SURVEY

Ante-Hoc Methods for Interpretable Deep Models: A Survey

ANTONIO DI MARINO, Istituto Di Calcolo E Reti Ad Alte Prestazioni, Rende, Rende, CS, Italy

VINCENZO BEVILACQUA, Istituto Di Calcolo E Reti Ad Alte Prestazioni, Rende, Rende, CS, Italy

ANGELO CIARAMELLA, Parthenope University of Naples, Naples, NA, Italy

IVANOE DE FALCO, Italian National Research Council, Rome, RM, Italy

GIOVANNA SANNINO, Italian National Research Council, Rome, RM, Italy

Published: 07 May 2025
Online AM: 08 April 2025
Accepted: 01 April 2025
Revised: 27 January 2025
Received: 06 June 2024

Ante-Hoc Methods for Interpretable Deep Models: A Survey

ANTONIO DI MARINO, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Naples, Italy
VINCENZO BEVILACQUA, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Naples, Italy
ANGELO CIARAMELLA, University of Naples Parthenope, Napoli, Italy
IVANOE DE FALCO, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Napoli, Italy
GIOVANNA SANNINO, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Napoli, Italy

The increasing use of black-box networks in high-risk contexts has led researchers to propose explainable methods to make these networks transparent. Most methods that allow us to understand the behavior of Deep Neural Networks (DNNs) are post-hoc approaches, implying that the explainability is questionable, as these methods do not clarify the internal behavior of a model. Thus, this demonstrates the difficulty of interpreting the internal behavior of deep models. This systematic literature review collects the ante-hoc methods that provide an understanding of the internal mechanisms of deep models and which can be helpful to researchers who need to use interpretability methods to clarify DNNs. This work provides the definitions of strong interpretability and weak interpretability, which will be used to describe the interpretability of the methods discussed in this article. The results of this work are divided mainly into prototype-based methods, concept-based methods, and other interpretability methods for deep models.

Antonio Di Marino and Vincenzo Bevilacqua contributed equally to this research.
This work was supported by: the Future Artificial Intelligence Research (FAIR) project (PE0000013 - CUP B53C22003630006), Spoke 3 - Resilient AI, within the National Recovery and Resilience Plan (PNRR) of the Italian Ministry of University and Research (MUR); the Digital-Driven Diagnostics, Prognostics, and Therapeutics for Sustainable Health Care (D34Health) project (PNC0000001 - CUP B83C22006120001), within the National plan for investments complementary to the PNRR, financed by the European Union -NextGenerationEU; the Digital Twin and Fintech services for sustainable supply chain (SmarTwin) project (Fondo per la Crescita Sostenibile – Accordi per l'innovazione di cui al D.M. 31 dicembre 2021e D.D. 18 marzo 2022 - CUP B69J23000500005) Ministero dello Sviluppo Economico (MISE); the context-AwaRe deCision-making for Autonomus unmmaneD vehicles in mArine environmental monitoring (ARCAD-IA) project (PE00000013_1 - CUP E63C22002150007) cascade call of the Future Artificial Intelligence Research (FAIR) project Spoke 3 - Resilient AI, within the National Recovery and Resilience Plan (PNRR) of the Italian Ministry of University and Research (MUR).
Authors' Contact Information: Antonio Di Marino, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Naples, Campania, Italy; e-mail: antonio.dimarino@icar.cnr.it; Vincenzo Bevilacqua, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Naples, Campania, Italy; e-mail: vincenzo.bevilacqua@icar.cnr.it; Angelo Ciaramella, University of Naples Parthenope, Napoli, Campania, Italy; e-mail: angelo.ciaramella@uniparthenope.it; Ivanoe De Falco, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Napoli, Italy; e-mail: ivanoe.defalco@icar.cnr.it; Giovanna Sannino, Institute for High-Performance Computing and Networking (ICAR), Italian National Research Council (CNR), Napoli, Italy; e-mail: giovanna.sannino@icar.cnr.it.

ACM 0360-0300/2025/05-ART262
https://doi.org/10.1145/3728637

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computing methodologies** → **Artificial intelligence;**

Additional Key Words and Phrases: Interpretability, ante-hoc methods, deep neural networks, survey, systematic literature review

ACM Reference Format:
Antonio Di Marino, Vincenzo Bevilacqua, Angelo Ciaramella, Ivanoe De Falco, and Giovanna Sannino. 2025. Ante-Hoc Methods for Interpretable Deep Models: A Survey. *ACM Comput. Surv.* 57, 10, Article 262 (May 2025), 36 pages. https://doi.org/10.1145/3728637

1 Introduction

In the set of subjects in the ***Artificial Intelligence*** (**AI**) domain, ***Machine Learning*** (**ML**) and ***Deep Learning*** (**DL**) are among the most popular and widely used nowadays. Techniques and models developed in the ML and DL domains are now used in all fields. For example, in ***Natural Language Processing*** (**NLP**), different models are used for speech recognition and intelligent chatbot services; in Emotion Recognition, ML or DL models are used to classify the emotional state of human beings; and in Digital Advertising, models are used to determine which products to sponsor to a user based on their internet browsing data.

Otherwise, AI techniques and models can be used in high-risk contexts such as autonomous driving, Healthcare, and agriculture. For example, in autonomous driving the risk is that the model will make a decision that impacts people's safety. In Healthcare, the risk is that the model may take the wrong decision on the patient's disease status. Whereas, in agriculture, the risk is that, for example, a model has to detect whether a plant is infected to prevent the spread of the disease.

In all high-risk contexts, it is critical to give reasons why the model provided a particular output. In general, companies and researchers developing AI models focus more on proving their high accuracy, rather than motivating why a model behaved a certain way. As noted in Rudin's work [77], developing an explicable model is more difficult than developing a model with high accuracy; this phenomenon can be attributed to the false myth of the tradeoff between accuracy and interpretability prevalent in the scientific community, which results in little attention from the scientific community to issues such as model interpretability. Moreover, companies prefer not to provide explanations of model behavior, both as a matter of corporate secrecy and also because developing explainable models requires more time and resources.

The need to get an explanation from AI models is not only due to the consequences of missing explanations in high-risk contexts, but it has also become a duty required by institutions such as in the European AI ACT law[1] that will come into force in 2026. It requires any AI system to be transparent, in other words, clearly motivate its decisions as stated in Article 13 "High-risk AI systems shall be designed and developed in such a way as to ensure that their operation is sufficiently transparent to enable deployers to interpret a system's output and use it appropriately" (Article 13 of AI ACT[2]). This law is an expansion of the ***General Data Protection Regulation*** (**GDPR**) law [27] and of other AI regulation plans that govern "right to an explanation" [32].

In addition, a model that is understandable and provides explanations instills greater confidence in users. The efforts made by the scientific community in order to explain AI models have led to

[1] https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence
[2] https://artificialintelligenceact.eu/article/13/

the creation of two approaches that address the same topic in different ways: explainability and interpretability. In this article, we will refer to the term explanation both when we talk about explainability and when we talk about interpretability.

The term **explainability** denotes methods that provide explanations of an AI model using its output [95]. In this context, several works have been proposed based on *perturbation* [62, 70], *back-propagation* [4, 81, 84, 86, 111], and *approximation* [74].

The term **interpretability** denotes all those methods that make the internal behavior of an AI model transparent. In this way, the decisions an AI makes to provide an output become understandable to users.

Generally, in the field of ML, there are models that are inherently interpretable because the structure of the model is simple enough to explain and evaluate. Some inherently interpretable models are, for example, *linear regression* [7, 61, 93], *decision trees* [9, 71], and *decision rules* [20, 37, 51].

In DL, models automate feature extraction across deep layers by representing samples in a latent space that is not comprehensible to a human, making the models black boxes; consequently, these models are not inherently interpretable. Given the increasing use of DL models, several approaches have been proposed in recent years to introduce an interpretability component or make a deep model interpretable.

In this article, we aim to provide an overview of ante-hoc interpretability methods for deep models, since there is a lack of surveys related to this type of methods in the literature. As we show in the next sections, currently, there are surveys that only deal with interpretability methods applied only to ML models. Or, surveys that misuse the word interpretability to describe post-hoc explainability methods in DL.

Section 2 introduces the need to have models that can explain their choices, and we will define in more detail the terminology adopted in this article. Section 3 describes our approach for the ***Systematic Literature Review*** (**SLR**), listing related surveys on interpretability and explainability and showing what criteria and search queries were used to find the papers presented in this SLR. Section 4 illustrates the search results obtained for our SLR in the area of interpretability methods on deep models. Section 5 offers a comprehensive review of the interpretability properties afforded by the various methods. It also addresses the various aspects related to the usefulness of these methods in different applications and provides insights into interpretability. Finally, Section 6 presents the conclusions and future work.

2 Motivations and Terminology

In this chapter, we report on the motivations that led the scientific community to develop the concept of explaining neural networks, downstream of the increasing use of black-box models. The properties that constitute a good explanation will be described. In addition, it is necessary to distinguish between the terms used in this context and in this work.

2.1 Why We Need Explanations?

The need to obtain an explanation has always existed, as in the early days of AI, engineers had the same need as today to understand the decisions of an intelligent system. A series of historical examples of methods for obtaining explanations from AI models are listed in the work of Confalonieri et al. [21]. One example is MYCIN [10], a rule-based system developed in the late 1970s to support physicians diagnosing infections in patients.

In this section, we focus on the reasons for the need to obtain explanations in modern AI systems. This need comes to public attention as some governments begin to propose the first bills to introduce the right to explanation: GDPR, proposed by the European Union in 2016 [27]; "The

Development Plan for New Generation of Artificial Intelligence" by the Chinese government in 2017 [101]; or government programs such as the "***Explainable AI*** (**XAI**) program" of ***Defense Advanced Research Projects Agency*** (**DARPA**) started in 2017 [33] in order to direct research toward the development of XAI Systems.

The first signs of public interest in the need to obtain explanations from AI models can be found in articles such as "Can A.I. Be Taught to Explain Itself?" by Cliff Kuang [48], in which several examples are given of researchers who, in the years prior to the publication of this article, encountered the need to explain their models. A well-known example in the scientific community is the work of Caruana et al. [12], in which they trained a black-box model that predicts the risk of dying from pneumonia in a patient. Their model incorrectly stated that asthma patients had a lower risk of dying than healthy patients. This misbehavior was due to the fact that the model was trained on a dataset that had a systematic bias; that is, it was not considered that the samples of asthmatic people were collected on patients who were already undergoing therapeutic treatment under strict medical supervision. Consequently, the need to obtain an explanation of the behavior of black-box models depends not only on legislative requirements but also on the real need to understand the output of these models; otherwise, such systems cannot be used in complete safety.

2.2 How Should the Explanations Be?

Previously, we have often introduced the need to explain a model without defining how an explanation should be. Explanations must be comprehensible to humans; there are several fields of research: sociology, cognitive psychology, philosophy that study what the properties of a machine-generated explanation must be in order to be comprehensible to humans. A detailed overview of the guidelines for a good explanation can be found in Miller's paper [64], a summary of which can be found in a few paragraphs of Molnar's book [65]. In our work, we simply want to provide the main properties that constitute a good explanation, which we will analyze in the methods discussed in the next paragraphs.

Let's agree that in general, an explanation in AI should be an answer to a *why-question* [64] and should be an *"everyday"-explanation*, that is, an answer to a local question such as, *"Why, according to Caruana's model, do healthy patients have a higher risk of dying than asthma patients?"*. People request an explanation about events they consider unexpected or abnormal; *"everyday"-explanations* consisting of local explanations of why particular events occurred are preferred over more general explanations, such as scientific theories or answers to philosophical questions. An explanation should facilitate people's learning and enhance their understanding of the model's behavior.

A good explanation should be generated with the following properties, defined in the work of Miller et al. [64], summarized as follows:

- **Selective**: people prefer few explanations that best highlight the cause of a model's behavior rather than the whole list of possible explanations.
- **Contrastive** [58]: people prefer to receive an explanation based on a comparison with a reference case rather than an isolated explanation. In fact, the explanation based on a comparison is easier to understand because the user points out the differences from the reference case.
- **Social**: the explanation must take into account the target audience to which it is addressed; if the end user is an expert in the field, the explanation can be technical, while, for general users, the explanations should be user-friendly.

— **Truthful**: an explanation generates total confidence in the system if it includes all the causes of an event. There are cases, however, where the number of causes is undefined or where some explanations are more representative than others; in these cases, a more selective explanation is preferred. Too much selectivity risks reducing user confidence in the system, so truthfulness can be seen as a tradeoff between selectivity and exhaustiveness of explanations.
— **Focused on the abnormal**: given a set of causes of a pattern of behavior, people are more interested in explanations about abnormal and unforeseen causes than explanations about obvious and expected causes.

The primary characteristics that typically contribute to a transparent, user-centric explanation have been documented. However, there are additional attributes that will not be examined in this article, as they are exclusively pertinent to specific contexts.

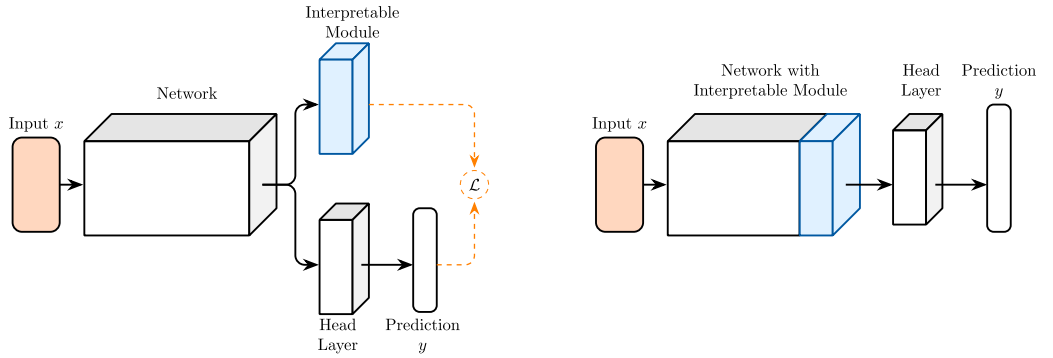
2.3 Terminology

In this section, we are going to define the terminology we will use in the article when presenting the various methods, as well as provide a general definition for explainability and interpretability.

— **Transparency**: is the ability of a model to make its operation clear; a transparent model is one whose behavior can be predicted by the user. There are different levels of transparency; a highly transparent model shows all its decisions that led it to generate an output.
— **Black-Box**: is the term used to define a model that hides its internal behavior and, in output, provides only its prediction. A black-box model is not a transparent model.
— **Explainability**: is the field in which an explanation is generated from the prediction of a black-box model in order to understand why a model provided a certain output a posteriori. Generally, explainability methods are portable because they do not depend on the architecture of the model on which to use them.
— **Interpretability**: there is no official definition of interpretability in the literature, as pointed out in [77], but we generally use it to refer to the field in which we study a model's behavior, in order to understand the intrinsic decisions that lead a model to generate an output. Using an interpretability method on a model allows us to improve its transparency.
— **Post-hoc**: the use of a second model to generate an explanation from the input/output of the main model; in general, explainability methods are post-hoc methods.
— **Ante-hoc**: we refer to methods that automatically generate an explanation during the train or use phase.

