) metrics for the subgroup - *female* (*f*) and *male* (*m*) as well as the difference (*d*) between them.

learning, we not only consider the regular in-context learning approach as detailed in Section 5.1, but we also experiment by flipping the labels of the few-shot examples to further examine their effect on the biases, as discussed in Section 5.2.

Again, for robustness, each experiment is conducted 5 times, with the mean and standard deviation reported.

5.1 Regular In-Context Learning

Previous works have demonstrated that LLMs can learn the input-label mappings in context (Akyürek et al., 2022; Xie et al., 2022; Von Oswald et al., 2023). However, the influence of in-context learning on the fairness has not been thoroughly examined. For in-context learning, the test example and task description, along with a few-shot examples, are provided to the LLMs for generating the final predictions. The few-shot examples are inserted before the test example in the prompt, as outlined in Section 3.3. We set the number of in-context examples as 50. For each dataset, we randomly select the in-context examples from the training set for each test example.

In Tables 1-3, we demonstrate that for two of the evaluated datasets (except for COMPAS), the incorporation of few-shot examples brings about performance improvements. Additionally, we observe that incorporating few-shot examples into prompting reduces the fairness metric gap between

				ACC	F1	SP	EoO
GPT-3.5-turbo	Zero-Shot		*AA*	0.657 $_{0.005}$	0.656 $_{0.004}$	0.395 $_{0.001}$	0.560 $_{0.002}$
			nAA	0.663 $_{0.002}$	0.588 $_{0.003}$	0.817 $_{0.002}$	0.893 $_{0.001}$
			d	-0.006 $_{0.005}$	0.068 $_{0.006}$	-0.423 $_{0.003}$	-0.334 $_{0.002}$
	Few-shot	Regular	*AA*	0.633 $_{0.002}$	0.626 $_{0.002}$	0.362 $_{0.003}$	0.495 $_{0.004}$
			nAA	0.642 $_{0.001}$	0.623 $_{0.002}$	0.614 $_{0.002}$	0.709 $_{0.002}$
			d	-0.008 $_{0.003}$	0.003 $_{0.003}$	-0.252 $_{0.003}$ ↓	-0.214 $_{0.005}$ ↓
		Label-flipping	*AA*	0.482 $_{0.004}$	0.482 $_{0.004}$	0.499 $_{0.003}$	0.481 $_{0.004}$
			nAA	0.412 $_{0.003}$	0.408 $_{0.003}$	0.471 $_{0.002}$	0.404 $_{0.003}$
			d	0.070 $_{0.005}$	0.074 $_{0.005}$	0.028 $_{0.005}$ ✓	0.077 $_{0.007}$ ✓
	Finetuning	Regular	*AA*	0.611 $_{0.016}$	0.610 $_{0.016}$	0.464 $_{0.031}$	0.576 $_{0.034}$
			nAA	0.616 $_{0.013}$	0.586 $_{0.016}$	0.657 $_{0.032}$	0.724 $_{0.029}$
			d	-0.005 $_{0.017}$	0.024 $_{0.024}$	-0.193 $_{0.030}$ ↓	-0.148 $_{0.027}$ ↓
		Oversampling	*AA*	0.609 $_{0.007}$	0.608 $_{0.007}$	0.494 $_{0.071}$	0.605 $_{0.066}$
			nAA	0.625 $_{0.020}$	0.583 $_{0.024}$	0.706 $_{0.037}$	0.771 $_{0.036}$
			d	-0.016 $_{0.016}$	0.025 $_{0.018}$	-0.212 $_{0.037}$	-0.166 $_{0.046}$
		Undersampling	*AA*	0.591 $_{0.010}$	0.591 $_{0.012}$	0.513 $_{0.053}$	0.605 $_{0.047}$
			nAA	0.641 $_{0.008}$	0.612 $_{0.009}$	0.663 $_{0.035}$	0.749 $_{0.037}$
			d	-0.050 $_{0.016}$	-0.021 $_{0.022}$	-0.150 $_{0.033}$	-0.144 $_{0.039}$
RF		Regular	*AA*	0.662 $_{0.004}$	0.662 $_{0.004}$	0.496 $_{0.006}$	0.660 $_{0.007}$
			nAA	0.671 $_{0.004}$	0.617 $_{0.002}$	0.767 $_{0.008}$	0.859 $_{0.009}$
			d	-0.009 $_{0.007}$	0.045 $_{0.005}$	-0.271 $_{0.011}$	-0.199 $_{0.014}$
		Oversampling	*AA*	0.660 $_{0.005}$	0.660 $_{0.005}$	0.493 $_{0.010}$	0.655 $_{0.013}$
			nAA	0.671 $_{0.002}$	0.624 $_{0.002}$	0.743 $_{0.003}$	0.839 $_{0.004}$
			d	-0.010 $_{0.006}$	0.037 $_{0.006}$	-0.250 $_{0.012}$	-0.184 $_{0.016}$
		Undersampling	*AA*	0.648 $_{0.002}$	0.647 $_{0.002}$	0.491 $_{0.004}$	0.639 $_{0.004}$
			nAA	0.667 $_{0.005}$	0.614 $_{0.007}$	0.761 $_{0.006}$	0.851 $_{0.006}$
			d	-0.020 $_{0.007}$	0.033 $_{0.008}$	-0.270 $_{0.009}$	-0.211 $_{0.008}$
NN		Regular	*AA*	0.666 $_{0.003}$	0.665 $_{0.002}$	0.462 $_{0.034}$	0.630 $_{0.034}$
			nAA	0.662 $_{0.003}$	0.613 $_{0.006}$	0.742 $_{0.019}$	0.831 $_{0.017}$
			d	0.005 $_{0.006}$	0.052 $_{0.007}$	-0.280 $_{0.019}$	-0.201 $_{0.018}$
		Oversampling	*AA*	0.656 $_{0.001}$	0.653 $_{0.012}$	0.507 $_{0.090}$	0.665 $_{0.101}$
			nAA	0.643 $_{0.013}$	0.580 $_{0.034}$	0.757 $_{0.107}$	0.828 $_{0.091}$
			d	0.013 $_{0.014}$	0.073 $_{0.043}$	-0.249 $_{0.049}$	-0.163 $_{0.046}$
		Undersampling	*AA*	0.660 $_{0.019}$	0.657 $_{0.023}$	0.477 $_{0.078}$	0.638 $_{0.097}$
			nAA	0.657 $_{0.013}$	0.602 $_{0.026}$	0.757 $_{0.051}$	0.839 $_{0.040}$
			d	0.003 $_{0.024}$	0.055 $_{0.043}$	-0.280 $_{0.041}$	-0.202 $_{0.064}$

Table 3: **Fairness evaluation for COMPAS dataset** for the subgroup - *African American (AA)*, and *Non African American* (*nAA*) as well as the difference (*d*). The significant fairness disparities are highlighted in red. Both in-context learning and finetuning can lead to bias reduction (indicated by ↓), and label-flipped in-context learning can further minimize bias (indicated by ✓).

different subgroups. However, a significant fairness issue still persists. Moreover, for the Adult and COMPAS datasets, the disparity in fairness metrics of in-context learning is more notable when compared to traditional models, such as RF and NN. This highlights the inherent biases embedded within LLMs, which are not solely derived from the task datasets.

