Natural Example-Based Explainability: a Survey

Antonin Poché, Lucas Hervier, Mohamed-Chafik Bakkay

Natural Example-Based Explainability: a Survey

Antonin Poché*[1,2][0009−0000−0930−3313], Lucas Hervier*[1,2][0000−0002−6642−467X], and Mohamed-Chafik Bakkay[1,2][0009−0005−4362−6569]

[1] IRT Saint Exupéry, Toulouse, France name.surname@irt-saintexupery.com
[2] IRT SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France
name.surname@irt-systemx.fr

Abstract. Explainable Artificial Intelligence (XAI) has become increasingly significant for improving the interpretability and trustworthiness of machine learning models. While saliency maps have stolen the show for the last few years in the XAI field, their ability to reflect models' internal processes has been questioned. Although less in the spotlight, example-based XAI methods have continued to improve. It encompasses methods that use examples as explanations for a machine learning model's predictions. This aligns with the psychological mechanisms of human reasoning and makes example-based explanations natural and intuitive for users to understand. Indeed, humans learn and reason by forming mental representations of concepts based on examples.
This paper provides an overview of the state-of-the-art in natural example-based XAI, describing the pros and cons of each approach. A "natural" example simply means that it is directly drawn from the training data without involving any generative process. The exclusion of methods that require generating examples is justified by the need for plausibility which is in some regards required to gain a user's trust. Consequently, this paper will explore the following family of methods: similar examples, counterfactual and semi-factual, influential instances, prototypes, and concepts. In particular, it will compare their semantic definition, their cognitive impact, and added values. We hope it will encourage and facilitate future work on natural example-based XAI.

Keywords: Explainability · XAI · Survey · Example-based · Case-based · Counterfactuals · Semi-factuals · Influence Functions · Prototypes · Concepts

1 Introduction

With the ever-growing complexity of machine learning models and their large diffusion, understanding models' decisions and behavior became a necessity. Therefore, explainable artificial intelligence (XAI), the field that aims to understand and clarify models, flourished with a huge diversity of methods. To differentiate between methods several taxonomies were proposed, and common components emerged [2,4,50]: i) Local vs global: Local methods explain a specific decision of

* These authors contributed equally to this work

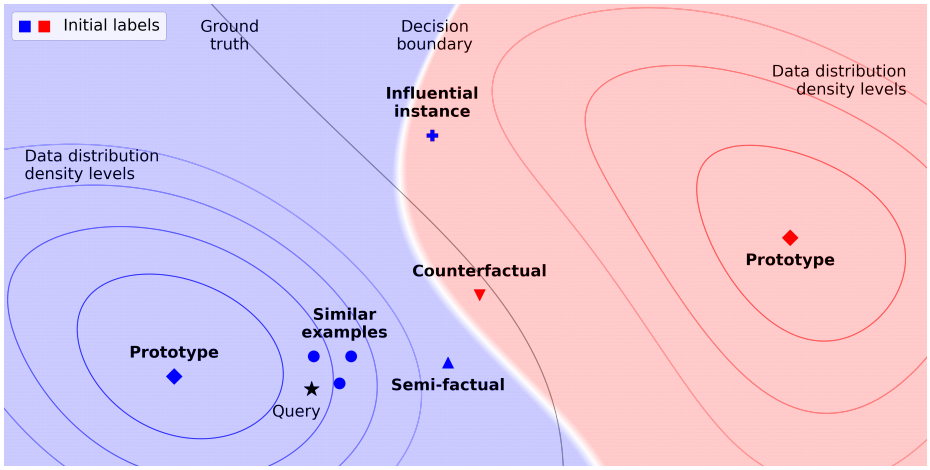

Fig. 1. Natural example-based explanation formats w.r.t the studied sample (or query) and the decision boundary. We can see similar examples are the closest elements to the query, while counterfactuals and semi-factual are on either side of the point of the decision boundary the closest to the query. Prototypes are representative of each class in a dense zone of the dataset and the influential instance bends the decision boundary.

the model (in this case, the model's input is called the studied sample or query). Global methods give insight into the general model behavior. Methods can also explain the dataset but it will not be covered in this survey. ii) *Post-hoc* vs intrinsic vs explainable by-design: *Post-hoc* methods are applied on an already trained model. By-design methods produce models explainable by construction. Intrinsic methods need to be taken into account during the training of the model but do not affect the final state of the model and explain either the training process or the trained model. iii) Black-box vs white-box: White-box methods need access to the model's weights and/or gradients. iv) The format of the explanation: The multiplicity of methods translates through the large range of forms of explanations such as attribution methods [33,102], concepts [35,66], surrogate models [67,94], rule-based explanations [112], natural language explanations [20], dependencies [40,49], and finally example-based explanations [57,111].