Generally, in DL, post-hoc methods are widely used to generate explanations from the black box. As Lipton suggests in his paper [59], in which he encourages researchers to define their own metrics of evaluations, in the absence of a general mathematical definition, we define two levels of interpretability that we will use in this article:

— **Weak Interpretability**: as shown in Figure 1(a), the interpretable component of the network **does not contribute directly** to the generation of the network output but contributes indirectly, e.g., only in the training phase for the contribution of the calculation of the loss function.
— **Strong Interpretability**: as shown in Figure 1(b), the interpretable component **contributes directly** to the generation of a network's output.



(a) Weak Interpretability (b) Strong Interpretability

Fig. 1. Examples of weak and strong interpretability. (a) The weak interpretability module only contributes during the calculation of the loss function $\mathcal{L}$. (b) The strong interpretability module always contributes during the generation of output y.

In this article, we highlight the component that adds interpretability in the models, as shown in Figure 1, highlighting in light blue the interpretability module in each method discussed.

3 SLR Approach

In this chapter, we describe our methodology for conducting an SLR. We follow the widely used guidelines provided by Kitchenham [45]. First, an overview of previous surveys that have addressed the topic of explainability in ML models will be given, noting that few surveys describe interpretability exclusively in deep models. Next, through some **research questions** (**RQs**), we provide the rationale for compiling this SLR. Finally, we describe the search methods we used to find suitable papers, defining the **inclusion and exclusion criteria** (**IEC**).

3.1 Summary of Previous Reviews

Previous surveys were found using the following search engines: *Google Scholar*,[3] *Scopus*,[4] and *IEEE Xplore*.[5] The following queries were used to perform the search:

— *{deep learning} AND interpretability AND survey.*
— *{deep learning} AND explainable AND survey.*
— *{deep learning} AND interpretable AND survey.*
— *survey AND xai AND {deep learning}.*

These queries were used to conduct a search of the surveys dealing with interpretability methods in DL. As can be seen in Table 1, surveys dealing with interpretable DL became popular since 2017.

It can be seen that, in recent years, there has been an increase in the production of scholarly articles on this topic, demonstrating that as the use of black-box models in DL increases, so does the need to explain them.

The predominant surveys have been found in the field of healthcare [42, 67, 92, 94, 96], where, for example, the work of Tjoa et al. [94] makes a categorization of existing explainability methods by dividing them into: *perceptive interpretability* and *mathematical structure interpretability*, applying

[3] https://scholar.google.com/
[4] https://www.scopus.com/
[5] https://ieeexplore.ieee.org/

the categorization in the medical field, providing guidelines and future directions for clinicians and practitioners. Also, in the work of Teng et al. [92], there is a categorization of explainability techniques for DL, summarizing their application for disease diagnosis. Another example is the work of van der Velden [96] where an overview of explainability methods used purely for medical image analysis is made.

Another safe-critical application concerns autonomous driving systems, as discussed in the work of Zablocki et al. [106], which provides an overview of post-hoc explainability methods on DL models for autonomous driving systems. The work of Ahmed et al. [1] presents an overview of AI, explainability methods, and applications of AI in Industry 4.0. In the paper of Zhang et al. [109], an overview of XAI methods applied to the field of Cyber Security is made by describing the cyber security systems that XAI techniques use as well as describing possible cyber attacks and the work to defend against these attacks.

While many surveys propose explainability and interpretability techniques applied to deep models operating on images or tabular data, a noteworthy survey is the work of Zini et al. [112], in which the authors give a general overview of explainability and interpretability techniques applied on NLP models. From this work, it is found that the lack of explainable NLP models depends on the difficulty of visualizing the internal representation of deep models that work on words. Moreover, they are the first to analyze the explainability of word embeddings models that cover a crucial role in NLP, then they discuss the internal representations of NLP networks, and discuss the degree of interpretability of the attention mechanisms used in word processing transformers.

Some other surveys aim to provide guidelines and define future challenges for AI interpretability. An example is the work of Rudin et al. [78] where 10 technical challenges are identified for ML, such as: how to build sparse models for tabular data, including scoring system, or how to understand, explore, and measure the Rashomon set of accurate predictive models. In addition, this paper highlights the problem that in the literature there is confusion in the use of the terms explainability and interpretability, which are often used synonymously even in the description of techniques. We note that this problem also plagues some surveys that we have reported in Table 1, in which the term interpretability is misused to describe explainability methods on deep models. Therefore, we want to define an overview of ante-hoc methods to clearly introduce interpretability for DL models.

3.2 Review Questions

Previous works describe post-hoc explainability methods only for DL, while they analyse ante-hoc interpretability only for ML models. Therefore, in order to make the analysis of this topic more comprehensive, we decided to focus our SLR on ante-hoc interpretability methods for DL.

Therefore, we defined the following *RQs*:

RQ1. Do ante-hoc methods exist for interpreting DL models?
RQ2. Are the proposed interpretability methods effectively ante-hoc or is there confusion and are they post-hoc?
RQ3. In what contexts can the proposed interpretable method be used?
RQ4. How much user-friendly is the interpretability of a proposed method?
RQ5. What types of DL models are interpretable?

The previous questions encapsulate the doubts that arose during the search for ante-hoc interpretability methods for DL. The motivations for drafting the previous RQs are the following:

RQ1 comes from the great difficulty in finding interpretability methods in deep models, since most works in the literature provide post-hoc explainability methods. This trend is due to the fact that the internal behavior of deep models is difficult to interpret, so this problem, combined

Table 1. Table of Previous Surveys Found with the Search Engines *Google Scholar*, *Scopus,* and *IEEE Xplore*

Authors	Title	Year	Focus Area
Chakraborty et al. [13]	Interpretability of deep learning models: A survey of results	2017	XAI Techniques
Gilpin et al. [31]	Explaining explanations: An overview of interpretability of machine learning	2018	XAI Techniques
Zhang et al. [107]	Visual interpretability for deep learning: a survey	2018	XAI in CNNs
Dosilovic et al. [26]	Explainable artificial intelligence: A survey	2018	XAI Techniques
Xu et al. [102]	Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges	2019	Historical perspective of XAI
Chatzimparmpas et al. [14]	A survey of surveys on the use of visualization for interpreting machine learning models	2020	Survey of surveys in XAI
Mi et al. [63]	Review study of interpretation methods for future interpretable machine learning	2020	XAI Techniques
Cheng et al. [18]	Interpretability of deep learning: A survey	2020	XAI Techniques
Linardatos et al. [57]	Explainable ai: A review of machine learning interpretability methods	2020	XAI Techniques
Li et al. [55]	A Survey of Data-Driven and Knowledge-Aware eXplainable AI	2020	XAI Techniques
Liang et al. [56]	Explaining the black-box model: A survey of local interpretation methods for deep neural networks	2021	Local methods for DNNs
Tjoa et al. [94]	A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI	2021	XAI for Healthcare
Ivanovs et al. [41]	Perturbation-based methods for explaining deep neural networks: A survey	2021	Perturbation-based XAI methods
Zhang et al. [108]	A Survey on Neural Network Interpretability	2021	XAI Techniques
Fan et al. [28]	On interpretability of artificial neural networks: A survey	2021	Interpretation Methods
Kovalerchuk et al. [47]	Survey of Explainable Machine Learning with Visual and Granular Methods Beyond Quasi-Explanations	2021	Visual Approaches in XAI
Buhrmester et al. [11]	Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey	2021	Computer Vision
Li et al. [54]	Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond	2022	Trustworthy AI
Zablocki et al. [106]	Explainability of Deep Vision-Based Autonomous Driving Systems: Review and Challenges	2022	XAI for Autonomous Driving
Zhang et al. [109]	Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research	2022	XAI in Cyber Security
Ding et al. [25]	Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey	2022	XAI Techniques
Teng et al. [92]	A survey on the interpretability of deep learning in medical diagnosis	2022	XAI for Healthcare
Ahmed et al. [1]	From Artificial Intelligence to Explainable Artificial Intelligence in Industry 4.0: A Survey on What, How, and Where	2022	XAI in Industry 4.0
Alicioglu et al. [2]	A survey of visual analytics for Explainable Artificial Intelligence methods	2022	Visual Approaches in XAI
Zini et al. [112]	On the Explainability of Natural Language Processing Deep Models	2022	XAI in NLP

(Continued)

Table 1. Continued

Authors	Title	Year	Focus Area
van der Velden et al. [96]	Explainable artificial intelligence (XAI) in deep learning-based medical image analysis	2022	XAI for Healthcare
Jin et al. [42]	Explainable deep learning in healthcare: A methodological survey from an attribution view	2022	XAI for Healthcare
Rudin et al. [78]	Interpretable machine learning: Fundamental principles and 10 grand challenges	2022	Interpretability in ML
Saeed et al. [79]	Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities	2023	Challenges and future directions in XAI
Nazir et al. [67]	Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks	2023	XAI for Healthcare
Ciatto et al. [19]	Symbolic Knowledge Extraction and Injection with Sub-symbolic Predictors: A Systematic Literature Review	2024	XAI Techniques

with the difficulty in developing ante-hoc methods for deep models, led us to consider RQ1 when researching and studying the works in the literature.

RQ2 comes from the problem that plagues the literature in the area of explainability, whereby there is often confusion with the terminology by not defining it appropriately. For example, works were found where interpretable DL was mentioned but the proposed methods were post-hoc; consequently, in our survey, all research was done by looking particularly carefully at the terms used in order to present only ante-hoc methodologies for DL.

RQ3 comes from the observation that some interpretability methods, tested on experimental datasets (such as ***Modified National Institute of Standards and Technology*** (**MNIST**) database [50]), yield favorable results. RQ3 specifically examines the effectiveness of the methods in more complex contexts.

RQ4 comes from the observation that some interpretability methods generate a non-user-friendly explanation that an ordinary user is unlikely to appreciate. In some cases, even experienced users would have difficulty grasping its meaning. Therefore, RQ4 focuses on the method's ability to be easily understood.

RQ5 comes from the observation that some deep models, e.g., ***Graph Neural Networks*** (**GNNs**), use very complex mechanisms to be intrinsically interpreted, while there are some deep models that already possess an interpretable component and are therefore easier to interpret than others.

Our responses to the RQs are shown in Section 5.5, after reviewing the various works in the literature.

3.3 Review Methods

In this section, we discuss the search methodology used to find works that present ante-hoc interpretability methods for DL models. The next sections will describe the IEC used to select the most suitable papers, as well as quantitatively summarize the number of papers considered.

To conduct our SLR, three popular search engines for academic literature were used:

— Google Scholar
— IEEE Xplore
— Scopus

In order to conduct the search for the papers presented in this article, several queries were defined and then used in the search engines listed above; booleans *AND*, *OR*, and *NOT* were used in the query construction.

Let us consider that as filtering conditions, generic queries must contain generic keywords, for example, *"interpretability"* or *"self-explaining"*, while for selective queries we impose the constraint that they must consist of some targeted keywords, such as *"Transformers"* or *"Graph Neural Networks"*. In the search filtering stages, we consider only papers that are in English and have been published from 2017 onward.

To collect the papers, two types of searches were done. The first type of search was to obtain a broad spectrum of papers on the interpretability of DL by creating generic queries, as in the following examples:

— *interpretability AND method AND self-explaining AND "neural networks"*
— *interpretable AND "image recognition"*

The second type of search is used to obtain selective results of interpretability methods on specific deep architectures, as in the following examples:

— *Interpretability AND transformer OR transformers*
— *(Interpretable OR Interpretability) AND "graph neural networks" NOT survey NOT review*

The above queries are examples, since many queries were used in order to try to explore the whole domain. Then the results of these queries are further filtered based on their content.

3.3.1 Inclusion and Exclusion Criteria. Given the volume of papers found with the different queries used, IEC were applied in this SLR in order to consider only those papers deemed suitable and proposing methodologies to interpret deep models. We present the following criteria below:

IEC1. Articles must be published in peer-reviewed journals or at conferences. Articles that do not meet these criteria are excluded.
IEC2. Articles presenting the keywords "interpretability" or "interpretable" in the title or abstract, however, then presenting a post-hoc method of explainability were excluded.
IEC3. Articles that present interpretability methods in DL are included. While, Articles that do not present interpretability methods in DL but, for example, present methodologies to interpret inherently interpretable ML models were excluded.
IEC4. Articles that introduce a new interpretability method are included. While, articles that address interpretability in deep models but do not introduce a new interpretability method are excluded.
IEC5. Only early versions of the study were considered; later versions or those that did not bring a concrete innovation to the proposed methodology were excluded.
IEC6. Doctoral theses or dissertations are not considered in this review.
IEC7. Studies that do not present comparison metrics or evidence of experiments performed were excluded. Although interpretability is not quantifiable, we believe that comparison with other methods, even those that are not interpretable, is essential.

During our study, a total of 1, 226 papers were identified from 2017 to 2024 through the three aforementioned search engines. Following the exclusion criteria, 1, 200 of these papers were discarded. The remaining 26 papers will be the focus of this SLR.

4 SLR Results

In this section, we show the results obtained from research for this SLR. All the works shown in this section concern ante-hoc methods for interpretability on different types of deep architectures. We divide the methods found into: prototype-based methods and concept-based methods, as several variants have been proposed in the literature, and, finally, a subsection for other interpretability techniques.

Table 2. Prototype-Based Methods Ordered by Publication Date

Authors	Method	Model	Year
Oscar Li et al. [53]	First Prototype-based method	CNN	2018
Chaofan Chen et al. [16]	Prototypical Part Network (ProtoPNet)	CNN	2019
Gurmail Singh et al. [87]	Negative-Positive Prototypical Part Network (NP-ProtoPNet)	CNN	2021
Meike Nauta et al. [66]	Prototype-based method	CNN	2021
Zaixi Zhang et al. [110]	Prototype Graph Neural Network (ProtGNN)	GNN	2022
Sangwon Kim et al. [44]	Vision Transformers with Neural Tree Decoder (ViT-NeT)	Transformers	2022
Jingqi Wang et al. [100]	High Quality Prototypical Part Network (HQProtoPNet)	CNN	2023
Seo et al. [82]	Prototype-based Graph Information Bottleneck (PGIB)	GNN	2023
Gao et al. [30]	Transferable Conceptual Prototype Learning (TCPL)	CNN	2024
Peng et al. [69]	Decoupling Prototypical Network (DProtoNet)	CNN	2024

4.1 Prototype-Based Methods

As discussed in previous sections, for us, making a model interpretable means that a deep model, for example, a ***Convolutional Neural Network*** (**CNN**), must be able to explain why it produced a certain output $\hat{y}$ by showing its intermediate features that contributed to generate the output of the CNN itself.