5.2 Label Flipping

To delve deeper into the sources of biases within LLMs, we further examine the impact of the labels of in-context examples on fairness. As depicted in Tables 1-3, label flipping significantly reduces biases across all evaluated datasets. And for all evaluated datasets, the difference in statistical parity (SP) and equality of opportunity (EoO) is minimized with label-flipped in-context learning. For example, the absolute gap of EoO on the Adult dataset decreases from 0.483 in zero-shot prompting to 0.037, almost completely eliminating the bias. These findings further corroborates the existence of inherent biases in LLMs.

However, flipped labels lead to a significant drop in predictive performance. Though previous research suggests that the effectiveness of in-context learning predominantly stems from semantic priors, rather than learning the input-label mappings (Min et al., 2022; Wei et al., 2023b) and demonstrate that the performance of in-context learning is barely affected even with flipped or random labels for in-context examples, the focus of these works lies mainly on traditional natural language processing tasks.

In contrast, we observe that the labels of in-context examples hold substantial influence over predictive performance in our unique setup, where LLMs are deployed for predictions on tabular data. This could be attributed to the limited exposure of these models to tabular data during pre-training, thereby amplifying the role of input-label mapping of in-context examples.

6 Finetuning for Tabular Data

6.1 Regular Finetuning

Finally, we extend our investigation to assess if finetuning the models on the entire training set could aid in diminishing the social biases in LLMs. For GPT-3.5, fine-tuning is executed using the publicly released API from OpenAI. For RF and NN, we provide the training details in Appendix B. We still run all the experiments 5 times and compute the mean and standard deviation.

In Tables 1-3, we show that finetuning effectively reduces unfairness in all datasets, making them comparable and sometimes significantly better in terms of SP and EoO when compared to RF and NN. For example, the absolute difference in EoO after finetuning on Adult dataset is 0.0714, which is lower than 0.123 difference of a NN.

6.2 Resampling

We further explore the potential of resampling, a method frequently employed to enhance fairness in machine learning model training, particularly in scenarios where there is a significant class imbalance or bias in the data. To this end, we evaluate two approaches: oversampling the minority group and undersampling the majority group. As depicted in Tables 1-3, resampling fails to mitigate the social biases in LLMs when making tabular predictions, even though we demonstrate that oversampling generally reduces social biases for both RF and NN, except for a few instances such as, oversampling in NN for adult dataset worsens the fairness.

Our finetuning experiments show that the social biases inherited from LLM's pre-training data which are evident when making predictions on tabular data, can sometimes be mitigated through finetuning. Nevertheless, unlike the consistent outcomes typically seen in traditional machine learning models, like RF and NN, data resampling does not consistently produce similar results for finetuning LLMs.

7 Conclusion

In this work, we thoroughly investigate the under-explored problem of fairness of large language models (LLMs) for tabular tasks. Our study unfolds in several phases. Initially, we assess the inherent fairness displayed by LLMs, comparing their performance in zero-shot learning scenarios against traditional machine learning models like random forests (RF) and shallow neural networks (NN). Furthermore, we investigate how LLMs learn and propagate social biases when subjected to few-shot in-context learning, label-flipped in-context learning, fine-tuning, and data resampling techniques.

Our discoveries shed light on several key insights. We find that LLMs tend to heavily rely on the social biases inherited from their pre-training data when making predictions, which is a concerning issue. Moreover, we observe that few-shot in-context learning can partially mitigate the inherent biases in LLMs, yet it cannot entirely eliminate them. A significant fairness metric gap between different subgroups persists, and exceeds that observed in RF and NN. This observation underscores the existence of biases within the LLMs themselves, beyond just the task datasets. Additionally, label-flipping applied to the few-shot examples effectively reverses the effects of bias, again corroborating the existence of inherent biases in LLMs. However, as expected, this leads to a loss in predictive performance. Besides, our work reveals that while fine-tuning can sometimes improve the fairness of LLMs, data resampling does not consistently yield the same results, unlike what is typically observed in traditional machine learning models.

References

Abubakar Abid, Maheen Farooqi, and James Zou. Large language models associate muslims with violence. *Nature Machine Intelligence*, 3(6):461–463, 2021a. doi: 10.1038/s42256-021-00359-2. URL https://doi.org/10.1038/s42256-021-00359-2.

Abubakar Abid, Maheen Farooqi, and James Zou. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, pp. 298–306, New York, NY, USA, 2021b. Association for Computing Machinery. ISBN 9781450384735. doi: 10.1145/3461702.3462624.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021.

Christine Basta, Marta R. Costa-jussà, and Noe Casas. Evaluating the underlying gender bias in contextualized word embeddings. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, Sep 2019. doi: 10.18653/v1/w19-3805. URL http://dx.doi.org/10.18653/v1/w19-3805.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, October 2018. URL https://arxiv.org/abs/1810.01943.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Mar 2021. doi: 10.1145/3442188.3445922. URL http://dx.doi.org/10.1145/3442188.3445922.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f

R Bommasani. Opportunities and risks of foundation models, 2021. https://openai.com/reports/foundation-models/.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022. doi: 10.1109/TNNLS.2022.3229161.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In

H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8a

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017. doi: 10.1126/science.aal4230. URL https://www.science.org/doi/abs/10.1126/science.aal4230.

D. Dua and C. Graff. UCI machine learning repository, 2019. URL http://archive.ics.uci.edu/ml.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, Sheer El Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Scott Johnston, Andy Jones, Nicholas Joseph, Jackson Kernian, Shauna Kravec, Ben Mann, Neel Nanda, Kamal Ndousse, Catherine Olsson, Daniela Amodei, Tom Brown, Jared Kaplan, Sam McCandlish, Christopher Olah, Dario Amodei, and Jack Clark. Predictability and surprise in large generative models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pp. 1747–1764, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533229. URL https://doi.org/10.1145/3531146.3533229.

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=Fp7__phQszn.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, pp. 5549–5581. PMLR, 2023.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm, 2016. URL https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deelio-1.10. URL https://aclanthology.org/2022.deelio-1.10.

Yanchen Liu, William Held, and Diyi Yang. Dada: Dialect adaptation via dynamic aggregation of linguistic rules, 2023a.

Yanchen Liu, Timo Schick, and Hinrich Schtze. Semantic-oriented unlabeled priming for large-scale language models. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pp. 32–38, Toronto, Canada (Hybrid), July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.sustainlp-1.2. URL https://aclanthology.org/2023.sustainlp-1.2.

Li Lucy and David Bamman. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Third Workshop on Narrative Understanding*, pp. 48–55, Virtual, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.nuse-1.5. URL https://aclanthology.org/2021.nuse-1.5.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2304–2317, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.129. URL https://aclanthology.org/2023.acl-long.129.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL https://aclanthology.org/2022.emnlp-main.759.