Nonetheless, no matter the taxonomy of a method, its explanations are aimed at humans, hence, they should exploit the vast literature in philosophy, psychology, and cognitive science on how humans generate, understand, and react to explanations [78]. The psychology literature argued that, in everyday life, humans use examples as references to understand, explain something, or demonstrate their arguments [18,38,78,98]. Afterward, through user studies in the XAI field [35,51,61], researchers validated that example-based explainability provides better explanations over several other formats. Example-based explainability corresponds to a family of methods where explanations are represented by or communicated through samples, or part of samples like crops. This means the explanation's format is a data point (an example).

Examples can either be training samples (natural examples) or generated elements. To generate high-dimensional data points, methods are essentially based on deep neural networks [6,62]. However, for most high dimensional data, such methods fail to ensure that generated examples are plausible and belong to the manifold (subspace of the input space where samples follow the data distribution), and examples need to be realistic for humans to interpret them [19]. Therefore, natural examples have two advantages, they do not use a model to explain another model which eases their acceptance, and natural examples are plausible by definition. Hence, this survey will cover natural (non-generative) example-based explainability methods that explain AI models.

Explanations in example-based explainability are all data points but there exist different semantic meanings to a given example. Depending on the relation between the example, the query, and the model, the information provided by the example will differ. The semantic definition of an example and the kind of insight it provides divide the example-based format into sub-groups, which are presented in Fig. 1. This overview is organized around those sub-groups (also called formats), this work will unfold as follows:

The first format is **similar examples** (or factuals) (Section 2), for the model, they are the closest elements to the query. Factuals give confidence in the prediction or explain misclassification, but they are limited to the close range of the considered sample. To provide insight into the model behavior on a larger zone around the query, **counterfactuals** and **semi-factuals** (Sections 3.1 and 3.2) are more adapted. They are respectively the closest and the farthest samples on which the model makes a different and similar prediction. They are mainly used in classification, give insight into the decision boundary, and are complementary if paired. While they give an idea of the limit, they do not provide insights on how one could bend the decision boundaries of the model by altering the training data. This is addressed through **influential instances** (Section 4), the training samples with the highest impact on the model's state. In addition, contrary to previously listed example-based formats, influential instances are not limited to local explanations. Indeed, one can extract the most influential instances for the model in general. Another global explanation format is **Prototypes** (Section 5), which are a set of samples representative of either the dataset or a class. Most of the time they are selected without relying on the model and give an overview of the dataset, but some models are designed through prototypes, thus explainable by design. Concepts (Section 6), a closely-related format, is also investigated. A concept is the abstraction of the common elements between samples – e.g. for trees, the concepts could be trunk, branch, and leaf. To communicate such concepts, if they are not labeled, the easiest way is through examples of such concepts (often part of samples such as patches). Finally, **feature visualization** [89] are generated images that maximize the model prediction for a given class. It shows what the model associate with a given class, however, it is generative and will not be further discussed in this review.

Thus we could summarize the contributions of this paper as follows: i) To the best of our knowledge, we are the first to compile natural example-based

explainability literature in a survey. Previous works either covered the whole XAI literature with a superficial analysis of example-based XAI or focused on a given sub-format of example-based XAI. ii) For each format we provide simple definitions, semantic meanings, and examples. When possible, we additionally ground formats into social sciences and depict their cognitive added values. iii) We explore, classify, and describe available methods in each natural example-based XAI format. We highlight common points and divergences for the reader to understand each method easily, with a focus on key methods. (see Tab. 1)

1.1 Notations

Throughout the paper, methods will explain a machine learning model $h : \mathcal{X} \to \mathcal{Y}$, with $\mathcal{X}$ and $\mathcal{Y}$ being respectively the input and output domain. Especially, this model is parameterized by the weights $\theta \in \Theta \subseteq \mathbb{R}^d$. If not specified otherwise, h is trained on a training dataset $\mathcal{D}_{train} \subset (\mathcal{X} \times \mathcal{Y})$ of size n with the help of a loss function $l : (\mathcal{X}, \mathcal{Y}, \Theta) \to \mathbb{R}$. We denote a sample by the tuple $z = (x, y)| \quad x \in \mathcal{X}, y \in \mathcal{Y}$. When an index subscript as i or j is added, *e.g.* z_i, it is assumed that z_i belongs to the training dataset. If the subscript "test" is added, z_{test}, the sample does not belong to the training data. When there is no subscript, the sample can either be or not in the training data. Finally, the empirical risk function is denoted as $\mathcal{L}(\theta) := \frac{1}{n} \sum_{(x,y) \in \mathcal{D}_{train}} l(x, y, \theta) = \frac{1}{n} \sum_{z_j \in \mathcal{D}_{train}} l(z_j, \theta)$, the parameters that minimized this empirical risk as $\theta^* := \arg\min_\theta \mathcal{L}(\theta)$ and an estimator of θ^* is denoted $\hat{\theta}$.