One of the first solutions proposed for the interpretability of deep models was prototype-based methods. Prototype-based methods aim to represent and make observable the prototypes that the network uses internally, in order to compare them with a sample given as input to be evaluated. These methods give the user the ability to understand the behavior that allowed the deep model to generate the output. Through the years, prototype-based methods have been improved and used on increasingly complex deep architectures. All the methods that will be explained in this section are collected in Table 2. Figure 2 shows an example of an application of a prototype-based network, ProtoPNet [16], to show what it means to interpret a model visualizing the prototypes that contributed the most to the classification.

The first work that introduces the use of prototypes and their visualization in CNNs is the method proposed by Li et al. [53]. Other methods illustrated in this section are ***Prototypical Part Network*** (**ProtoPNet**) [16], ***High Quality Prototypical Part Network*** (**HQProtoPNet**) [100], ***Negative-Positive Prototypical Part Network*** (**NP-ProtoPNet**) [87], *This Looks Like That, Because* by Nauta et al. [66], ***Transferable Conceptual Prototype Learning*** (**TCPL**) [30], ***Decoupling Prototypical Network*** (**DProtoNet**) [69], ***Prototype Graph Neural Network*** (**ProtGNN**) [110], ***Vision Transformers with Neural Tree Decoder*** (**ViT-NeT**) [44], ***Prototype-based Graph Information Bottleneck*** (**PGIB**) [82].

Prototypes method by Li et al. [53]. This interpretability method has been proposed for image classification problems, in which the goal is to explain why a model has decided that an image belongs to a given class at the expense of other classes. The method exploits the latent space of deep models to create prototypes representing classes. Each prototype is a semantic representation of what the model considers important in assigning an input image to a given class. The network architecture involves the following:

— *Input*: image $x \in \mathbb{R}^p$
— *Autoencoder*:
  – Encoder $f(\cdot)$: for example, a CNN, to encode input image x in a latent space $\mathbb{R}^q$.
  – Decoder $g(\cdot)$: used to reconstruct an image $\hat{x}$ from representations in the latent space $\mathbb{R}^q$, so $g : \mathbb{R}^q \rightarrow \mathbb{R}^p$.
— *Prototype Classification Network* $h(\cdot)$: consists of three layers and produces the probability distribution over the K classes:

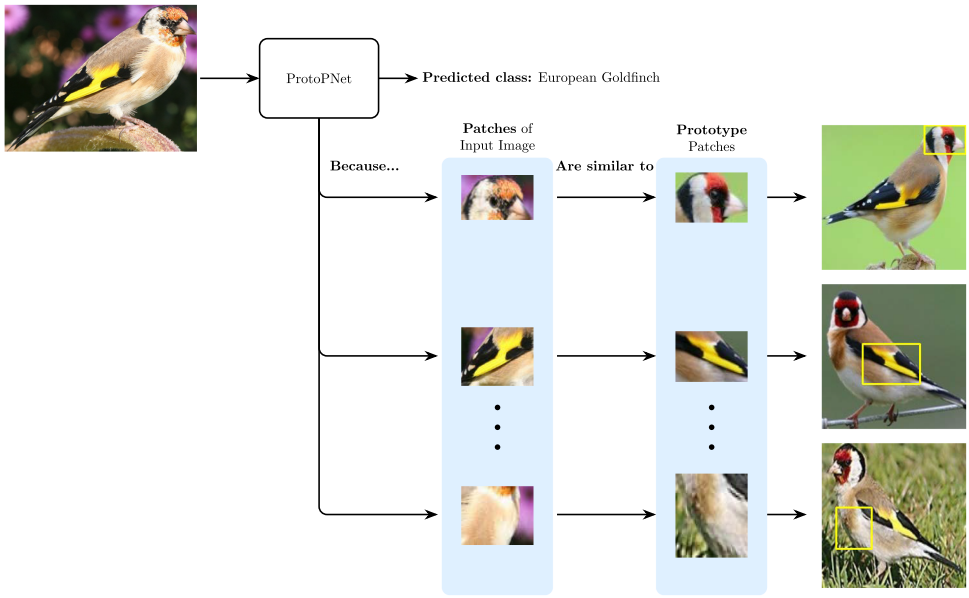


Fig. 2. Example application of ProtoPNet [16] on a test image of the *Caltech-UCSD Birds-200-2011* (CUB-200-2011) dataset [98]. ProtoPNet shows the patches found in the input image that correspond to prototypes of a given class.

- *Prototype Layer* $p(\cdot)$: is the layer that is responsible for storing m prototypes, $p : \mathbb{R}^q \rightarrow \mathbb{R}^m$, calculates the Euclidean distance between the encoded input $z = f(x_i)$ and each prototype vector $p_i \in \mathbb{R}^q \; \forall \; i = 1, \ldots, m$:

$$p(z) = \left[\|z - p_1\|_2^2, \|z - p_2\|_2^2, \ldots, \|z - p_m\|_2^2 \right]^T, \tag{1}$$

  the output of layer $p(z) \in \mathbb{R}^m$, i.e., is a vector of m Euclidean distances between z, i.e., the input x encoded in the latent space, and ith prototype p_i.
- *Fully-connected layer* $w(\cdot)$: is a classical layer head used in deep networks to obtain k logits as many as there are classes $w : \mathbb{R}^m \rightarrow \mathbb{R}^k$, then compute weighted distances $Wp(z)$, where W is a matrix of $K \times m$ weights.
- *Softmax Layer* $s(\cdot)$: is the layer that outputs the probability distribution over the k classes, so $s : \mathbb{R}^k \rightarrow \mathbb{R}^k$, where a kth component given in the output by the softmax layer is the estimated probability of belonging to the kth class.

Thus, this proposed classification model uses a distance-based classification algorithm in a low-dimensional learned feature space. Considering that the deep model was modified by adding the prototype layer, to enable the model to optimize prototypes, so that they best represent class features, Li et al. propose a new cost function:

$$\begin{aligned} L((f, g, h), D) = E(h \circ f, D) + \lambda R(g \circ f, D) \\ + \lambda_1 R_1(p_1, \ldots, p_m, D) \\ + \lambda_2 R_2(p_1, \ldots, p_m, D), \end{aligned} \tag{2}$$

where: λ, λ_1, and λ_2 are hyperparameters that weigh loss terms; E is the standard cross-entropy to optimize classification; R is the reconstruction loss to optimize the decoding phase of the network; while R_1 and R_2 are the two regularization terms. The minimization of R_1 will allow each prototype

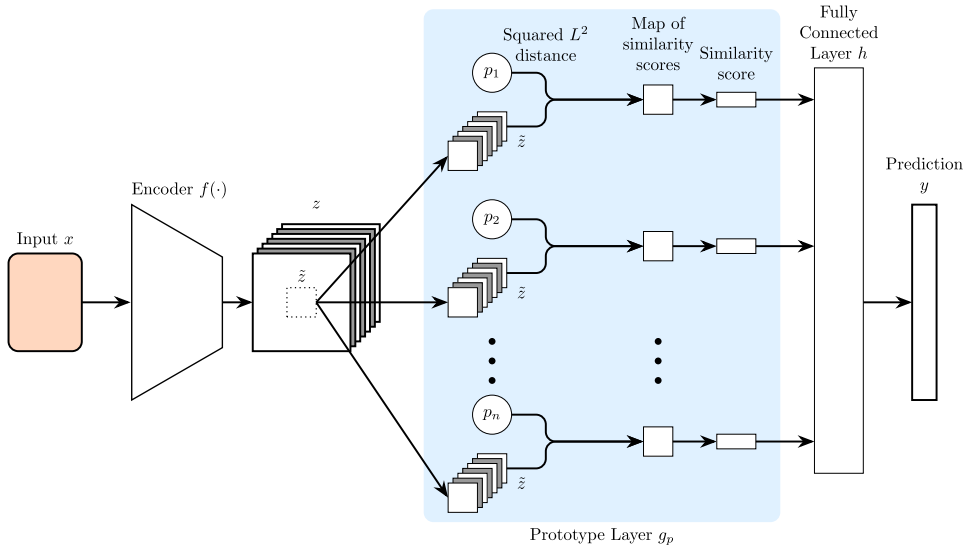


Fig. 3. ProtoPNet architecture replicated by us based on the original work [16].

vector to be as close as possible to at least one training example in latent space. The minimization of R_2 will allow each training example encoded in latent space to be as close as possible to at least one prototype vector. For more details on the individual loss terms, the reader is referred to the paper [53].

ProtoPNet [16]. This interpretability method is based on the work of Li et al. [53], with some modifications. ProtoPNet is proposed for fine-grained classification problems. Looking at Figure 3, where the architecture is replicated, ProtoPNet uses an encoder f to encode the input image x in a latent space, thus obtaining $z = f(x)$. Unlike Li et al. [53], where the prototypes represent entire fixed-size images, in this *ProtoPNet*, the prototypes learned from the prototype layer g_p are regions (patches) of the variable size image. This layer g_p computes the square of the Euclidean distances between the jth prototype p_j and all patches of z that have the same shape $p_j, \tilde{z} \in \mathbb{R}^q$, and then inverts these distances into similarity scores. These activation maps of similarity scores that are produced by each prototypical unit g_{p_j} are reduced to a single similarity score using a global max pooling, so as to understand how strongly that prototypical part is present in some patch of the input image. All similarity scores are given as input to the fully connected classifier h, to make the final classification.

Another difference between ProtoPNet [16] and the method by Li et al. [53] is the removal of the decoder, so, in the absence of this module, the display of prototypes in ProtoPNet is done differently. This method uses the patches of the training images that have the smallest Euclidean distance in the latent space to the prototypes learned with the same class. Mathematically, the association between the image patch z with a prototype p_j is expressed with the following equation:

$$vp_j \leftarrow \underset{z \in \mathcal{Z}_j}{\operatorname{argmin}} \|z - p_j\|_2, \ \textit{where } \mathcal{Z}_j = \{\tilde{z} : \tilde{z} \in \textit{patches}(f(x_i)) \ \forall i \ s.t. \ y_i = k\}, \tag{3}$$

where vp_j is the visual patch taken from the image associated with the prototype p_j that can be used in visualization. Another major difference compared to the work of Li et al. [53] is the different

loss function L used in ProtoPNet, like the following function:

$$L = \min_{P, w_{conv}} \frac{1}{n} \sum_{i=1}^{n} CrsEnt(h \circ g_p \circ f(x_i), y_i) + \lambda_1 Clst + \lambda_2 Sep. \quad (4)$$

They remove the reconstruction term R since the decoder has been removed, and the regularization terms *Cluster cost* (*Clst*) and *Separation cost* (*Sep*) encourage the model to have: a few patches close to a prototype of the same class, and every latent patch away from prototypes that do not belong to its class, respectively. The terms λ_1 and λ_2 are hyperparameters to adjust the importance of the terms.

HQProtoPNet [100]. This method proposes improved prototypes over ProtoPNet in order to have better prediction accuracy and stronger evidence reliability. The *HQProtoPNet* architecture introduces the use of random erasing to make prototype learning more robust, through the addition of an *Augmentor layer* before the encoder $f(\cdot)$. The authors introduce the ***MultiScale Matching*** (**MSM**) layer to convert the output of CNN $f(\cdot)$ into feature maps at different scales in order to obtain results more robust to scale variations. This MSM layer includes convolutional dimension reduction and multiscale pooling. The *Prototype layer* g_P, in place of Euclidean distance, uses the cosine distance to compute the similarity between patches obtained from the MSM layer and prototypes p_j. The *Classification layer* is stackable, which means that each classifier produces output logit scores for each class; these are to be accumulated with the scores of the other classifiers in the stack and then given as input to the *Softmax* activation function to estimate the probability of membership in each class. These changes lead to a modification to the regularization terms in the following loss function, L:

$$L = CrsEnt + \lambda_1 Gen + \lambda_2 Pec, \quad (5)$$

where *CrsEnt* is the cross entropy computed between the class y predicted by the HQProtoPNet and the ground truth, the *Generality function* (*Gen*) should reflect that the prototype should have similar image properties, and the *Peculiarity function* (*Pec*). This latter behaves opposite to *Gen*, that is, a prototype of a class should have peculiarities different from other classes. The terms λ_1 and λ_2 are hyperparameters to adjust the importance of the terms. For further details, we refer the reader to the original article.

NP-ProtoPNet [87]. This *NP-ProtoPNet* proposes a modification to ProtoPNet so that, in addition to positive motivation, it also adds negative motivation to interpret the classification of an image. Their idea of adding the negative motivations (prototypes) comes from wanting to have different opinions to answer the question: why did you choose this class? So in addition to the reasoning, "I assigned a class y because patches *looks like these* prototypes", they also add negative motivations, such as, "I also assigned this class y because the patch *does not look like these* prototypes of other classes". While in ProtoPNet the Fully Connected Layer's W-weight matrix used for connecting similarity scores and logits is initialized to $+1$ and -0.5, in NP-ProtoPNet they initialize connections with incorrect classes to -1 instead of -0.5. This modification allows for prototypes with negative motivations.

This Looks Like That, Because... by Nauta et al. [66]. The authors' contribution is to provide additional textual explanations of the prototypes that ProtoPNet considers most similar to image patches. The added textual information better explains the prototypes. The method quantifies the visual features of: shape, saturation, contrast, color hue, and texture. The five visual features are extracted from the original dataset by creating five new subdatasets, one for each visual feature. The dataset for each visual feature is created, so that it becomes difficult for ProtoPNet to classify those images based on the corresponding visual feature. For example, to create the shape feature

subdataset, the images in the original dataset are reduced in shape contribution by applying linear displacement that warps them.

After creating the five subdatasets, the importance of each feature is calculated with *local* and *global* scores. The *local* score calculates the importance of a visual feature on a single image, while the *global* score calculates the importance of a visual feature on a prototype in general, regardless of the input image. The *local* score is calculated with the following equation:

$$\phi_{local}^{i,j,k} = g_{j,k} - \hat{g}_{i,j,k}, \tag{6}$$

where $i \in \{shape, saturation, contrast, hue, texture\}$, $j \in \{1, 2, \ldots, n\}$ is the index of the jth prototype, and k is the index of the kth image of the test dataset. The similarity between the original image and the prototype is denoted by g, while the similarity between the image of the modified visual feature dataset and the prototype is denoted by $\hat{g}$.

The *global* score of a visual feature i on the jth prototype is calculated as a weighted average over the local scores calculated on the entire training dataset S_{train}, as shown by the following equation:

$$\phi_{global}^{i,j} = \frac{\sum_{k=1}^{|S_{train}|} \phi_{local}^{i,j,k} \cdot g_{j,k}}{\sum_{k=1}^{|S_{train}|} g_{j,k}}. \tag{7}$$

From the analysis that can be done with this method, it can be seen that the model does not always associate patches with prototypes according to visual features that are intuitive to human beings, but is often done through other visual features deemed useful by the model.