OpenAI. Gpt-4 technical report, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. In *8th ICML Workshop on Automated Machine Learning (AutoML)*, 2021. URL https://openreview.net/forum?id=vdgtepS1pV.

Dylan Slack and Sameer Singh. Tablet: Learning from instructions for tabular data. *arXiv*, 2023.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, et al. Symbol tuning improves in-context learning in language models. *arXiv preprint arXiv:2305.08298*, 2023a.

Jerry W. Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. *ArXiv*, abs/2303.03846, 2023b. URL https://api.semanticscholar.org/CorpusID:257378479.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models, 2021.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=RdJVFCHjUMI.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2979–2989, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1323. URL https://aclanthology.org/D17-1323.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.

Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta, and Diyi Yang. Multi-VALUE: A framework for cross-dialectal English NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 744–768, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.44. URL https://aclanthology.org/2023.acl-long.44.

A Description for each feature in each dataset

We provide a detailed description of each feature from the datasets evaluated in our paper.

A.1 Adult

The original Adult Income Dataset contains 14 features an the target *Income*, as described in Table 4. Following prior work (Slack & Singh, 2023), we omit *Education-Num* and *Fnlwgt* as they are not crucial for income prediction, along with *Race* and *Native-Country*, to center our attention on *Sex* as the protected attribute.

Feature	**Type**	**Description**
Age	Continuous	Represents the age of an individual.
Workclass	Categorical	Indicates the type of employment, such as private, self-employed, or government.
Fnlwgt	Continuous	Stands for "final weight" and is a numerical value used in sampling for survey data.
Education	Categorical	Specifies the highest level of education attained by the individual, such as high school, bachelor's degree, etc.
Education-Num	Continuous	Represents the numerical equivalent of the education level.
Marital-Status	Categorical	Describes the marital status of the individual, including categories like married, divorced, or single.
Occupation	Categorical	Indicates the occupation of the individual, such as managerial, technical, or clerical work.
Relationship	Categorical	Specifies the individual's role in the family, such as husband, wife, or child.
Race	Categorical	Represents the individual's race or ethnic background.
Sex	Categorical	Indicates the gender of the individual, either male or female.
Capital-Gain	Continuous	Refers to the capital gains, which are profits from the sale of assets, of the individual.
Capital-Loss	Continuous	Represents the capital losses, which are losses from the sale of assets, of the individual.
Hours-Per-Week	Continuous	Denotes the number of hours worked per week by the individual.
Native-Country	Categorical	Specifies the native country or place of origin of the individual.
Income (target)	Binary	The target variable indicating whether an individual's income exceeds a certain threshold, typically $50,000 per year.

Table 4: Features in the original **Adult** dataset. Those not used in our work are shown in *italics*.

A.2 German Credit

The original German Credit Dataset contains 20 features, as detailed in Table 5. For simplicity and consistency with prior work, only the features not shown in *italics* are retained in our work. Furthermore, we extract *Sex* as an additional protected attribute from the *Personal Status and Sex* feature.

A.3 COMPAS

The raw COMPAS Recidivism dataset contains more than 50 attributes. Following the approach of Larson et al. (2016), we carry out the necessary preprocessing. More specifically, we group the *race* attribute into *African-American* and *Not African-American*, and consider only the features *sex*, *race*, *age*, *charge degree*, *priors count*, *risk*, and *two-year recid* (target). We frame the task as predicting

Feature	Type	Description
Credit Amount	Continuous	The amount of credit requested by the applicant.
Duration	Continuous	The duration of the credit in months.
Installment Rate	Ordinal	The installment rate in percentage of disposable income.
Residence Since	Ordinal	The number of years the applicant has lived at their current residence.
Age	Continuous	The age of the applicant.
Number of Existing Credits	Ordinal	The number of existing credits at this bank.
Number of Dependents	Ordinal	The number of dependents of the applicant.
Checking Account Status	Categorical	The status of the applicant's checking account, such as "no checking, "<0 DM," "0-200 DM," or "no known checking."
Credit History	Categorical	The credit history of the applicant, including categories like "critical/other existing credit," "existing paid," "delayed previously," etc.
Purpose	Categorical	The purpose of the credit, such as "radio/tv," "education," "new car," etc.
Savings Account	Categorical	The status of the applicant's savings account/bonds, including categories like "unknown/none," "<100 DM," "500-1000 DM," etc.
Employment Since	Categorical	The duration of the applicant's current employment, such as "unemployed," "<1 year," "4-7 years," etc.
Personal Status and Sex	Categorical	The personal status and sex of the applicant, including categories like "male single," "female div/dep/mar," etc.
Other Debtors/Guarantors	Categorical	Indicates the presence of other debtors/guarantors, such as "none," "guarantor," "co applicant."
Property	Categorical	Describes the type of property owned by the applicant, such as "real estate," "life insurance," "car or other," etc.
Other Installment Plans	Categorical	The presence of other installment plans.
Housing	Categorical	The housing situation of the applicant, such as "own," "for free," "rent."
Job	Categorical	The type of job held by the applicant, including categories like "skilled," "unskilled resident," "high qualif/self emp/mgmt," etc.
Telephone	Binary	Indicates whether the applicant has a telephone (yes/no).
Foreign Worker	Binary	Indicates whether the applicant is a foreign worker (yes/no).
Risk (target)	Binary	The target variable indicating credit risk (good/bad).

Table 5: Features in the original **German Credit** dataset. Those not used in our work are shown in *italics*. Additionally, from the original feature *Personal Status and Sex*, we extract *Sex* as a protected attribute.

whether an individual will recidivate within two years (*Did Not Reoffend* or *Reoffended*), based on their demographic and criminal history. Due to page limitations, we provide descriptions for only the features used in our work in Table 6.

B RF AND NN HYPERPARAMETERS

For RF, we fix number of trees to 100 for all datasets as well as models. For NN, we use a 3 hidden-layered network with hyperparameters described in Table 7.

Feature	Type	Description
Sex	Categorical	The gender of the individual.
Race	Categorical	The race of the individual, grouped into *African-American* and *Not African-American*.
Age	Continuous	The age of the individual.
Charge Degree	Categorical	The degree of the charge against the individual.
Priors Count	Continuous	The number of prior convictions or charges.
Risk	Categorical	The risk assessment for recidivism.
Two-Year Recid (target)	Binary	The target variable indicating whether an individual recidivated within two years.

Table 6: Features in the **COMPAS** Recidivism Dataset (Preprocessed).

	h1	h2	h3	lr	batch size	epochs
Adult	16	64	16	0.07	128	300
German Credit	64	64	32	0.07	128	300
COMPAS	64	128	64	0.09	128	300

Table 7: Hyperparameters for all datasets for a 3 layer neural network, where h1, h2 and h3 represent the number of neurons in first, second and third hidden layers respectively, lr represents the learning rate, and is followed by the batch size and number of epochs the models are trained for.

C PROMPT TEMPLATES FOR EACH DATASET

Beyond the Adult dataset, we provide the serialization and prompt templates utilized in our work for the German Credit and COMPAS datasets here.

C.1 GERMAN CREDIT

```
Predict the credit risk of a German bank customer based on their profile
data. Answer with one of the following: bad | good.
Example 1 -
Age: 25 - 45
Sex: female
Job: highly skilled
Housing: rent
Saving accounts: little
Checking account: little
Credit amount: 2606
Duration: 21
Purpose: radio/TV
Answer: good

Age: 25 - 45
Sex: male
Job: skilled
Housing: own
Saving accounts: little
Checking account: little
Credit amount: 1345
Duration: 18
Purpose: radio/TV
Answer:
```

Figure 2: Prompt Template for **German Credit** Dataset.

C.2 COMPAS

```
Predict whether an individual will recidivate with in two years based on
demographic and criminal history. Answer with one of the following: Did
Not Reoffend | Reoffended.
Example 1 -
sex: Male
race: African-American
age cat: 25 - 45
c charge degree: F
priors count: 0
risk: Low
Answer: Did Not Reoffend

sex: Male
race: African-American
age cat: 25 - 45
c charge degree: M
priors count: 13
risk: High
Answer:
```

Figure 3: Prompt Template for **COMPAS** Dataset.

Bibliography

[I] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “This looks more like that: Enhancing self-explaining models by prototypical relevance propagation.” *Pattern Recognition* 136 (2023), p. 109172.

[II] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation.” In: *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2022, pp. 1–5.

[III] Srishti Gautam, Ahcene Boubekki, Stine Hansen, Suaiba Salahuddin, Robert Jenssen, Marina Höhne, and Michael Kampffmeyer. “Protovae: A trustworthy self-explainable prototypical variational model.” *Advances in Neural Information Processing Systems* 35 (2022), pp. 17940–17952.

[IV] Srishti Gautam, Ahcene Boubekki, Marina Höhne, and Michael C Kampffmeyer. “Prototypical Self-Explainable Models Without Re-training.” *Under Review* (2023).

[V] Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. “Investigating the Fairness of Large Language Models for Predictions on Tabular Data.” *Under Review* (2023).

[6] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “This looks more like that: Enhancing self-explaining models by prototypical relevance propagation.” *National Conference on Image Processing and Machine Learning (NOBIM)* (2021). Extended abstract and oral presentation.

[7] Stine Hansen, Srishti Gautam, Robert Jenssen, and Michael Kampffmeyer. “Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels.” *National Conference on Image Processing and Machine Learning (NOBIM)* (2021). Extended abstract and oral presentation.

[8] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “Artifact Detection with Prototypical Relevance Propagation.” *Visual Intelligence Days* (2021). Oral presentation.

[9] Srishti Gautam. “Self-Explainability and Artifact detection: Along with applications to medical data.” *COMP-7950-T04 – Advanced Machine Learning Event, University of Manitoba, Canada* (2021). Invited Talk.

[10] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation.” *NORA Annual Conference* (2022). Extended abstract and oral presentation.

[11] Srishti Gautam, Marina M-C Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. “Demonstrating the risk of imbalanced datasets in chest x-ray image-based diagnostics by prototypical relevance propagation.” *Visual Intelligence Days* (2022). Poster presentation.

[12] Stine Hansen, Srishti Gautam, Robert Jenssen, and Michael Kampffmeyer. “Anomaly detection-inspired few-shot medical image segmentation through self-supervision with supervoxels.” *Medical Image Analysis* 78 (2022), p. 102385.

[13] Suaiba Amina Salahuddin, Stine Hansen, Srishti Gautam, Michael Kampffmeyer, and Robert Jenssen. “A self-guided anomaly detection-inspired few-shot segmentation network.” CEUR Workshop Proceedings. 2022.

[14] Srishti Gautam. “Bias in Machine Learning.” *Bias in Artificial Intelligence Workshop at UiT – The Arctic University of Norway* (2023). Invited Talk.

[15] Stine Hansen, Srishti Gautam, Suaiba Amina Salahuddin, Michael Kampffmeyer, and Robert Jenssen. “ADNet++: A few-shot learning framework for multi-class medical image volume segmentation with uncertainty-guided feature refinement.” *Medical Image Analysis* (2023), p. 102870.

[16] Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. “Investigating the Fairness of Large Language Models for Predictions on Tabular Data.” *NeurIPS Workshop on Socially Responsible Language Modelling Research* (2023).

[17] Michael I Jordan and Tom M Mitchell. “Machine learning: Trends, perspectives, and prospects.” *Science* 349.6245 (2015), pp. 255–260.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning.” *nature* 521.7553 (2015), pp. 436–444.

[19] Jürgen Schmidhuber. “Deep learning in neural networks: An overview.” *Neural Networks* 61 (2015), pp. 85–117.

[20] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. “Deep learning for computer vision: A brief review.” *Computational intelligence and neuroscience* 2018 (2018).

[21] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. “Deep learning-enabled medical computer vision.” *NPJ digital medicine* 4.1 (2021), p. 5.

[22] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: `2005.14165 [cs.CL]`.

[23] Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

[24] Pin Wang, En Fan, and Peng Wang. "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning." *Pattern Recognition Letters* 141 (2021), pp. 61–67.

[25] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[26] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. "Summary of chatgpt-related research and perspective towards the future of large language models." *Meta-Radiology* (2023), p. 100017.

[27] Dinggang Shen, Guorong Wu, and Heung-Il Suk. "Deep learning in medical image analysis." *Annual review of biomedical engineering* 19 (2017), pp. 221–248.

[28] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. "Deep learning for financial applications: A survey." *Applied Soft Computing* 93 (2020), p. 106384.

[29] Xuxin Chen, Ximin Wang, Ke Zhang, Kar-Ming Fung, Theresa C Thai, Kathleen Moore, Robert S Mannel, Hong Liu, Bin Zheng, and Yuchen Qiu. "Recent advances and clinical applications of deep learning in medical image analysis." *Medical Image Analysis* 79 (2022), p. 102444.

[30] Mingqing Wang, Qilin Zhang, Saikit Lam, Jing Cai, and Ruijie Yang. "A review on application of deep learning algorithms in external beam radiotherapy automated treatment planning." *Frontiers in oncology* 10 (2020), p. 580919.

[31] Rogelio A Mancisidor, Michael Kampffmeyer, Kjersti Aas, and Robert Jenssen. "Deep generative models for reject inference in credit scoring." *Knowledge-Based Systems* 196 (2020), p. 105758.

[32] Cuicui Luo, Desheng Wu, and Dexiang Wu. "A deep learning approach for credit scoring using credit default swaps." *Engineering Applications of Artificial Intelligence* 65 (2017), pp. 465–470.

[33] Thibaut Théate and Damien Ernst. "An application of deep reinforcement learning to algorithmic trading." *Expert Systems with Applications* 173 (2021), p. 114632.

[34] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. "A survey of deep learning techniques for autonomous driving." *Journal of Field Robotics* 37.3 (2020), pp. 362–386.

[35] B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. "Deep reinforcement learning for autonomous driving: A survey." *IEEE Transactions on Intelligent Transportation Systems* 23.6 (2021), pp. 4909–4926.