TCPL [30]. In this paper, the authors propose a method that adapts in an unsupervised context with the goal of improving the knowledge transfer process. Prototypes are used in the hierarchical prototype module that adapts prototypes from the source (supervised) domain to the target (unsupervised) domain.

DProtoNet [69]. The authors propose a novel architecture that incorporates a coding, inference, and interpretation module. Prototypes are generated through the encoding module, which utilizes decoupled feature masks to facilitate the generation of prototypes. The inference module employs feature and prototype vectors to make predictions based on similarity comparisons. The interpretation module represents the primary innovation of the method for interpreting prototype-based networks and consists of a decoder of multiple dynamic masks.

Although the prototypes were designed for CNNs and were used after the encoding phase, other work has proposed using the same prototype-based interpretability on other deep architectures different from CNNs.

ProtGNN [110]. The authors propose a GNN as the *Encoder* $f(\cdot)$ because their goal is to integrate the use of the ProtoPNet prototype layer by changing the encoder. The problem they face is that, while for images it is easy to take a patch to compare with prototypes in a latent space, in GNNs a patch is comparable to a subgraph that is not easy to find. The authors propose two methods for searching a subgraph from an input graph and perform this step using the *Conditional Subgraph Sampling Module*. The first method for searching a subgraph in a graph is to employ the ***Monte Carlo Tree Search*** (**MCTS**) algorithm [85]. This is a computationally onerous iterative algorithm. Then, as a second method for searching a subgraph in a graph, the authors propose ProtGNN+, which consists of a ***Multi Layer Perceptron*** *Network* (**MLP**) addition that estimates the arcs of a subgraph from the graph nodes and the prototype in latent space. ProtGNN+ is activated after an initial training phase with MCTS to allow the network to perform the prototype projection. Once the subgraphs are found, they are used in latent space, as is done in ProtoPNet using a *Prototype Layer* g_p, which calculates the distance between the subgraph (patch) and the nearest prototypes. The similarity scores are used by a *Fully Connected Layer* c to effect the final classification.

The Loss function is as follows:

$$L = \frac{1}{n}\sum_{i=1}^{n} CrsEnt(c \circ g_p \circ f(x_i), y_i) + \lambda_1 Clst + \lambda_2 Sep + \lambda_3 Div, \tag{8}$$

where the *Clst* and *Sep* terms behave analogously to the work of ProtoPNet, while the new *diversity loss* term (*Div*) uses the cosine distance to penalize prototypes that are too close to each other and thus represent too similar a subgraph. The terms λ_1, λ_2, and λ_3 are hyperparameters to adjust the importance of the terms.

ViT-NeT [44]. In this method, the authors want to use prototype interpretability in a network that uses a *Swin transformer*, a hierarchical transformer, as the encoder because the patch size can be variable. This is useful to best represent both small and large objects. The network also consists of an *Interpretable Neural Tree Decoder*, which is a perfect binary tree with a set of nodes $\mathcal{N}(\cdot)$, leaves $\mathcal{L}(\cdot)$, and arcs $\mathcal{E}_{i,j}(\cdot)$ between parent node i and child node j. To classify an image, the method proposes a branch routing in which it compares the encoded image patches $\tilde{z}$ with the prototypes $\mathcal{P}_i$ of the ith node through a routing score $\mathcal{N}_i$ measured by the logarithm of the Euclidean distance, ϵ is used to prevent the denominator becoming zero when patch and prototype are identical, as shown in the following equation:

$$\mathcal{N}(z_i) = \max_{\tilde{z} \in patches(z_i)} \log\left(\left(\|\tilde{z} - \mathcal{P}_i\|_2^2 + 1\right) / \left(\|\tilde{z} - \mathcal{P}_i\|_2^2 + \epsilon\right)\right). \tag{9}$$

To reinforce discriminative patches, in routing from an ith parent to a jth child, the patch z passes through a ***Contextual Transformer Module*** (**CTM**). This provides a better description of the object by aggregating the global context into the patches of each position, as shown in the following equation:

$$z_j = \mathcal{E}_{i,j}(z_i) = CTM(z_j). \tag{10}$$

Each leaf in the tree corresponds to a deep model of prediction $\mathcal{L}(\cdot)$, which predicts the probability of belonging to the kth class. The final prediction $\tilde{y}$ is calculated with the contribution of all leaf predictions $\mathcal{L}$ multiplied by the accumulated routing scores p_l, σ is the sigmoid activation function, as shown in the following equation:

$$\tilde{y} = \sum_{l \in L} \sigma(\mathcal{L}(z_l, x)) \cdot p_l(z_1), \tag{11}$$

the final prediction $\tilde{y}$ is optimized using a negative logarithmic likelihood loss with the ground truth label y.

PGIB [82]. The integration of prototypes into a GNN is proposed as a superior alternative to ProtGNN. Prototypes are then utilized to identify key subgraphs that have an impact on prediction. The incorporation of prototype learning into an **Information Bottleneck** (**IB**) module enables prototypes to identify key subgraphs of the input graph.

4.2 Concept-Based Methods

An alternative to prototype-based methods is concept-based methods; the goal of these methods always remains to make a deep model interpretable. Unlike prototype-based methods, which are based on optimizing a distance in a latent space, concept-based methods involve optimizing the weights of a layer through the use of regularization terms added to the loss function. As seen before, in prototype-based methods, the networks learn "fixed" prototypes for each class, and thus, in inference to motivate a classification, the same set of prototypes is always associated with different input images. Instead, in concept-based methods, the concept is generalized by the weights of the concept layer $h(\cdot)$, so the concept in the output generated by the layer $h(\cdot)$ changes according to the input and semantically represents a general concept learned by the network. All the methods

Table 3. Concept-Based Methods Ordered by Publication Date

Authors	Method	Model	Year
David Alvarez Melis et al. [3]	Self-Explaining Neural Network (SENN)	CNN	2018
Pang Wei Koh et al. [46]	Concept-Bottleneck Model (CBM)	CNN	2020
Zhi Chen et al. [17]	Concept Whitening (CW)	CNN	2020
Dmitry Kazhdan et al. [43]	Concept-based Model Extraction (CME)	CNN	2020
Mattia Rigotti et al. [75]	ConceptTransformer (CT)	Transformer	2021
Anirban Sarkar et al. [80]	A concept-based method	CNN	2022
Han Xuanyuan et al. [105]	Global concept-based interpretability	GNN	2023
Hila Chefer et al. [15]	Conceptor	DM	2024
Hong et al. [38]	Concept-Centric Transformers (CCT)	Transformer	2024
Tan et al. [90]	OpenCBM	CNN	2024
Xu et al. [103]	Energy-Based Concept Bottleneck Model (ECBM)	CNN	2024

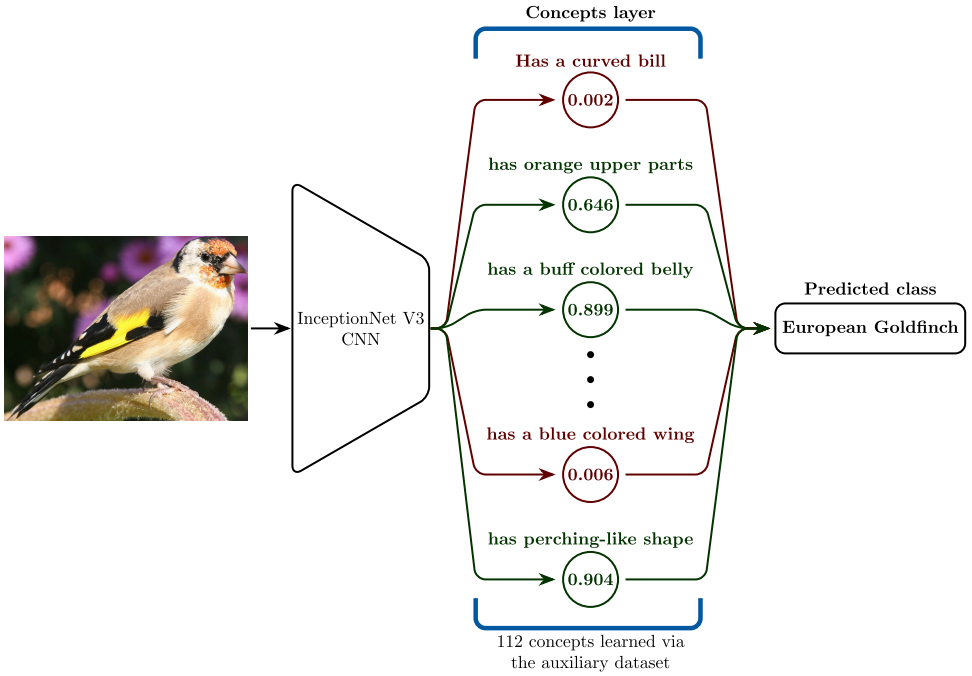


Fig. 4. Example of CBM on a test image of CUB-200-2011 dataset. CBM [46] shows which concepts were most activated to classify the input image.

that will be explained in this section are collected in Table 3. Figure 4 shows an example of an application of a concept-based network, CBM [46], to show what it means to interpret a model using the concepts that contributed to the classification.

The first fully unsupervised ante-hoc concept learning method is the ***Self-Explaining Neural Network*** (**SENN**) [3]. Other methods illustrated in this section are the ***Concept-Bottleneck Model*** (**CBM**) [46], a *concept-based method by Sarkar et al.* [80], ***Concept Whitening*** (**CW**) [17], and ***Concept-based Model Extraction*** (**CME**) [43], *OpenCBM* [90] , ***Energy-Based Concept Bottleneck Model*** (**ECBM**) [103], *a method proposed by Xuanyuan et al.* [105], *Conceptor* [15], ***ConceptTransformer*** (**CT**) [75], ***Concept-Centric Transformers*** (**CCT**) [38].

SENN [3]. It operates as a simple model that can be interpreted locally but not globally; it's authors achieve this through a regularization scheme that ensures that their model not only resembles a linear model but behaves locally as such. The idea of SENN is to extract high-order features from images and then represent them as interpretable basis concepts and assign each concept a local relevance score.

The model $f(x)$ is composed of three modules: (i) the concept encoder $h(x) : \mathcal{X} \to \mathcal{Z}$ included in $\mathbb{R}^k$, where k should be a small number to keep the explanation simple and understandable; (ii) the relevance parameterizer $\theta(x)$, where $\theta_i : \mathcal{X} \to \mathbb{R}^m$ and $\theta_i(x) \in \mathbb{R}^m$, is a vector that corresponds to the relevance of the ith concept with respect to each output dimension m; (iii) the aggregator $g(\theta_1(x)h_1(x), \ldots, \theta_k(x)h_k(x))$ that aggregates the k concepts with respect to their relevance. The loss is composed of three terms:

$$\mathcal{L} = \mathcal{L}_y(f(x), y) + \lambda \mathcal{L}_\theta(f) + \xi \mathcal{L}_h(x, \hat{x}), \tag{12}$$

where $\mathcal{L}_y$ is a cross-entropy between the model prediction $f(x)$ and the ground truth y, $\mathcal{L}_\theta$ is the robustness loss that encourages the complete model to behave locally as a linear function on $h(x)$ with $\theta(x)$ parameters, making interpretation of both concepts and relevance scores immediate. The last term of the loss $\mathcal{L}_h$ is used to train the encoder $h(\cdot)$ in conjunction with the rest of the model. The terms λ and ξ are hyperparameters to adjust the importance of the terms. Only for the training phase, the decoder is also used $h_{dec}(\cdot) : \mathbb{R}^k \to \mathbb{R}^n$ associated with $h(\cdot)$, so that the input is approximately reconstructed as $\hat{x} = h_{dec}(h(x))$.

CBM [46]. The approach proposed by the authors of this method makes it possible to convert an end-to-end model into a concept bottleneck model. CBM is a network that, given an input $x \in \mathbb{R}^d$, first predicts a concept c and then uses this concept to obtain the prediction $y \in \mathbb{R}$. They achieve this by aligning an intermediate layer of the network to the size of the concepts and adding a regularization term to the loss function, which optimizes the generation of concepts. Concepts are attributes provided by the dataset, and consequently, it is necessary to have a dataset labeled with the concepts in order to train a CBM.

Generally, a CBM is considered in the form of $f(g(x))$, where $g : \mathbb{R}^d \to \mathbb{R}^k$ maps an input x to the k-dimensional space of concepts, i.e., it generates the concepts for input x, and $f : \mathbb{R}^k \to \mathbb{R}$ maps the concepts to the final prediction, i.e., starting from the concept generates y. Considering that g and f are two distinct modules, the authors propose several ways for their training and optimization of loss functions. Let $L_{C_j} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ be the loss function that measures the discrepancy between the predicted concept and the jth true concept, and let $L_Y : R \times \mathbb{R} \to \mathbb{R}_+$ be the loss that measures the discrepancy between the prediction and the true target. There are several ways to learn the $(\hat{f},\hat{g})$ model:

(1) *Independent Bottleneck*: having concept labels and target labels makes it possible to train f and g separately, optimizing $\hat{f} = \text{argmin}_f \sum_i L_Y(f(c^{(i)}); y^{(i)})$ and $\hat{g} = \text{argmin}_g \sum_{i,j} L_{C_j}(g_j(x^{(i)}); c_j^{(i)})$. In the use phase, f will not have true c as input but will have $\hat{g}(x)$ as input.
(2) *Sequential bottleneck*: train $\hat{g}$ first, then use $\hat{g}(x)$ to generate concepts and use it to train $\hat{f} = \text{argmin}_f \sum_i L_Y(f(\hat{g}(x^{(i)}); y^{(i)})$.
(3) *Joint Bottleneck*: where you minimize the weighted sum

$$\hat{f}, \hat{g} = \underset{f,g}{\text{argmin}} \sum_i \left[L_Y(f(g(x^{(i)})); y^{(i)}) + \sum_j \lambda L_{C_j}(g(x^{(i)}); c^{(i)}) \right],$$

where λ controls the tradeoff between concepts and task loss and $c^{(i)}$ is the true concept.