[36] Cynthia Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead." *Nature machine intelligence* 1.5 (2019), pp. 206–215.

[37] Julie Gerlings, Arisa Shollo, and Ioanna Constantiou. "Reviewing the need for explainable artificial intelligence (xAI)." *arXiv preprint arXiv:2012.01007* (2020).

[38] Thomas Hellström, Virginia Dignum, and Suna Bensch. "Bias in Machine Learning–What is it Good for?" *arXiv preprint arXiv:2004.00686* (2020).

[39] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. "A survey on bias and fairness in machine learning." *ACM computing surveys (CSUR)* 54.6 (2021), pp. 1–35.

[40] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. "Trustworthy AI: From principles to practices." *ACM Computing Surveys* 55.9 (2023), pp. 1–46.

[41] Leon Sixt, Maximilian Granz, and Tim Landgraf. "When explanations lie: Why many modified bp attributions fail." In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9046–9057.

[42] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. "This looks like that: deep learning for interpretable image recognition." *Advances in neural information processing systems* 32 (2019).

[43] Oskar Pfungst. *Clever Hans:(the horse of Mr. Von Osten.) a contribution to experimental animal and human psychology*. Holt, Rinehart and Winston, 1911.

[44] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Unmasking Clever Hans predictors and assessing what machines really learn." *Nature communications* 10.1 (2019), p. 1096.

[45] Christopher J Anders, Talmaj Marinc, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. "Analyzing imagenet with spectral relevance analysis: Towards imagenet un-hans' ed." *arXiv preprint arXiv:1912.11425* 2.3 (2019).

[46] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. "Backdoor attacks and countermeasures on deep learning: A comprehensive review." *arXiv preprint arXiv:2007.10760* (2020).

[47] Naveed Akhtar and Ajmal Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey." *Ieee Access* 6 (2018), pp. 14410–14430.

[48] Roxana Daneshjou, Mary P Smith, Mary D Sun, Veronica Rotemberg, and James Zou. "Lack of transparency and potential bias in artificial intelligence data sets and algorithms: a scoping review." *JAMA dermatology* 157.11 (2021), pp. 1362–1369.

[49] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. "Bias and fairness in large language models: A survey." *arXiv preprint arXiv:2309.00770* (2023).

[50] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders." *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[51] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[52] Igor Kononenko. "Machine learning for medical diagnosis: history, state of the art and perspective." *Artificial Intelligence in medicine* 23.1 (2001), pp. 89–109.

[53] Hyunseok Seo, Masoud Badiei Khuzani, Varun Vasudevan, Charles Huang, Hongyi Ren, Ruoxiu Xiao, Xiao Jia, and Lei Xing. "Machine learning techniques for biomedical image segmentation: an overview of technical aspects and introduction to state-of-art applications." *Medical physics* 47.5 (2020), e148–e167.

[54] Ronald A Fisher. "The use of multiple measurements in taxonomic problems." *Annals of eugenics* 7.2 (1936), pp. 179–188.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[56] C Lakshmi Devasena, T Sumathi, VV Gomathi, and M Hemalatha. "Effectiveness evaluation of rule based classifiers for the classification of iris data set." *Bonfring International Journal of Man Machine Interface* 1 (2011), p. 5.

[57] David MW Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." *arXiv preprint arXiv:2010.16061* (2020).

[58] Sotiris B Kotsiantis, Ioannis Zaharakis, P Pintelas, et al. "Supervised machine learning: A review of classification techniques." *Emerging artificial intelligence applications in computer engineering* 160.1 (2007), pp. 3–24.

[59] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. "Machine learning: a review of classification and combining techniques." *Artificial Intelligence Review* 26 (2006), pp. 159–190.

[60] Dastan Maulud and Adnan M Abdulazeez. “A review on linear regression comprehensive in machine learning.” *Journal of Applied Science and Technology Trends* 1.4 (2020), pp. 140–147.

[61] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. “A general framework for object detection.” In: *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE. 1998, pp. 555–562.

[62] Li Deng and Xiao Li. “Machine learning paradigms for speech recognition: An overview.” *IEEE Transactions on Audio, Speech, and Language Processing* 21.5 (2013), pp. 1060–1089.

[63] Thiago S Guzella and Walmir M Caminhas. “A review of machine learning approaches to spam filtering.” *Expert Systems with Applications* 36.7 (2009), pp. 10206–10222.

[64] Pin Wang, En Fan, and Peng Wang. “Comparative analysis of image classification algorithms based on traditional machine learning and deep learning.” *Pattern Recognition Letters* 141 (2021), pp. 61–67.

[65] Fionn Murtagh. “Multilayer perceptrons for classification and regression.” *Neurocomputing* 2.5-6 (1991), pp. 183–197.

[66] Corinna Cortes and Vladimir Vapnik. “Support-vector networks.” *Machine learning* 20.3 (1995), pp. 273–297.

[67] Lior Rokach and Oded Maimon. “Decision trees.” *Data mining and knowledge discovery handbook* (2005), pp. 165–192.

[68] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. “Activation functions: Comparison of trends in practice and research for deep learning.” *arXiv preprint arXiv:1811.03378* (2018).

[69] Shun-ichi Amari. “Backpropagation and stochastic gradient descent method.” *Neurocomputing* 5.4-5 (1993), pp. 185–196.

[70] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. “A comprehensive survey of loss functions in machine learning.” *Annals of Data Science* (2020), pp. 1–26.

[71] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980* (2014).

[72] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. “Assessing the accuracy of prediction algorithms for classification: an overview.” *Bioinformatics* 16.5 (2000), pp. 412–424.

[73] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, and Jerome Friedman. “Unsupervised learning.” *The elements of statistical learning: Data mining, inference, and prediction* (2009), pp. 485–585.

[74] Ramadass Sathya, Annamma Abraham, et al. “Comparison of supervised and unsupervised learning algorithms for pattern classification.” *International Journal of Advanced Research in Artificial Intelligence* 2.2 (2013), pp. 34–38.

[75] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. “Data clustering: a review.” *ACM computing surveys (CSUR)* 31.3 (1999), pp. 264–323.

[76] Laurens Van Der Maaten, Eric O Postma, H Jaap van den Herik, et al. “Dimensionality reduction: A comparative review.” *Journal of Machine Learning Research* 10.66-71 (2009), p. 13.

[77] Mohamed Alloghani, Dhiya Al-Jumeily, Jamila Mustafina, Abir Hussain, and Ahmed J. Aljaaf. “A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science.” In: *Supervised and Unsupervised Learning for Data Science*. Ed. by Michael W. Berry, Azlinah Mohamed, and Bee Wah Yap. Cham: Springer International Publishing, 2020, pp. 3–21.

[78] James MacQueen et al. “Some methods for classification and analysis of multivariate observations.” In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[79] Ulrike Von Luxburg. “A tutorial on spectral clustering.” *Statistics and computing* 17 (2007), pp. 395–416.

[80] Yan Yang and Hao Wang. “Multi-view clustering: A survey.” *Big Data Mining and Analytics* 1.2 (2018), pp. 83–107.