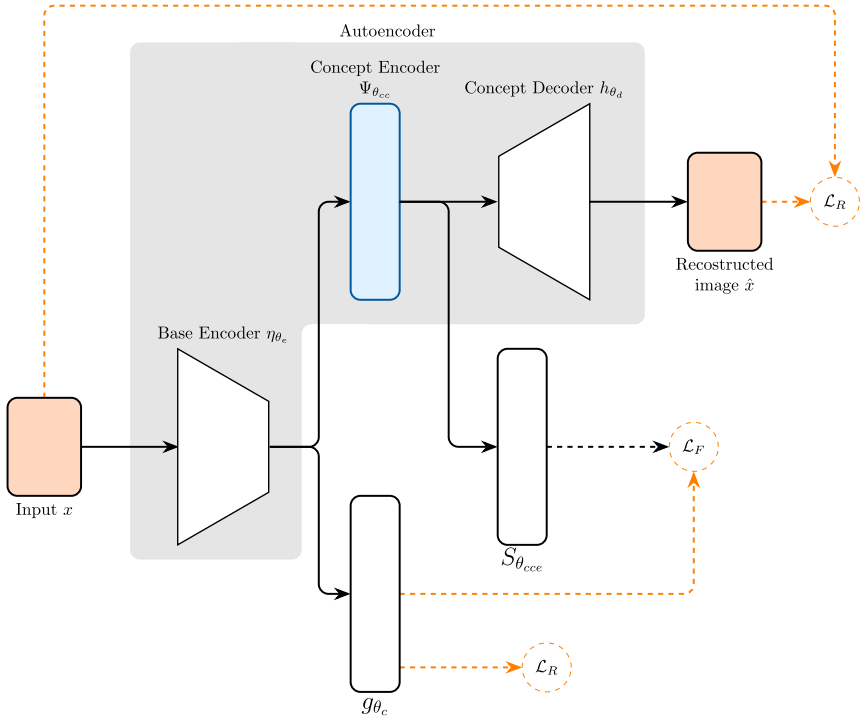


Fig. 5. Architecture replicated by us based on the network proposed by Sarkar et al. [80].

(4) *Standard Model*: ignore concepts and directly minimize $\hat{f}, \hat{g} = \text{argmin}_{f,g} \sum_i L_Y(f(g(x^{(i)})); y^{(i)})$.

A concept-based method by Sarkar et al. [80]. The method proposed by Sarkar et al. is a *weak interpretability* module that can be added to any DNN to generate explanations through contributing concepts during the training phase. The interesting aspect of this approach is that the authors have demonstrated its effectiveness even on complex datasets, such as ImageNet [24]. The concepts introduced by this method can be learned in an unsupervised, supervised, or self-supervised manner. The method involves the use of a generic framework, shown in Figure 5, to incorporate the ante-hoc explanation module.

The network is formalized as $f_\theta = \{\eta_{\theta_e}(\cdot), g_{\theta_c}(\cdot)\}$, where $\eta_{\theta_e}(\cdot)$ is the base encoder (feature extractor) that extracts representations to go as input to $g_{\theta_c}(\cdot)$, which is the classification function. In addition, the model consists of a $\Psi_{\theta_{cc}}(\cdot)$ concept encoder to learn an interpretable set of concepts $\{\psi^1, \ldots, \psi^C\}$, taking as input the features extracted by the base encoder $\eta_{\theta_e}(\cdot)$.

In general, the concepts are represented in low dimensionality. In order to visualize the concepts, the module consists of a decoder $h_{\theta_d}(\cdot)$, which has the task of reconstructing the image as $\hat{x}$ using the $\{\psi^1, \ldots, \psi^C\}$ learned concepts as input. In addition, this decoder encourages the model to represent image concepts in a semantically appropriate way. The concepts should explain the prediction made by the f_θ network, but since they do not contribute directly to the classification layer $g_{\theta_c}(\cdot)$, a classification layer $s_{\theta_{ccc}}(\cdot)$ is added, which, unlike $g_{\theta_c}(\cdot)$, takes $\{\psi^1, \ldots, \psi^C\}$ concepts as input. A fidelity loss $\mathcal{L}_F$ is used to encourage $g_{\theta_c}(\cdot)$ and $s_{\theta_{cce}}(\cdot)$ to produce the same output but using different inputs. In this way, indirectly, the concepts contribute to the classification. The

overall loss of the model can be written as follows:

$$\mathcal{L}_O = \mathcal{L}_C(y_i, \tilde{y}_i) + \alpha \mathcal{L}_R(x_i, \hat{x}_i) + \beta \mathcal{L}_F(f_\theta(x_i), s_{\theta_{cce}}(\Psi_{\theta_{ce}}(x_i))), \tag{13}$$

where $\mathcal{L}_C$ is the cross-entropy for the classification between the $\tilde{y}_i$ estimated by the classifier $g_{\theta_c}(\cdot)$ and the ground truth y_i, the reconstruction loss $\mathcal{L}_R$ is used to penalize the model if the concepts are not good enough to generate the reconstruction $\hat{x}_i$ of an input image x_i, an L_2 is used as the loss. The last term is the fidelity loss introduced earlier. By default, the proposed method behaves as an unsupervised concept method and uses only the loss $\mathcal{L}_O$ during the training phase. The terms α and β are hyperparameters to adjust the importance of the terms.

In case one has the availability of an auxiliary dataset of concepts, e.g., AwA2 [49], another loss term $\mathcal{L}_E(\Psi_{\theta_{ce}}(x_i), \alpha_{x_i})$ is introduced, where α_{x_i} are the concept annotations and $\Psi_{\theta_{ce}}(x_i)$ are the concepts estimated by the model. The total loss then will be $\mathcal{L}_O + \mu\mathcal{L}_E$. In the case where self-supervised training of concepts is to be adopted, one of several auxiliary tasks can be applied. The authors propose to use as an auxiliary task an $\zeta_{\theta_{SS}}(\cdot)$ classifier for the prediction of rotation angles r on the image. This classifier shares some parameters of the f_θ model. In the training phase, the loss of the auxiliary task $\mathcal{L}_{SS}(\zeta_{\theta_{SS}}(\cdot), r)$ is added; the total loss will be $\mathcal{L}_O + \gamma\mathcal{L}_{SS}$. In the case of deciding to do unsupervised training, the parameters μ and γ for supervised and self-supervised concepts, respectively, are set to zero.

CW [17]. The authors propose a method for understanding how a layer works, which is an alternative to batch normalization and can be applied to any deep model in any layer. The idea of CW is to normalize and align the axes of the latent space with concepts, so that each axis is representative of a concept learned from an auxiliary dataset of concepts. They use whitening to decorrelate and standardize the data in a latent space and an orthogonal rotation matrix to align concepts to axes. The authors show that, by taking any pre-trained deep model, replacing the *Batchnorm layers* with their CW layer, and training the model for only one additional epoch, they succeed in obtaining an interpretable model with the same accuracy and performance as the uninterpretable model. Furthermore, the authors show that the use of Concept Whitening at different layers allows for a semantic understanding: in the first layers, the model learns to extract point-wise details, while, in the last layers it learns more complex concepts such as shapes and objects.

CME [43]. The authors propose a framework applicable to any DNN to explain and improve the performance of the DNN. They define a DNN f decomposable into concepts, so that it can be well approximated by two functions p and q such that $f(x) = q(p(x))$. The function $p : \mathcal{X} \to \mathcal{C}$ is the input-to-concept function that maps the input $x \in \mathcal{X}$ to their concept representation $c \in \mathcal{C}$, while $q : \mathcal{C} \to \mathcal{Y}$ is the concept-to-output function, which maps concepts into the output space $\mathcal{Y}$. The major difference of CME, compared to other concept-based methods, concerns the use of multivariate concepts, i.e., nonbinary concepts; concepts are not only represented on the last layer but also concepts represented by low-level layers are used. In this way, functional relationships between concept and output are extracted and shown.

OpenCBM [90] Concept-based models involve the use of an auxiliary dataset of concepts. This means that a fixed set of concepts is defined for each class. The authors, in their work, propose an enhancement to CBM in which they employ an open vocabulary set of concepts and provide the user with the ability to delete or add any concept. To use an open vocabulary, the model makes use of a pre-trained language model such as CLIP [72].

ECBM [103] In the context of a CBM model, the process of correcting a concept does not inherently extend to closely related concepts. A limitation of CBM models is the inability to define high-level interactions between concepts and to quantify dependencies between them. In their work, the authors propose an energy-based CBM model, which involves calculating the joint energy

between the input concept and class. The correction, prediction, and quantification of dependencies are represented by conditional probabilities derived from the combined energy functions.

Since concept-based methods, as well as prototype-based methods, have demonstrated their high efficiency and effectiveness on deep architectures such as CNNs, other authors have also explored the use of concepts in other deep architectures, such as GNNs and DMs.

Method proposed by Xuanyuan et al. [105]. The authors propose using concepts to interpret GNNs because GNNs are difficult networks to interpret and are always used as black boxes without analyzing their internal behavior. Generally, explanations about GNNs are local and post-hoc, whereas with the method proposed by Xuanyuan et al., the explanation is done with global concepts generated internally by the network.

The main contributions of this paper are to (i) demonstrate that the neurons of a GNN can specialize on concepts and act as detectors of them, and also demonstrate that the neurons of a GNN lend themselves well to representing the concepts formulated as logical compositions of neighborhood properties; (ii) qualitatively evaluate the concepts detected by each neuron in the GNN; and (iii) demonstrate that they can generate explanations of single interpretable concepts supported by logical descriptions.

Their method is to perform a compositional concepts search on the activation space of a GNN. Then, given a graph $G \in \mathcal{G}$, the authors represent concepts C as functions $C : \mathcal{G} \rightarrow \{0,1\}^{|V|}$ that output binary masks and indicate whether a node is part of a concept. Their goal is to determine whether a neuron is representative of a concept using a metric similar to an ***Intersection Over Union*** (**IOU**) [73], which represents the divergence between the concept mask C and the activation of the kth neuron $H_G^l[k,:]$ stripped using τ_k. The search objective becomes the process of finding a concept that maximizes a score $f(C,k)$ for the neuron k, as in the following equation:

$$\underset{C}{\operatorname{argmax}} f(C,k)\ s.t.\ f(C,k) = \min_{\tau_k} |\mathcal{D}|^{-1} \sum_{G \in \mathcal{D}} Div\big(C(G), H_G^l[k,:], \tau_k\big), \tag{14}$$

where $\mathcal{D}$ is the training set of graphs. Then, to obtain global explanations, the authors align the explanation with the concepts by searching for $\mathcal{E}_{glob} = \{\operatorname{argmax}_{C \in \mathcal{C}} f(k,C) \mid k \in S_{glob}\}$, which is the set of the highest scoring concepts for neurons in S_{glob}. For more details, we advise the reader to refer to the paper by Xuanyuan et al. [105].

Conceptor [15]. The authors propose a concept-based method for interpreting a diffusion model that starts with text and generates an image. In general, diffusion models that generate images from text do not provide details about the intermediate states of the denoising process, not semantically motivating which concepts most influenced the generation of the image. Conceptor is a new method that extracts the textual concepts that the DM uses in its internal representation to generate an output image. The authors have shown that, because of the interpretability provided by Conceptor, we are able to observe model reasoning structures that are often inconsistent with human reasoning structures. This proves that, generally, deep models obtain output by making associations between features that a human, in most cases, would not have done. Conceptor uses a state-of-the-art ***Stable Diffusion*** (**SD**) model [76], that employs a ***Denoising Diffusion Probabilistic Model*** (**DDPM**) [36]. Conceptor's goal is to interpret the internal representation of a DM ε_θ through the use of a vocabulary $\mathcal{V}$ of expressive and diverse concepts c, containing roughly N=50,000 human-understandable tokens. Conceptor learns a decomposition of the concepts using the $\mathcal{V}$ vocabulary. This decomposition is realized as a pseudo-token $w^* \notin \mathcal{V}$, constructed as the following linear combination:

$$w^* = \sum_{i=1}^{n} \alpha_i w_i \quad s.t. \quad w_i \in \mathcal{V}, \alpha_1, \ldots, \alpha_n \geq 0, \tag{15}$$

where $n << N$ is a hyperparameter that determines the number of tokens to use in the combination. The coefficient α is obtained from a 2-layer MLP that takes as input each word embedding vector w, as shown in the following equation:

$$\forall w \in \mathcal{V} \; : \; \alpha = f(w) = \sigma(W_2(\sigma(W_1(w)))). \tag{16}$$

Then Equation (16) can be rewritten as $w_N^* = \sum_{i=1}^{N} f(w_i)w_i$, where w_N^* are all the decomposed tokens of the vocabulary, and w^* is obtainable by taking the top n tokens from w_N^*. The authors add a regularization loss $\mathcal{L}_{sparsity}$ to encourage the pseudo-token w_N^* to be dominated by these top n tokens, as in the following equation:

$$\mathcal{L}_{sparsity} = 1 - cosine(w^*, w_N^*). \tag{17}$$

The total loss is then as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{sparsity}\mathcal{L}_{sparsity}, \tag{18}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss used in the DDPM model and $\lambda_{sparsity}$ is a hyperparameter to adjust the importance of the term $\mathcal{L}_{sparsity}$.

CT [75]. The authors of this paper propose the use of concepts to ensure interpretability in a network based on the attention mechanism. Their proposed module is designed to be a classification head in a generic deep architecture. CT uses three types of input: (i) the input image x used as embedding of P visual patches that are linearly projected and concatenated into a query matrix $Q \in \mathbb{R}^{P \times d_m}$; (ii) concepts represented as embeddings encoded in a key matrix $K \in \mathbb{R}^{C \times d_m}$; (iii) concepts linearly projected using a value projection matrix and concatenated to obtain the value matrix $V \in \mathbb{R}^{C \times d_m}$.

The module performs the cross-attention between the query matrix Q and the key matrix K. From the cross-attention operation, attention weights are then generated between each patch-concept pair, combined into an attention matrix $A = [\alpha_{pc} \in \mathbb{R}^{p \times c}]$, as illustrated in the following equation:

$$\alpha_{pc} = softmax\left(\frac{1}{\sqrt{d_m}} QK^T\right)_{pc} \quad with \quad p = 1, \ldots, P, \quad c = 1, \ldots, C, \tag{19}$$

where P and C are the numbers of patches and the numbers of concepts, respectively.

The final output of CT is the product obtained by multiplying the attention map A, the value matrix V, and the output matrix $O \in \mathbb{R}^{d_m \times n_c}$, which projects onto the logit n_c (not normalized) output classes and averages the patches:

$$logit_i = \frac{1}{P} \sum_{p=1}^{P} [AVO]_{pi} \quad with \quad i = 1, \ldots, n_c. \tag{20}$$

The authors of the paper [75] describe a single-head attention model through the above formulas, but, in their experiments, they use the multi-head version of Vaswani et al. [97]. The method was then tested on datasets such as *Caltech-UCSD Birds-200-2011* (CUB-200-2011) [98], MNIST Even/Odd [6], and ***attribute Pascal and Yahoo*** (**aPY**) [29].

CCT [38] In this paper, the authors explored the use of a shared memory in which the various modules of the proposed framework compete to write to it. They added a broadcast communication mechanism, which updates each module when a specific module has been written to the shared memory. This mechanism, inspired by the **Shared Global Workspace** (**SGW**), allows computational modules to communicate effectively with each other. The functionality was tested in an interpretable model inspired by CT, with the difference that the CCT framework generalizes the approach by not using image patches.

Table 4. Other Interpretability Methods for DNNs Ordered by Publication Date

Authors	Method	Model	Year
Liangzhi Li et al. [52]	SCOUTER	Transformers	2021
Moritz Böhle et al. [8]	B-cos Networks	CNN	2022
Eslam Mohamed Bakr et al. [5]	ToddlerDiffusion	DM	2023
Huang et al. [40]	ProtoCBM	CNN	2024
Wan et al. [99]	Semantic Prototype Analysis Network (SPANet)	CNN	2024

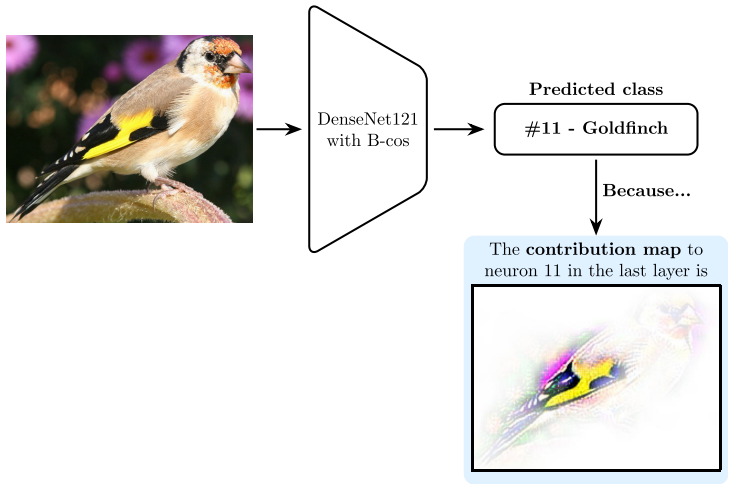


Fig. 6. Example of B-cos on a test image of CUB-200-2011 dataset. B-cos [8] can show the contribution map for each neuron of each layer of the model, it shows that to classify, the model focused mainly on the peculiar wings of the bird.

4.3 Other Interpretability Methods

This section discusses other work on interpretability methods for DNNs that do not rely on the use of prototypes or concepts but propose alternative strategies in order to make a model interpretable. All the methods that will be explained in this section are collected in Table 4. Figure 6 shows an example of an application of B-cos [8], to show what it means to interpret a model using an alternative method to both prototypes and concepts.

B-cos [8]. The authors propose a new approach to increase the interpretability of DNNs by promoting weights-input alignment during training. They propose to replace the linear transformations in a DNN with their B-cos transform. They designed the B-cos layer so that it can be compatible with common deep architectures such as ***Visual Geometry Group*** (**VGG**) networks [86], ***Residual neural Network*** (**ResNet**) [34], *InceptionNets* [89], and ***Densely connected convolutional Networks*** (**DenseNets**) [39], while still maintaining similar performance on ImageNet [24]. The authors define their B-cos transform as a variation of a linear transform; a linear transform is represented in the following equation:

$$f(x;w) = w^T x = \|w\| \, \|x\| \, c(x,w) \quad with \quad c(x,w) = cos(\angle(x,w)). \tag{21}$$

While this other equation shows the B-cos transformation proposed by the authors:

$$\text{B-cos}(x;w) = \|\hat{w}\| \, \|x\| \, |c(x,\hat{w})|^B \times sgn(c(x,\hat{w})), \tag{22}$$

where B is a hyperparameter, the hat operator scales $\hat{w}$ to the unit norm, and *sgn* denotes the sign function. It is noted that if $B = 1$, the B-cos transform is equivalent to the linear transform with $\hat{w}$.

The authors state that these changes in the function are useful for three reasons. First, they bound the output of B-cos neurons. Second, by increasing the B exponent, it is possible to further suppress the output for misaligned weights, and the respective B-cos unit can produce outputs close to its maximum. These two properties allow the B-cos transform to provide explanations based on similarities; a sample is well classified if it is aligned with the corresponding vector of weights. The third property is that, despite these introduced changes, the B-cos transform retains a property of the linear transform, namely, that a sequence of B-cos transforms can be faithfully represented as a single linear transform d.

SCOUTER [52]. The authors propose an attention-based interpretability method that can be used both to generate a positive explanation of class membership and to generate a negative explanation of nonmembership. They use *explainable Slot attention* (xSlot) module [60] as a mechanism that uses self-attention to learn how to represent individual objects. SCOUTER's major contribution is the addition of the regularization term ℓ_{Area} to the loss function, which allows them to limit the coverage area of a single slot, since, without this term, a slot could also cover too large an area. The final loss used by SCOUTER is as follows:

$$\ell_{SCOUTER} = \ell_{CE} + \lambda \ell_{Area}, \tag{23}$$

where ℓ_{CE} is the cross-entropy loss and λ is a hyperparameter to adjust the importance of the term ℓ_{Area}.

The authors propose to train two models to be used in parallel: $SCOUTER_{+}$, which generates explanations for belonging to a given class, and $SCOUTER_{-}$, which generates explanations for not belonging to a given class.

ToddlerDiffusion [5]. The authors propose a method for interpreting the intermediate stages of a diffusion model that operates in the image-to-image domain for generating synthetic images. The interpretability of the intermediate stages of a diffusion model allows users to be able to interact with the intermediate stages and edit them in order to influence the model to generate new synthetic images. Unlike traditional diffusion models that, in the denoising stage, at each step t, estimate an ϵ^* amount of noise to be removed, thus making it impossible to understand and impossible to manipulate an intermediate stage of this stage. ToddlerDiffusion makes the denoising stage transparent by decomposing this process into three stages: *Stage Abstract Structures* and *Stage Palette*, and combining both stages in the third *Stage Detailed Image*, to generate a single synthetic image.

Stage Abstract Structures aims to generate abstract contours $\mathcal{S} \in \mathbb{R}^{H \times W \times 3}$ starting from pure noise. The authors formulate sketch generation as a mapping function that goes from the domain $y = x_T$ to the domain $x = x_0$. The forward process is formulated with the following equation:

$$x_t = \alpha_t \mathcal{F}_d(x_0, t) + (1 - \alpha_t)y + \alpha_t^2 \epsilon_t, \tag{24}$$

where α_t is a weight factor between the two domains, α_t^2 is the variance of the noise, and $\mathcal{F}_d(x_0, t)$ is a dropout function that takes the groundtruth sketch x_0 and the current timestep t and generates a more sparse version of x_0, performing a white pixel mask. In general, in diffusion models, y is a Gaussian distribution, whereas, in their work, the authors set y as a black image, to which they then add brighter points during convergence to obtain the contours.

After generating the sketch $\mathcal{S}$, the authors propose the *Stage Palette* to generate color information. The formulation follows the stage abstract structure:

$$x_t = \alpha_t \mathcal{F}_p(x_0, \mathcal{K}_t, \mathcal{J}_t) + (1 - \alpha_t)y + \alpha_t^2 \epsilon_t, \tag{25}$$

where α_t is the weight factor between the two domains, α_t^2 is the variance of the noise, and $\mathcal{F}_p(x_o, \|_t, \mathcal{J}_t)$ is the pixelization function for the GT palette using a kernel $\mathcal{K}_t$ and stride $\mathcal{J}_t$.

The third and final stage, *Detailed Image*, uses a fusion of sketches and palettes as a starting point. The formulation is as follows:

$$x_t = \alpha_t x_0 + (1 - \alpha_t)y + \alpha_t^2 \epsilon_t. \tag{26}$$

As a loss function, the authors propose an adaptation of the *Variational Lower BOund* (ELBO):

$$\begin{aligned}\mathcal{L}_{ELBO} = &- \mathbb{E}_q(D_{KL}(q(x_T|x_0, y)||p(x_T|y)) + \sum_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0, y)||p_\theta(x_{t-1}|x_t, y)) \\ &- \log p_\theta(x_0|x_1, y)),\end{aligned} \tag{27}$$

where p_θ is an approximation of functions to predict starting from x_t and D_{KL} is the *Kullback–Leibler Divergence* function.

ProtoCBM [40] In the context of concept-based models, the process of mapping from input to concept remains an area of uncertainty because of its inherent black-box nature. To overcome this limitation, the authors proposed an improved CBM in which, instead of the black box, prototypes are used as in ProtoPNet, and then the score maps obtained by using the prototypes are used as input for the Concept Predictor as CBM. In addition, they developed a metric to evaluate the reliability of concepts within CBM models. This metric uses a reliability score to assess the reliability of concepts, thus contributing to the advancement of the field.

SPANet [99] In this work, the authors proposed the use of both concepts and prototypes for the creation of more inclusive and human-oriented explanations. They developed an interpretable network that uses semantic prototypes, which are the result of combining prototypical parts and semantic concepts. The network consists of three modules: the Semantic Attachment Module, which encodes the input image and the list of concepts; the Prototype Recognition Module, which compares the features generated by the previous module with the stored prototypes; and the Reconstruction Module facilitates the visualization of semantic prototypes, which represent the amalgamation of concepts and prototypes.

5 Discussion

In this section, we want to offer readers a discussion of some key points of interpretability. As the number of deep black-box models grows in direct proportion to the need to obtain explanations, there is a need to gather interpretability methodologies for deep models into a single SLR.

As you can see from the previous sections, this SLR has discussed methods of interpretability based on prototypes, concepts, and others. We want to evaluate interpretability for each of these methods. Here, we discuss the strengths and weaknesses of the different types. Table 5 shows the properties of interpretability for each method discussed. The differences between the different types of interpretability are discussed using the properties defined in the Section 2.2. In addition, we want to offer insights into the uses of interpretability.

5.1 Interpretability of Prototypes

In general, prototype-based methods benefit from strong interpretability because the prototype layer, which is the interpretable component of the network, contributes directly to model prediction. As shown in Table 5:

— **Selectivity**: is guaranteed by the hyperparameter m, which defines the number of prototypes learned by the network for each class. If an adequate number of prototypes is chosen, then the explanations of the model will be selective; on the other hand, if a large number of

Table 5. Overview of the Five Properties of Interpretability Guaranteed by Different Models

Category	Method	Selectivity	Contrastive	Social	Truthful	Focused on the abnormal
Prototype	Oscar Li et al. [53]	O	✓	✓	✓	✗
	ProtoPNet [16]	O	✓	✓	✓	✗
	NP-ProtoPNet [87]	O	✓	✓	✓	✗
	Nauta et al. [66]	O	✓	✓	✓	✗
	ProtGNN [110]	O	✓	✓	✓	✗
	ViT-NeT [44]	O	✓	✓	✓	✗
	HQProtoPNet [100]	O	✓	✓	✓	✗
	PGIB [82]	O	✓	✓	✓	✗
	TCPL [30]	O	✓	✓	✓	✗
	DProtoNet [69]	O	✓	✓	✓	✗
Concept	SENN [3]	O	✗	O	O	O
	CBM [46]	O	✗	O	O	O
	CW [17]	O	✗	O	O	O
	CME [43]	O	✗	O	O	O
	CT [75]	O	✗	O	O	O
	Sarkar et al. [80]	O	✗	O	O	O
	Xuanyuan et al. [105]	O	✗	O	O	O
	Conceptor [15]	O	✗	O	O	O
	CCT [38]	O	✗	O	O	O
	OpenCBM [90]	O	✗	O	O	O
	ECBM [103]	O	✗	O	O	O
Other	SCOUTER [52]	✗	✗	✓	✓	✗
	B-cos [8]	O	✗	✓	O	✗
	ToddlerDiffusion [5]	O	✗	✓	✓	✗
	ProtoCBM [40]	O	✗	O	✓	O
	SPANet [99]	O	✓	✓	✓	✗

The following symbols are used for: (✗) non-guaranteed property; (✓) guaranteed property; (O) guaranteed property if a condition is met; in general, the condition for Prototype and Others methods depends on the values of their hyperparameters, while for Concept methods the condition depends on the auxiliary dataset used.

prototypes is chosen, then there is a risk of having redundant prototypes that are used in explanation, which would confuse users.

— **Contrastive**: a prototype is a template that the model uses to compare with an image it wants to evaluate. Consequently, the use of prototypes ensures an explanation based on contrastiveness.

— **Social**: the explanations given by prototype-based methods take advantage of prototypical images to motivate the choice of the template; consequently, these images are easily understood even by users who are not experts in a particular field.

— **Truthful**: although the explanations given by methods using prototypes tend to have high selectivity, they actually succeed in ensuring good truthfulness because the explanations granted by the model clearly motivate all model choices. Thus, even though there are few prototypes, they manage to capture all the significant aspects of each class that the model uses to make predictions.

— **Focused on the abnormal**: this is the only property not guaranteed by prototypes, since the explanations are based only on patterns that commonly define a class and not on abnormalities in that class.

The explanations provided by prototype-based methods are not easy to obtain; the difficulty faced by the different authors concerns, in most cases, the choice of a good number of prototypes, since choosing too many prototypes runs the risk of having redundant or "background" prototypes, while choosing too few means ignoring fundamental patterns. The authors address this problem by using a pruning algorithm, which is a phase of removing redundant or "background" prototypes after the training phase.

5.2 Interpretability of Concepts

The proposed concept-based methods generally enjoy *strong interpretability* as is the case with all methods except the method of Sarkar et al. that has *weak interpretability* because the concepts are used only to influence loss but do not contribute to the output in inference. As shown in Table 5:

- **Selectivity**: considering that most methods use an auxiliary dataset of concepts, selectivity is given by the number of concepts for each class in the dataset. One exception is the method of Sarkar et al., in which they also provide unsupervised training for concepts and, in this case, the selectivity is given by the width of the layer of concepts. If we have in the dataset a few concepts that are representative of the problem for each class, we will have high selectivity; otherwise, we will have little selectivity in explanation.
- **Contrastive**: concept-based methods do not rely on comparisons but provide explanations by extracting the concepts from the input samples; consequently, the contrastivity of these methods is low.
- **Social**: explanations generated by concept-based methods are easy to understand if the auxiliary dataset of concepts is easy to understand; however, if the dataset has complex concepts, then the explanation generated by the model will be equally complex and understandable only by experts in the field. Generally, the proposed works use datasets of relatively simple concepts.
- **Truthful**: is guaranteed because the model's explanations are closely related to the concepts provided by the auxiliary dataset, so the model will always be comprehensive with respect to the concepts in the dataset.
- **Focused on the abnormal**: generally, concept-based methods do not provide explanations based on abnormal cases since the concept is a general and common representation of a frequently appearing feature. However, if one constructs a dataset with concepts representing abnormal features, one could cause the model to generate explanations based on the abnormal.

Concept-based methods, such as Conceptor [15], can prove to be a powerful tool for bias discovery and reduction in deep models. This method manipulated the concepts used in the internal representation of the DM to find out which concepts, when deactivated, reduced the bias in image generation. Other methods, such as CW [17], are applied to all layers of a deep model, showing that, for semantic understanding, in the first layers, the model learns to extract point details and little generality, while in the last layers, it learns more complex concepts such as shapes and objects.