[81] Stuart Lloyd. “Least squares quantization in PCM.” *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[82] Abhishek Kumar and Hal Daumé. “A co-training approach for multi-view spectral clustering.” In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 393–400.

[83] Man-Sheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. “Multi-view clustering in latent embedding space.” In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 34. 04. 2020, pp. 3513–3520.

[84] Bin Zhao, James T Kwok, and Changshui Zhang. “Multiple kernel clustering.” In: *Proceedings of the 2009 SIAM international conference on data mining*. SIAM. 2009, pp. 638–649.

[85] Zhao Kang, Xiao Lu, Jinfeng Yi, and Zenglin Xu. “Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification.” *arXiv preprint arXiv:1806.07697* (2018).

[86] Jiyuan Liu, Xinwang Liu, Yuexiang Yang, Xifeng Guo, Marius Kloft, and Liangzhong He. “Multiview subspace clustering via co-training robust data representation.” *IEEE Transactions on Neural Networks and Learning Systems* 33.10 (2021), pp. 5177–5189.

[87] Xiaochun Cao, Changqing Zhang, Huazhu Fu, Si Liu, and Hua Zhang. “Diversity-induced multi-view subspace clustering.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 586–594.

[88] Daniel J Trosten, Sigurd Lokse, Robert Jenssen, and Michael Kampffmeyer. “Reconsidering representation alignment for multi-view clus-

tering." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 1255–1265.

[89] Ahcène Boubekki, Michael Kampffmeyer, Ulf Brefeld, and Robert Jenssen. "Joint optimization of an autoencoder for clustering and embedding." *Machine Learning* 110.7 (2021), pp. 1901–1937.

[90] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, Eftychios Protopapadakis, et al. "Deep learning for computer vision: A brief review." *Computational intelligence and neuroscience* 2018 (2018).

[91] Monica Bianchini and Franco Scarselli. "On the complexity of neural network classifiers: A comparison between shallow and deep architectures." *IEEE transactions on neural networks and learning systems* 25.8 (2014), pp. 1553–1565.

[92] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. "A survey of convolutional neural networks: analysis, applications, and prospects." *IEEE transactions on neural networks and learning systems* (2021).

[93] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012).

[94] Qing Li, Weidong Cai, Xiaogang Wang, Yun Zhou, David Dagan Feng, and Mei Chen. "Medical image classification with convolutional neural network." In: *2014 13th international conference on control automation robotics & vision (ICARCV)*. IEEE. 2014, pp. 844–848.

[95] Anamika Dhillon and Gyanendra K Verma. "Convolutional neural network: a review of models, methodologies and applications to object detection." *Progress in Artificial Intelligence* 9.2 (2020), pp. 85–112.

[96] Dominik Scherer, Andreas Müller, and Sven Behnke. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition." In: *Artificial Neural Networks – ICANN 2010*. Ed. by Konstantinos Diamantaras, Wlodek Duch, and Lazaros S. Iliadis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 92–101.

[97] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In: *International conference on machine learning*. pmlr. 2015, pp. 448–456.

[98] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[99] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. "A survey of the recent architectures of deep convolutional neural networks." *Artificial intelligence review* 53 (2020), pp. 5455–5516.

[100] Yann LeCun et al. "LeNet-5, convolutional neural networks." *URL: http://yann. lecun. com/exdb/lenet* 20.5 (2015), p. 14.

[101] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: `1409.1556 [cs.CV]`.

[102] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. *Going Deeper with Convolutions*. 2014. arXiv: `1409.4842 [cs.CV]`.

[103] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[104] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. *Densely Connected Convolutional Networks*. 2018. arXiv: `1608.06993 [cs.CV]`.

[105] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. "Variational Autoencoder for Deep Learning of Images, Labels and Captions." In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016.

[106] Arash Vahdat and Jan Kautz. "NVAE: A Deep Hierarchical Variational Autoencoder." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 19667–19679.

[107] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability." *Special lecture on IE* 2.1 (2015), pp. 1–18.

[108] Adrian Alan Pol, Victor Berger, Cecile Germain, Gianluca Cerminara, and Maurizio Pierini. "Anomaly detection with conditional variational autoencoders." In: *2019 18th IEEE international conference on machine learning and applications (ICMLA)*. IEEE. 2019, pp. 1651–1657.

[109] Dor Bank, Noam Koenigstein, and Raja Giryes. "Autoencoders." *Machine Learning for Data Science Handbook: Data Mining and Knowledge Discovery Handbook* (2023), pp. 353–374.

[110] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections." *Advances in neural information processing systems* 29 (2016).

[111] Junyuan Xie, Linli Xu, and Enhong Chen. "Image denoising and inpainting with deep neural networks." *Advances in neural information processing systems* 25 (2012).

[112] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning." *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312.

[113] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[114] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[115] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[116] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: `2010.11929 [cs.CV]`.

[117] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. *LLaMA: Open and Efficient Foundation Language Models*. 2023. arXiv: `2302.13971 [cs.CL]`.

[118] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: `2005.14165 [cs.CL]`.

[119] Dylan Slack and Sameer Singh. *TABLET: Learning From Instructions For Tabular Data*. 2023. arXiv: `2304.13188 [cs.LG]`.

[120] Erico Tjoa and Cuntai Guan. "A survey on explainable artificial intelligence (xai): Toward medical xai." *IEEE transactions on neural networks and learning systems* 32.11 (2020), pp. 4793–4813.

[121] Éloi Zablocki, Hédi Ben-Younes, Patrick Pérez, and Matthieu Cord. "Explainability of deep vision-based autonomous driving systems: Review and challenges." *International Journal of Computer Vision* 130.10 (2022), pp. 2425–2452.

[122] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. "Explainable artificial intelligence: an analytical review." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11.5 (2021), e1424.

[123] Wojciech Samek and Klaus-Robert Müller. "Towards explainable artificial intelligence." *Explainable AI: interpreting, explaining and visualizing deep learning* (2019), pp. 5–22.

[124] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-

Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.” *Information Fusion* 58 (2020), pp. 82–115.

[125] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.” *PloS one* 10.7 (2015), e0130140.

[126] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “" Why should i trust you?" Explaining the predictions of any classifier.” In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[127] David Alvarez Melis and Tommi Jaakkola. “Towards robust interpretability with self-explaining neural networks.” *Advances in neural information processing systems* 31 (2018).

[128] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. “A survey of methods for explaining black box models.” *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[129] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. “Feature Visualization.” *Distill* (2017).

[130] Tessa Han, Suraj Srinivas, and Himabindu Lakkaraju. “Which explanation should i choose? a function approximation perspective to characterizing post hoc explanations.” *Advances in Neural Information Processing Systems* 35 (2022), pp. 5256–5268.

[131] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. *Towards Faithful Model Explanation in NLP: A Survey*. 2023. arXiv: `2209.11326 [cs.CL]`.

[132] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. “A survey of methods for explaining black box models.” *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[133] Andrea Galassi, Marco Lippi, and Paolo Torroni. “Attention in natural language processing.” *IEEE transactions on neural networks and learning systems* 32.10 (2020), pp. 4291–4308.

[134] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions.” *Advances in neural information processing systems* 30 (2017).

[135] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. “Deep inside convolutional networks: Visualising image classification models and saliency maps.” *arXiv preprint arXiv:1312.6034* (2013).

[136] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. “Striving for simplicity: The all convolutional net.” *arXiv preprint arXiv:1412.6806* (2014).

[137] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. “Axiomatic attribution for deep networks.” In: *International conference on machine learning*. PMLR. 2017, pp. 3319–3328.

[138] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. “Learning deep features for discriminative localization.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[139] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. “Grad-cam: Visual explanations from deep networks via gradient-based localization.” In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[140] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks.” In: *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2018, pp. 839–847.

[141] Maximilian Kohlbrenner, Alexander Bauer, Shinichi Nakajima, Alexander Binder, Wojciech Samek, and Sebastian Lapuschkin. “Towards best practice in explaining neural network decisions with LRP.” In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–7.

[142] Riccardo Guidotti. “Counterfactual explanations and how to find them: literature review and benchmarking.” *Data Mining and Knowledge Discovery* (2022), pp. 1–55.

[143] Pang Wei Koh and Percy Liang. “Understanding black-box predictions via influence functions.” In: *International conference on machine learning*. PMLR. 2017, pp. 1885–1894.

[144] Seungeon Lee, Xiting Wang, Sungwon Han, Xiaoyuan Yi, Xing Xie, and Meeyoung Cha. “Self-explaining deep models with logic rule reasoning.” *Advances in Neural Information Processing Systems* 35 (2022), pp. 3203–3216.

[145] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. *AttentionViz: A Global View of Transformer Attention*. 2023. arXiv: `2305.03210 [cs.HC]`.

[146] Evelyn Fix and J. L. Hodges. “Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties.” *International Statistical Review / Revue Internationale de Statistique* 57.3 (1989), pp. 238–247.

[147] Aravindh Mahendran and Andrea Vedaldi. “Visualizing deep convolutional neural networks using natural pre-images.” *International Journal of Computer Vision* 120 (2016), pp. 233–255.

[148] Suraj Srinivas and François Fleuret. “Rethinking the role of gradient-based attribution methods for model interpretability.” *arXiv preprint arXiv:2006.09128* (2020).

[149] Jessica Dai, Sohini Upadhyay, Ulrich Aivodji, Stephen H Bach, and Himabindu Lakkaraju. “Fairness via explanation quality: Evaluating disparities in the quality of post hoc explanations.” In: *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*. 2022, pp. 203–214.

[150] Julius Adebayo, Michael Muelly, Harold Abelson, and Been Kim. “Post hoc explanations may be ineffective for detecting unknown spurious correlation.” In: *International conference on learning representations*. 2021.

[151] Anna Hedström, Leander Weber, Daniel Krakowczyk, Dilyara Bareeva, Franz Motzkus, Wojciech Samek, Sebastian Lapuschkin, and Marina M-C Höhne. “Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond.” *Journal of Machine Learning Research* 24.34 (2023), pp. 1–11.

[152] Trevor J Hastie. “Generalized additive models.” In: *Statistical models in S*. Routledge, 2017, pp. 249–307.

[153] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. “Neural additive models: Interpretable machine learning with neural nets.” *Advances in neural information processing systems* 34 (2021), pp. 4699–4711.

[154] Filip Radenovic, Abhimanyu Dubey, and Dhruv Mahajan. “Neural basis models for interpretability.” *Advances in Neural Information Processing Systems* 35 (2022), pp. 8414–8426.

[155] Jayneel Parekh, Pavlo Mozharovskyi, and Florence d’Alché-Buc. “A framework to learn with interpretation.” *Advances in Neural Information Processing Systems* 34 (2021), pp. 24273–24285.

[156] Yipei Wang and Xiaoqian Wang. “Self-interpretable model with transformation equivariant interpretation.” *Advances in Neural Information Processing Systems* 34 (2021), pp. 2359–2372.

[157] Eunji Kim, Siwon Kim, Minji Seo, and Sungroh Yoon. “XProtoNet: diagnosis in chest radiography with global and local explanations.” In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15719–15728.

[158] Jiaqi Wang, Huafeng Liu, Xinyue Wang, and Liping Jing. “Interpretable image recognition by constructing transparent embedding space.” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 895–904.

[159] Meike Nauta, Ron Van Bree, and Christin Seifert. “Neural prototype trees for interpretable fine-grained image recognition.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 14933–14943.

[160] Dawid Rymarczyk, Łukasz Struski, Michał Górszczak, Koryna Lewandowska, Jacek Tabor, and Bartosz Zieliński. “Interpretable image classification

with differentiable prototypes assignment." In: *European Conference on Computer Vision*. Springer. 2022, pp. 351–368.

[161] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. "Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification." In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 1420–1430.

[162] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. "Detecting backdoor attacks on deep neural networks by activation clustering." *arXiv preprint arXiv:1811.03728* (2018).

[163] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. "Robust physical-world attacks on deep learning visual classification." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1625–1634.

[164] John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric K Oermann. "Confounding variables can degrade generalization performance of radiological deep learning models." *arXiv preprint arXiv:1807.00431* (2018).

[165] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. "Towards understanding the spectral bias of deep learning." *arXiv preprint arXiv:1912.01198* (2019).

[166] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*.

[167] Cesar F Caiafa, Zhe Sun, Toshihisa Tanaka, Pere Marti-Puig, and Jordi Solé-Casals. *Machine learning methods with noisy, incomplete or small datasets*. 2021.

[168] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. "Learning from noisy labels with deep neural networks: A survey." *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[169] Yue Li, Hongxia Wang, and Mauro Barni. "A survey of deep neural network watermarking techniques." *Neurocomputing* 461 (2021), pp. 171–193.

[170] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey." *Neurocomputing* 312 (2018), pp. 135–153.

[171] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. "Targeted backdoor attacks on deep learning systems using data poisoning." *arXiv preprint arXiv:1712.05526* (2017).

[172] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. "Adversarial examples: Attacks and defenses for deep learning." *IEEE transactions on neural networks and learning systems* 30.9 (2019), pp. 2805–2824.

[173] Hyun Kwon. “Detecting backdoor attacks via class difference in deep neural networks.” *IEEE Access* 8 (2020), pp. 191049–191056.

[174] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. “Backdoor attacks against deep learning systems in the physical world.” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6206–6215.

[175] Susu Sun, Lisa M Koch, and Christian F Baumgartner. “Right for the Wrong Reason: Can Interpretable ML Techniques Detect Spurious Correlations?” In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2023, pp. 425–434.

[176] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. “From imagenet to image classification: Contextualizing progress on benchmarks.” In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9625–9635.

[177] Joy Buolamwini and Timnit Gebru. “Gender shades: Intersectional accuracy disparities in commercial gender classification.” In: *Conference on fairness, accountability and transparency*. PMLR. 2018, pp. 77–91.