5.3 Other Methods

The other interpretability methods guarantee the properties of Table 5, as follows:

- **Selectivity**: in *B-cos* increasing its hyperparameter B results in greater selectivity in the explanations. *SCOUTER* has low selectivity because it provides heat maps as explanations, positive and negative, for each class. Moreover, it may be the case that these explanations are also contradictory. *ToddlerDiffusion* generates a number of interpretable feature maps equal to the number of states t of the diffusion model, which can also be large, the selectivity

depending on t. In *ProtoCBM* selectivity depends on the number of concepts learned, while in *SPANet* selectivity depends on the number of semantic prototypes learned.
— **Contrastive**: among the methods discussed, only SPANet allows a contrastive explanation because it uses semantic prototypes.
— **Social**: the explanations generated by *B-cos* are visual, so they are therefore understandable to all kinds of users. *SCOUTER*'s explanations are fairly easy to understand by a wide range of users, as it generates heatmaps that are superimposed on the source image. The explanations generated by *ToddlerDiffusion* are understandable to a wide range of users because they have made it possible to visualize the hidden stages of a diffusion model, which were previously incomprehensible to anyone. *ProtoCBM* uses Prototypes only to generate the concepts, so the social property is subject to the same variability as the concepts. *SPANet* provides understandable explanations to all types of users, exploiting both the prototypes and the concepts.
— **Truthful**: *B-cos* provides high truthfulness when this interpretability method is used on multiple layers so as to generate multiple explanations at different depths. *SCOUTER* generates positive and negative explanations for all classes to try to corroborate explanations consistent with the classification made, so *SCOUTER* has high truthfulness. The explanations generated by *ToddlerDiffusion* have high truthfulness because the number of transparent feature maps generated in the hidden stages allows for exhaustive explanations of the image generation pipeline, providing explanations of both shapes across edges and also explanations of color regions. *ProtoCBM*, in addition to the classic concepts, also provides location maps that make the explanations exhaustive. *SPANet* using exhaustive semantic prototypes ensures high truthfulness.
— **Focused on the abnormal**: Potentially only ProtoCBM could exploit concepts representing abnormal cases in order to provide this kind of explanation. However, none of the four other methods focus on abnormal explanations, as they all provide explanations based on more common cases.

These methods show that explanations can also be obtained through alternative approaches; these models generally focus on generating generic explanations; none of these methods generate explanations based on comparisons with a reference case; and, they do not focus on the abnormal. These methods are good alternatives for focused problems, such as in image generation, where using *ToddlerDiffusion* one is able to observe the intermediate stages of a diffusion model in a transparent way.

5.4 Scientific Discovery through Interpretability

Making a model interpretable not only changes the approach of users as they work or use AI models, but leads to a new way of doing research. Having an interpretable model means being able to follow the "reasoning" of a network or observe phenomena that, without interpretability, remained obscure to users' eyes. As discussed in previous sections, in high-risk contexts, having an interpretable model is indispensable. In [88], the authors propose the use of a prototype-based network for early glaucoma diagnosis, which not only provides excellent performance in terms of accuracy but also generates user confidence as it explains the decision made, allowing clinicians to check whether the model choices were correct. In [68], the authors propose a method for evaluating the prototype-based networks for breast cancer detection through mammography. Using their prototype quality metrics, they are able to determine which prototype network is best to use for this type of problem. Evaluating prototypes allows users to understand which prototypes are the key classification contributors and which ones do not contribute definitively. In [104], the authors demonstrate the ability to effectively determine COVID-19 patients using a prototypical network.

In [35], they use a prototypical model for a regression task and use it to predict brain age. The use of these methods is not only useful in clinical contexts, for example in [91], they develop a concept-based method for interpreting the language models such as BERT and GPT2. It is not only high-risk contexts that benefit from explainable models; an example is the work of Cremades et al. [23], in which, using an explainable model, they are able to identify important structures in complex problems in classical physics. Another example is the work of Cranmer et al. [22], in which through their approach they are able to extract equations associated with a GNN model; in their example, they are able to generate a new equation that predicts the concentration of dark matter from cosmic structures, thus enabling the discovery of new physical principles.

5.5 General Considerations

In this SLR, we have always focused on the descriptions of the proposed methodologies, avoiding describing the results of the methods in terms of performance, such as accuracy or other evaluation metrics. However, the belief in the literature that there is a tradeoff between interpretability and accuracy is not to be overlooked. There is a belief that making a model deeply interpretable means reducing its performance [33]. While writing this article, having considered all the methods discussed, we agree with Rudin's work [77], concluding that this tradeoff is not always true. The proposed methods succeed in providing both interpretability and high accuracy.

Based on our RQ in Section 3.3, our formal responses to the RQs are as follows:

- RQ1. There are ante-hoc methods for interpreting DL models, and in this article we analyze the methods found.
- RQ2. The models discussed in this article are all ante-hoc methods of interpretability. There is confusion in the literature because there are works that use the term interpretability, but they are post-hoc methods.
- RQ3. Most of the interpretability methods found are proposed on experimental datasets, so they are used in laboratory settings. Few works use natural image datasets, such as ImageNet [24], which suggests that potentially these methods can also be used in real-world contexts.
- RQ4. The methods found provide user-friendly explanations, because they are either based on visual explanations that are easy to understand or are based on concepts predetermined by humans. So they are not machine-oriented explanations, but they are human-oriented explanations.
- RQ5. Our research found that methods have been proposed to interpret the most widely used deep architectures such as CNNs, GNNs, Transformers, *Diffusion probabilistic Models* (DMs).

Another point we would like to make concerns the use of clean and simple datasets with which the various authors have demonstrated the performance of their models. We believe that a good challenge to address would be to adapt and test the same methods on real more complex datasets.

Another aspect we want to bring to the attention of readers in these discussions is the quality of the information content of transformers. In general, in some work using transformers, attention is also used to explain a model's choices. The use of the attention mechanism is justified because it is assumed to contain information that the model considers important. The work of Serrano et al. [83] shows that in NLP tasks, in which textual input is worked on, attention does not always contain information content that is useful in explaining the model. More importantly, through several experiments, they show that its content can be easily manipulated.

A question worth asking is: "Given the difficulty in creating interpretable DL models, is it worth creating them for non-high-risk contexts?" Obviously, it depends a lot on the context. Although some issues are not high-risk, it may be useful to interpret the model to increase user

confidence. It could also be useful as a debugging tool for researchers themselves, for example, in the context of using deep model ensembles. Recall that even if these are not high-risk issues, governments are moving in the direction of mandating the development and use of interpretable models.

6 Conclusion and Future Work

In this SLR, we have demonstrated the importance of interpreting DNNs and highlighted the efforts of legislative bodies to regulate the use of AI through the development of interpretable AI. In addition, we motivated the use of interpretable approaches, especially in high-risk contexts where it is dangerous to rely on black-box models, which, by not providing reasons for their predictions, turn out to have low reliability. We also demonstrated a clear difference between explainability and interpretability in order to help researchers gain clarity, and as a result of our research, we thought it appropriate to distinguish interpretability methods into strong and weak interpretability.

In this SLR, 26 main studies were selected after applying some precision criteria. The confusion that permeates in the scientific community between explainability and interpretability led us to exclude many scientific articles that improperly use the term interpretability, as they did not propose concrete ante-hoc methods but proposed post-hoc methods of analysis, i.e., explainability. Moreover, the small number of articles found also highlights the difficulty in finding and developing new ante-hoc methods for deep models.

In this SLR, we thought it appropriate to focus on the innovative methodologies proposed to make deep models interpretable, rather than comparing them with each other through score metrics. We point out that one of the problems of interpretability methods is that they are heterogeneous. Therefore, they can hardly be compared with one and the same scoring metric; in fact, the authors proposing these papers themselves hardly compare their method with methodologies of a different interpretability class.

We thought it appropriate to analyze interpretability methods by dividing them into prototype-based and concept-based methods, and we also thought it appropriate to point out the existence of other interpretability methods that differ from prototype-based and concept-based methods.

These initially proposed interpretability models were created for CNNs, which are the most popular deep model; we have shown that, over the years, they have also been adapted to more complex deep architectures, such as GNNs and Transformers. In addition, we have observed that researchers, instead of proposing methodologies, alternatives to prototypes and concepts, are more inclined to propose improved versions of already existing methods based on prototypes and concepts. A future direction might be to draw inspiration from these methods to try to introduce them into newer deep architectures, such as generative models.

In this study, it was observed that, with the exception of concept-based approaches, interpretability methods do not employ anomaly-based explanations. So as a future direction, researchers could focus on explaining abnormal cases.

A question we asked during this SLR concerns how reliable these methods can be in real life, as they have been shown to be effective by training the models on simple or unchallenging datasets. So as a future direction, one could try to expand the study of these methods in real-life settings, using datasets affected by noise and environmental phenomena.

In conclusion, we believe that this article can be useful to researchers who are approaching the study of interpretability in deep models, and we think it can be a good starting point to get a clear overview of the work developed so far. These and subsequent methods developed in the future can be of enormous use in many contexts, both high-risk, such as medical or legal contexts, and also in social contexts where the user prefers transparent systems.

References

[1] Imran Ahmed, Gwanggil Jeon, and Francesco Piccialli. 2022. From artificial intelligence to explainable artificial intelligence in Industry 4.0: A survey on what, how, and where. *IEEE Transactions on Industrial Informatics* 18, 8 (2022), 5031–5042.

[2] Gulsum Alicioglu and Bo Sun. 2022. A survey of visual analytics for explainable artificial intelligence methods. *Computers & Graphics* 102 (Feb. 2022), 502–520. https://doi.org/10.1016/j.cag.2021.09.002

[3] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 7786–7795.

[4] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One* 10, 7 (2015), e0130140.

[5] Eslam Mohamed Bakr, Liangbing Zhao, Vincent Tao Hu, Matthieu Cord, Patrick Perez, and Mohamed Elhoseiny. 2023. ToddlerDiffusion: Flash interpretable controllable diffusion model. arXiv:2311.14542. Retrieved from https://arxiv.org/abs/2311.14542

[6] Pietro Barbiero, Gabriele Ciravegna, Francesco Giannini, Pietro Lió, Marco Gori, and Stefano Melacci. 2022. Entropy-based logic explanations of neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 6 (June 2022), 6046–6054. DOI: https://doi.org/10.1609/aaai.v36i6.20551

[7] Joseph Berkson. 1953. A statistically precise and relatively simple method of estimating the bio-assay with quantal response, based on the logistic function. *Journal of the American Statistical Association* 48, 263 (1953), 565–599.

[8] Moritz Bohle, Mario Fritz, and Bernt Schiele. 2022. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. IEEE, 10319–10328. DOI: https://doi.org/10.1109/CVPR52688.2022.01008

[9] Leo Breiman (Ed.). 1984. *Classification and Regression Trees*. Wadsworth International Group, Belmont, CA, USA.

[10] Bruce G. Buchanan and Edward H. Shortliffe. 1984. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (the Addison-Wesley Series in Artificial Intelligence)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[11] Vanessa Buhrmester, David Münch, and Michael Arens. 2021. Analysis of explainers of black box deep neural networks for computer vision: A survey. *Machine Learning and Knowledge Extraction* 3, 4 (2021), 966–989.

[12] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for HealthCare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*. ACM, New York, NY, USA, 1721–1730. DOI: https://doi.org/10.1145/2783258.2788613

[13] Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuveer M. Rao, et al. 2017. Interpretability of deep learning models: A survey of results. In *Proceedings of the 2017 IEEE Smartworld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*. IEEE, 1–6. DOI: https://doi.org/10.1109/UIC-ATC.2017.8397411

[14] Angelos Chatzimparmpas, Rafael M. Martins, Ilir Jusufi, and Andreas Kerren. 2020. A survey of surveys on the use of visualization for interpreting machine learning models. *Information Visualization* 19, 3 (2020), 207–233.

[15] Hila Chefer, Oran Lang, Mor Geva, Volodymyr Polosukhin, Assaf Shocher, michal Irani, Inbar Mosseri, and Lior Wolf. 2024. The hidden language of diffusion models. In *Event 12th International Conference on Learning Representations ICLR*, Hybrid, Vienna, Austria, 20 pages.

[16] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. 2019. This looks like that: Deep learning for interpretable image recognition. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, Article 801, 12 pages.

[17] Zhi Chen, Yijie Bei, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence* 2, 12 (2020), 772–782.

[18] Keyang Cheng, Ning Wang, and Maozhen Li. 2021. Interpretability of deep learning: A survey. In *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery*, Hongying Meng, Tao Lei, Maozhen Li, Kenli Li, Ning Xiong, and Lipo Wang (Eds.). Springer International Publishing, Cham, 475–486. https://doi.org/10.1007/978-3-030-70665-4_54

[19] Giovanni Ciatto, Federico Sabbatini, Andrea Agiollo, Matteo Magnini, and Andrea Omicini. 2024. Symbolic knowledge extraction and injection with sub-symbolic predictors: A systematic literature review. *ACM Computing Surveys* 56, 6 (March 2024), 161:1–161:35. DOI: https://doi.org/10.1145/3645103

[20] William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on International Conference on Machine Learning (ICML'95)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 115–123.

[21] Roberto Confalonieri, Ludovik Coba, Benedikt Wagner, and Tarek R. Besold. 2021. A historical perspective of explainable artificial intelligence. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11, 1 (2021), e1391.

[22] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. 2020. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.). Vol. 33. Curran Associates, Inc., 17429–17442. https://proceedings.neurips.cc/paper_files/paper/2020/hash/c9f2f917078bd2db12f23c3b413d9cba-Abstract.html

[23] Andrés Cremades, Sergio Hoyas, Rahul Deshpande, Pedro Quintero, Martin Lellep, Will Junghoon Lee, Jason P. Monty, Nicholas Hutchins, Moritz Linkmann, Ivan Marusic, et al. 2024. Identifying regions of importance in wall-bounded turbulence through explainable deep learning. *Nature Communications* 15, 1 (May 2024), 3864. DOI: https://doi.org/10.1038/s41467-024-47954-6

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 248–255. DOI: https://doi.org/10.1109/CVPR.2009.5206848

[25] Weiping Ding, Mohamed Abdel-Basset, Hossam Hawash, and Ahmed M. Ali. 2022. Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey. *Information Sciences* 615, C (Nov 2022), 238–292. DOI: https://doi.org/10.1016/j.ins.2022.10.013

[26] Filip Karlo Dosilovic, Mario Brcic, and Nikica Hlupic. 2018. Explainable artificial intelligence: A survey. In *Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO'18)*. IEEE, 0210–0215. DOI: https://doi.org/10.23919/MIPRO.2018.8400040

[27] European Parliament and Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council*. Official Journal of the European Union. Retrieved February 15, 2024 from https://eur-lex.europa.eu/eli/reg/2016/679/2016-05-04

[28] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. 2021. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences* 5, 6 (2021), 741–760.

[29] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. 2009. Describing objects by their attributes. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1778–1785. DOI: https://doi.org/10.1109/CVPR.2009.5206772 ISSN: 1063-6919.