[178] Gal Vardi. “On the implicit bias in deep-learning algorithms.” *Communications of the ACM* 66.6 (2023), pp. 86–93.

[179] Yuchi Tian, Ziyuan Zhong, Vicente Ordonez, Gail Kaiser, and Baishakhi Ray. “Testing DNN image classifiers for confusion & bias errors.” In: *Proceedings of the acm/ieee 42nd international conference on software engineering*. 2020, pp. 1122–1134.

[180] Mingcai Chen, Hao Cheng, Yuntao Du, Ming Xu, Wenyu Jiang, and Chongjun Wang. “Two wrongs don’t make a right: Combating confirmation bias in learning with label noise.” In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 12. 2023, pp. 14765–14773.

[181] Yanbing Mao, Emrah Akyol, and Naira Hovakimyan. “Impact of confirmation bias on competitive information spread in social networks.” *IEEE Transactions on Control of Network Systems* 8.2 (2021), pp. 816–827.

[182] Lachlan Urquhart and Diana Miranda. “Policing faces: the present and future of intelligent facial surveillance.” *Information & communications technology law* 31.2 (2022), pp. 194–219.

[183] Jeffrey Dastin. “Amazon scraps secret AI recruiting tool that showed bias against women.” In: *Ethics of data and analytics*. Auerbach Publications, 2022, pp. 296–299.

[184] Mutale Nkonde. “Automated anti-blackness: facial recognition in Brooklyn, New York.” *Harvard Journal of African American Public Policy* 20 (2019), pp. 30–36.

[185] Taylor Telford. “Apple Card algorithm sparks gender bias allegations against Goldman Sachs.” *Washington Post* 11 (2019).

[186] Gina Neff. "Talking to bots: Symbiotic agency and the case of Tay." *International Journal of Communication* (2016).

[187] Anna Arias-Duart, Ferran Parés, Dario Garcia-Gasulla, and Victor Giménez-Ábalos. "Focus! rating XAI methods and finding biases." In: *2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2022, pp. 1–8.

[188] Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. """ Kelly is a Warm Person, Joseph is a Role Model": Gender Biases in LLM-Generated Reference Letters." *arXiv preprint arXiv:2310.09219* (2023).

[189] Jesutofunmi A Omiye, Jenna Lester, Simon Spichak, Veronica Rotemberg, and Roxana Daneshjou. "Beyond the hype: large language models propagate race-based medicine." *medRxiv* (2023), pp. 2023–07.

[190] Pranav Narayanan Venkit, Sanjana Gautam, Ruchi Panchanadikar, Shomir Wilson, et al. "Nationality Bias in Text Generation." *arXiv preprint arXiv:2302.02463* (2023).

[191] Alberto Acerbi and Joseph M Stubbersfield. "Large language models show human-like content biases in transmission chain experiments." *Proceedings of the National Academy of Sciences* 120.44 (2023), e2313790120.

[192] Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. "Red teaming chatgpt via jailbreaking: Bias, robustness, reliability and toxicity." *arXiv preprint arXiv:2301.12867* (2023), pp. 12–2.

[193] Yunqi Li and Yongfeng Zhang. *Fairness of ChatGPT*. 2023. arXiv: `2305.18569 [cs.LG]`.

[194] Po-Sen Huang, Huan Zhang, Ray Jiang, Robert Stanforth, Johannes Welbl, Jack Rae, Vishal Maini, Dani Yogatama, and Pushmeet Kohli. *Reducing Sentiment Bias in Language Models via Counterfactual Evaluation*. 2020. arXiv: `1911.03064 [cs.CL]`.

[195] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: `2201.11903 [cs.CL]`.

[196] Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. *Post Hoc Explanations of Language Models Can Improve Language Models*. 2023. arXiv: `2305.11426 [cs.CL]`.

[197] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. "Deep Neural Networks and Tabular Data: A Survey." *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–21.

[198] Andreas Mogelmose, Mohan Manubhai Trivedi, and Thomas B. Moeslund. "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey." *IEEE Transactions on Intelligent Transportation Systems* 13.4 (2012), pp. 1484–1497.

[199] Hong-Yu Zhou, Shuang Yu, Cheng Bian, Yifan Hu, Kai Ma, and Yefeng Zheng. “Comparing to Learn: Surpassing ImageNet Pretraining on Radiographs by Comparing Image Representations.” In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*. Ed. by Anne L. Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A. Zuluaga, S. Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz. Cham: Springer International Publishing, 2020, pp. 398–407.

[200] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. “ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[201] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. 2019. arXiv: `1901.07031 [cs.CV]`.

[202] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. “UMAP: Uniform Manifold Approximation and Projection.” *Journal of Open Source Software* 3.29 (2018), p. 861.

[203] Li Deng. “The mnist database of handwritten digit images for machine learning research.” *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[204] Adam Coates, Andrew Ng, and Honglak Lee. “An Analysis of Single-Layer Networks in Unsupervised Feature Learning.” In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Geoffrey Gordon, David Dunson, and Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, Nov. 2011, pp. 215–223.

[205] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. “Deep Neural Networks and Tabular Data: A Survey.” *IEEE Transactions on Neural Networks and Learning Systems* (2022), pp. 1–21.

[206] Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. “Why do tree-based models still outperform deep learning on typical tabular data?” In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022.

[207] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri,

Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. *AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias*. Oct. 2018.

[208] Barry Becker and Ronny Kohavi. *Adult*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20. 1996.

[209] Tin Kam Ho. “Random decision forests.” In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282.

[210] Lei Cai, Hongyang Gao, and Shuiwang Ji. “Multi-stage variational auto-encoders for coarse-to-fine image generation.” In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM. 2019, pp. 630–638.

[211] Darius Chira, Ilian Haralampiev, Ole Winther, Andrea Dittadi, and Valentin Liévin. *Image Super-Resolution With Deep Variational Autoencoders*. 2022. arXiv: 2203.09445 [cs.CV].

[212] Rewon Child. “Very deep vaes generalize autoregressive models and can outperform them on images.” *arXiv preprint arXiv:2011.10650* (2020).

[213] Danilo Rezende and Shakir Mohamed. “Variational inference with normalizing flows.” In: *International conference on machine learning*. PMLR. 2015, pp. 1530–1538.

[214] BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. “Bloom: A 176b-parameter open-access multilingual language model.” *arXiv preprint arXiv:2211.05100* (2022).

[215] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. “Tree of thoughts: Deliberate problem solving with large language models.” *arXiv preprint arXiv:2305.10601* (2023).

[216] Xi Ye and Greg Durrett. “The Unreliability of Explanations in Few-shot Prompting for Textual Reasoning.” In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 30378–30392.

[217] Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. *Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting*. 2023. arXiv: 2305.04388 [cs.CL].

[218] Chirag Agarwal, Satyapriya Krishna, Eshika Saxena, Martin Pawelczyk, Nari Johnson, Isha Puri, Marinka Zitnik, and Himabindu Lakkaraju. “Openxai: Towards a transparent evaluation of model explanations.” *Advances in Neural Information Processing Systems* 35 (2022), pp. 15784–15799.