[30] Junyu Gao, Xinhong Ma, and Changsheng Xu. 2024. Learning transferable conceptual prototypes for interpretable unsupervised domain adaptation. *IEEE Transactions on Image Processing* 33 (2024), 5284–5297. DOI: https://doi.org/10.1109/TIP.2024.3459626

[31] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA'18)*. IEEE, 80–89. DOI: https://doi.org/10.1109/DSAA.2018.00018

[32] Bryce Goodman and Seth Flaxman. 2017. European Union regulations on algorithmic decision making and a "Right to Explanation". *AI Magazine* 38, 3 (Sep 2017), 50–57. DOI: https://doi.org/10.1609/aimag.v38i3.2741

[33] David Gunning, Eric Vorm, Jennifer Yunyan Wang, and Matt Turek. 2021. DARPA's explainable AI (XAI) program: A retrospective. *Applied AI Letters* 2, 4 (Dec. 2021), e61. DOI: https://doi.org/10.1002/ail2.61

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. IEEE, 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[35] Linde S. Hesse, Nicola K. Dinsdale, and Ana I. L. Namburete. 2024. Prototype learning for explainable brain age prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, Hawaii, 7903–7913.

[36] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Proceedings of the 34th International Conference on Neural Information Processing Systems* . 6840–6851.

[37] Robert C. Holte. 1993. Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11, 1 (1993), 63–90. https://doi.org/10.1023/A:1022631118932

[38] Jinyung Hong, Keun Hee Park, and Theodore P. Pavlic. 2024. Concept-centric transformers: Enhancing model interpretability through object-centric concept learning within a shared global workspace. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, Hawaii, 4880–4891.

[39] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. IEEE, 2261–2269. DOI: https://doi.org/10.1109/CVPR.2017.243

[40] Qihan Huang, Jie Song, Jingwen Hu, Haofei Zhang, Yong Wang, and Mingli Song. 2024. On the concept trustworthiness in concept bottleneck models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 19 (March 2024), 21161–21168. DOI: https://doi.org/10.1609/aaai.v38i19.30109

[41] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters* 150 (Oct. 2021), 228–234. https://doi.org/10.1016/j.patrec.2021.06.030

[42] Di Jin, Elena Sergeeva, Wei-Hung Weng, Geeticka Chauhan, and Peter Szolovits. 2022. Explainable deep learning in healthcare: A methodological survey from an attribution view. *WIREs Mechanisms of Disease* 14, 3 (2022), e1548.

[43] Dmitry Kazhdan, Botty Dimanov, Mateja Jamnik, Pietro Liò, and Adrian Weller. 2020. Now you see me (CME): Concept-based model extraction. arXiv:2010.13233. Retrieved from https://arxiv.org/abs/2010.13233

[44] Sangwon Kim, Jaeyeal Nam, and Byoung Chul Ko. 2022. ViT-NeT: Interpretable vision transformers with neural tree decoder. In *Proceedings of the 39th International Conference on Machine Learning* . Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.), Proceedings of Machine Learning Research, Vol. 162, PMLR, Baltimore, Maryland, USA, 11162–11172. Retrieved from https://proceedings.mlr.press/v162/kim22g.html

[45] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[46] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20) (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, Virtual Event, 5338–5348.

[47] Boris Kovalerchuk, Muhammad Aurangzeb Ahmad, and Ankur Teredesai. 2021. Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. *Studies in Computational Intelligence* 937 (Mar 2021), 217–267. https//doi.org/10.1007/978-3-030-64949-4_8

[48] Cliff Kuang. 2017. *Can A.I. Be Taught to Explain Itself?* The New York Times. Retrieved February 15, 2024 from https://www.nytimes.com/2017/11/21/magazine/can-ai-be-taught-to-explain-itself.html

[49] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2013. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2013), 453–465.

[50] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.

[51] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. 2015. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics* 9, 3 (2015), 1350–1371. DOI: https://doi.org/10.1214/15-AOAS848

[52] Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. 2021. SCOUTER: Slot attention-based classifier for explainable image recognition. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*. IEEE, 1026–1035. DOI: https://doi.org/10.1109/ICCV48922.2021.00108

[53] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 432, 8 pages.

[54] Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. 2022. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems* 64, 12 (2022), 3197–3234.

[55] Xiao-Hui Li, Caleb Chen Cao, Yuhan Shi, Wei Bai, Han Gao, Luyu Qiu, Cong Wang, Yuanyuan Gao, Shenjia Zhang, Xun Xue, et al. 2020. A survey of data-driven and knowledge-aware explainable AI. *IEEE Transactions on Knowledge and Data Engineering* 34, 1 (2020), 29–49.

[56] Yu Liang, Siguang Li, Chungang Yan, Maozhen Li, and Changjun Jiang. 2021. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing* 419 (Jan. 2021), 168–182. https://doi.org/10.1016/j.neucom.2020.08.011

[57] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2020. Explainable AI: A review of machine learning interpretability methods. *Entropy* 23, 1 (2020), 18.

[58] Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplement* 27 (March 1990), 247–266. https://doi.org/10.1017/S1358246100005130

[59] Zachary C. Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018), 31–57.

[60] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. 2020. Object-centric learning with slot attention. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* . 11525–11538.

[61] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'12)*. ACM, New York, NY, USA, 150–158. DOI: https://doi.org/10.1145/2339530.2339556

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

[63] Jian-Xun Mi, An-Di Li, and Li-Fang Zhou. 2020. Review study of interpretation methods for future interpretable machine learning. *IEEE Access* 8 (Oct. 2020), 191969–191985.

[64] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (Feb. 2019), 1–38. https://doi.org/10.1016/j.artint.2018.07.007

[65] Christoph Molnar. 2019. *Interpretable Machine Learning*. Lulu.com, Research Triangle, NC, USA.

[66] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. 2021. This looks like that, because ... explaining prototypes for interpretable image recognition. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Michael Kamp, Irena Koprinska, and Adrien Bibal, et al. (Eds.). Springer International Publishing, Cham, 441–456. https://doi.org/10.1007/978-3-030-93736-2_34

[67] Sajid Nazir, Diane M. Dickson, and Muhammad Usman Akram. 2023. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine* 156, C (Apr 2023), 29 pages. DOI: https://doi.org/10.1016/j.compbiomed.2023.106668

[68] Shreyasi Pathak, Jörg Schlötterer, Jeroen Veltman, Jeroen Geerdink, Maurice van Keulen, and Christin Seifert. 2024. Prototype-based interpretable breast cancer prediction models: Analysis and challenges. In *Explainable Artificial Intelligence*. Luca Longo, Sebastian Lapuschkin, and Christin Seifert (Eds.), Springer Nature Switzerland, Cham, 21–42. DOI: https://doi.org/10.1007/978-3-031-63787-2_2

[69] Yitao Peng, Lianghua He, Die Hu, Yihang Liu, Longzhen Yang, and Shaohua Shang. 2024. Decoupling deep learning for enhanced image recognition interpretability. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 10 (Oct. 2024), 309:1–309:24. DOI: https://doi.org/10.1145/3674837

[70] Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: Randomized input sampling for explanation of black-box models. arXiv:1806.07421. Retrieved from https://arxiv.org/abs/1806.07421

[71] R. Quinlan. 1986. Induction of decision trees. *Machine Learning* 1, 1 (March 1986), 81–106. https://doi.org/10.1007/BF00116251

[72] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al.. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Virtual Event, 8748–8763. Retrieved from https://proceedings.mlr.press/v139/radford21a.html

[73] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. IEEE, 658–666. DOI: https://doi.org/10.1109/CVPR.2019.00075

[74] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, USA, 1135–1144. DOI: https://doi.org/10.1145/2939672.2939778

[75] Mattia Rigotti, Christoph Miksovic, Ioana Giurgiu, Thomas Gschwind, and Paolo Scotton. 2022. Attention-based interpretability with concept transformers. In *Proceedings of the International Conference on Learning Representations (ICLR'22)*. OpenReview.net, Virtual Event, 16 pages.

[76] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. IEEE, 10674–10685. DOI: https://doi.org/10.1109/CVPR52688.2022.01042

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

[78] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys* 16, none (Jan. 2022). https://doi.org/10.1214/21-SS133

[79] Waddah Saeed and Christian Omlin. 2023. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems* 263 (March 2023), 110273. https://doi.org/10.1016/j.knosys.2023.110273

[80] Anirban Sarkar, Deepak Vijaykeerthy, Anindya Sarkar, and Vineeth N. Balasubramanian. 2022. A framework for learning ante-hoc explainable models via concepts. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*. IEEE, 10276–10285. DOI: https://doi.org/10.1109/CVPR52688.2022.01004

[81] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision* 128, 2 (Feb 2020), 336–359. DOI: https://doi.org/10.1007/s11263-019-01228-7

[82] Sangwoo Seo, Sungwon Kim, and Chanyoung Park. 2023. Interpretable prototype-based graph information bottleneck. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* . 76737–76748. Retrieved from https://proceedings.neurips.cc/paper_files/paper/2023/hash/f224f056694bcfe465c5d84579785761-Abstract-Conference.html

[83] Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2931–2951. DOI: https://doi.org/10.18653/v1/P19-1282

[84] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning (ICML'17, Vol. 70)*. JMLR.org, 3145–3153. Retrieved from https://proceedings.mlr.press/v70/shrikumar17a.html

[85] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[86] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations ICLR*, Banff, AB, Canada, 8 pages.

[87] Gurmail Singh and Kin-Choong Yow. 2021. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access* 9 (March 2021), 41482–41493.

[88] Mohana Singh, B. S. Vivek, Jayavardhana Gubbi, and Arpan Pal. 2024. Prototype-based interpretable model for glaucoma detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, Seattle, WA, USA, 5056–5065.

[89] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. IEEE, 2818–2826. DOI: https://doi.org/10.1109/CVPR.2016.308

[90] Andong Tan, Fengtao Zhou, and Hao Chen. 2025. Explain via any concept: Concept bottleneck model with open vocabulary concepts. In *Computer Vision – ECCV 2024*. Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.), Springer Nature Switzerland, Cham, 123–138. DOI: https://doi.org/10.1007/978-3-031-73016-0_8

[91] Zhen Tan, Lu Cheng, Song Wang, Bo Yuan, Jundong Li, and Huan Liu. 2024. Interpreting pretrained language models via concept bottlenecks. In *Advances in Knowledge Discovery and Data Mining*. De-Nian Yang, Xing Xie, Vincent S. Tseng, Jian Pei, Jen-Wei Huang, and Jerry Chun-Wei Lin (Eds.), Springer Nature, Singapore, 56–74. DOI: https://doi.org/10.1007/978-981-97-2259-4_5

[92] Qiaoying Teng, Zhe Liu, Yuqing Song, Kai Han, and Yang Lu. 2022. A survey on the interpretability of deep learning in medical diagnosis. *Multimedia Systems* 28, 6 (2022), 2335–2355.

[93] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58, 1 (1996), 267–288.

[94] Erico Tjoa and Cuntai Guan. 2020. A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems* 32, 11 (2020), 4793–4813.

[95] Naeem Ullah, Javed Ali Khan, Ivanoe De Falco, and Giovanna Sannino. 2024. Explainable artificial intelligence: Importance, use domains, stages, output shapes, and challenges. *ACM Computing Surveys* 57, 4 (2024), 1–36.

[96] Bas H. M. Van Der Velden, Hugo J. Kuijf, Kenneth G. A. Gilhuijs, and Max A. Viergever. 2022. Explainable artificial intelligence (XAI) in deep learning-based medical image analysis. *Medical Image Analysis* 79 (July 2022), 102470. https://doi.org/10.1016/j.media.2022.102470

[97] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* .

[98] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report CNS-TR-2011-001. California Institute of Technology.

[99] Qiyang Wan, Ruiping Wang, and Xilin Chen. 2024. Interpretable object recognition by semantic prototype analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, Hawaii, 800–809.

[100] Jingqi Wang, Peng Jiajie, Zhiming Liu, and Hengjun Zhao. 2023. HQProtoPNet: An evidence-based model for interpretable image recognition. In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN'23)*. IEEE, 1–8. DOI: https://doi.org/10.1109/IJCNN54540.2023.10191863

[101] Graham Webster, Rogier Creemers, Elsa Kania, and Paul Triolo. 2017. *Full Translation: China's 'New Generation Artificial Intelligence Development Plan'*. DigiChina. Retrieved February 15, 2024 from https://digichina.stanford.edu/work/full-translation-chinas-new-generation-artificial-intelligence-development-plan-2017/

[102] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. 2019. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *Proceedings of the 8th CCF International Conference on Natural Language Processing and Chinese Computing, NLPCC 2019, Proceedings, Part II* (Dunhuang, China). Springer-Verlag, Berlin, 563–574. DOI: https://doi.org/10.1007/978-3-030-32236-6_51

[103] Xinyue Xu, Yi Qin, Lu Mi, Hao Wang, and Xiaomeng Li. 2024. Energy-Based concept bottleneck models: unifying prediction, concept intervention, and probabilistic interpretations. In *International Conference on Learning Representations ICLR*. Vienna, Austria, 23 pages.

[104] Yang Xu and Zuqiang Meng. 2024. Interpretable vision transformer based on prototype parts for COVID-19 detection. *IET Image Processing* 18, 7 (May 2024), 1927–1937. DOI: https://doi.org/10.1049/ipr2.13074

[105] Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. 2023. Global concept-based interpretability for graph neural networks via neuron analysis. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence and 35th Conference on Innovative Applications of Artificial Intelligence and 13th Symposium on Educational Advances in Artificial Intelligence (AAAI'23/IAAI'23/EAAI'23, Vol. 37)*. AAAI Press, Washington, DC, USA, 10675–10683. DOI: https://doi.org/10.1609/aaai.v37i9.26267

[106] Éloi Zablocki, Hédi Ben-Younes, Patrick Pérez, and Matthieu Cord. 2022. Explainability of deep vision-based autonomous driving systems: Review and challenges. *International Journal of Computer Vision* 130, 10 (2022), 2425–2452.

[107] Quan-shi Zhang and Song-Chun Zhu. 2018. Visual interpretability for deep learning: A survey. *Frontiers of Information Technology & Electronic Engineering* 19, 1 (2018), 27–39.

[108] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. 2021. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence* 5, 5 (2021), 726–742.

[109] Zhibo Zhang, Hussam Al Hamadi, Ernesto Damiani, Chan Yeob Yeun, and Fatma Taher. 2022. Explainable artificial intelligence applications in cyber security: State-of-the-art in research. *IEEE Access* 10 (Sept. 2022), 93104–93139. https://doi.org/10.1109/ACCESS.2022.3204051

[110] Zaixi Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Cheekong Lee. 2022. ProtGNN: Towards self-explaining graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 8 (June 2022), 9127–9135. DOI: https://doi.org/10.1609/aaai.v36i8.20898

[111] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*. IEEE, 2921–2929. DOI: https://doi.org/10.1109/CVPR.2016.319

[112] Julia El Zini and Mariette Awad. 2022. On the explainability of natural language processing deep models. *ACM Computing Surveys* 55, 5 (2022), 1–31.

Received 6 June 2024; revised 27 January 2025; accepted 1 April 2025