2 Similar examples

In the XAI literature, similar examples, also referred to as factual examples (see Fig. 2), are often used as a way to provide intuitive and interpretable explanations. The core idea is to retrieve the most similar, or the closest, elements in the training set to a sample under investigation z_{test} and to use them as a way to explain a model's output. Specifically, Case-Based Reasoning (CBR) is of particular interest as it mimics the way humans draw upon past experiences to navigate novel situations [38,98]. For example, when learning to play a new video game, individuals do not typically begin from a complete novice level. Instead, they rely on their pre-existing knowledge and skills in manipulating game controllers and draw upon past experiences with similar video games to adapt and apply strategies that have been successful in the past. As described by Aamodt and Plaza [1], a typical CBR cycle can be delineated by four fundamental procedures: i) RETRIEVE: Searching for the most analogous case or cases, ii) REUSE: Employing the information and expertise extracted from that case to address the problem, iii) REVISE: Modifying the proposed solution as necessary, iv) RETAIN: Preserving the pertinent aspects of this encounter that could be beneficial for future problem-solving endeavors. The CBR approach has gained popularity in fields that require transparent systems to justify their outcomes, such as medicine [15], due to its psychological plausibility. In addition

to being intuitive, the cases retrieved by a CBR system for a given prediction are natural explanations for this output.

While CBR systems are a must-know in the XAI literature, we will not review them as they have already been well analyzed, reviewed, motivated, and described many times [27,29,100]. Instead, the focus here is on case-based explanations (CBE) [100]. CBE are methods that use CBR to explain other systems, also referred to as twin systems [57,60]. Indeed, the CBR system must be coupled with the system you want to explain. In particular, explanations of the system under inspection are generally the outcomes of the RETRIEVE functionality of the twinned CBR system, which oftentimes rely on k-nearest neighbor (k-NN) retrieval [25]. The idea behind k-NN is to retrieve the k most similar training samples (cases) to a test sample z_{test}. In fact, presenting similar examples to an end-user as an explanation for a model's outcomes has been shown through user studies to be generally more convincing than other approaches [53,112].

2.1 Defining similarity

Defining similarity is not trivial. Indeed, there are many ways of defining similarity measures, and different approaches are appropriate for different representations of a training sample [29]. Generally, CBR systems assume that similar input features are likely to produce similar outcomes. Thus, using a distance metric defined on those input features engenders a similarity measure: the closer the more similar they are. One of the simplest is the unweighted Euclidean distance:

$$dist(z, z') = ||x - x'||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{1}$$

However, **where** – *i.e.* in which space – the distance is computed does have major implications. As pointed out by Hanawa *et al.* [46], the input space does not seem to bring pieces of information on the internal working of the model under inspection but provides more of a data-centric analysis. Thus, recent methods rely instead on either computing the distance in a latent space or weighting features for the k-NN algorithm [32].

Computing distance in a latent space is one possibility to include the model in the similarity measure which is of utmost importance if we want to explain it, as pointed out by Caruana *et al.* [21]. Consequently, Caruana *et al.* [21] suggested applying the Euclidean distance on the last hidden units h_{-1} of a trained Deep Neural Network (DNN) as a similarity which considers the model's predictions:

$$dist_{DNN}(z, z') = ||h_{-1}(x) - h_{-1}(x')||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{2}$$

Similarly, for Deep Convolutional Neural Networks, Papernot and McDaniel [90], and Sani *et al.* [96] suggested conducting the k-NN search in the latent representation of the network and using the cosine similarity distance.

Weighting features is another popular paradigm in CBE. For instance, Shin *et al.* [104] proposed various **global weighting** schemes – *i.e.* methods in which the weights assigned to each input's feature remain constant across all samples as in Eq. (3) – where the weights are computed using the trained network to reveal the input features that were the most relevant for the network's prediction.

$$dist_{features\_weights}(z, z') = ||w(\hat{\theta})^T(x - x')||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{3}$$

Alternatively, Park *et al.* [91] examined **local weighting** by considering varying feature weights across the instance space. However, their approach is not *post-hoc* for DNN. Besides, Nugent *et al.* [87] also focused on local weighting and proposed a method that can be applied to any black-box model. However, their method involves generating multiple synthetic datasets around a specific sample, which may not be suitable for explaining a large number of samples or high-dimensional inputs. In the same line of work, Kenny and Keane [60,61] proposed COLE, by suggesting the direct k-NN search in the attribution space – *i.e* computing saliency maps [7,105,108] for all instances and performing a k-NN search in the resulting dataset of attributions. By denoting $c(\hat{\theta}, z)$ the attribution map of the sample z for the model parameterized by $\hat{\theta}$ that gives:

$$dist_{COLE}(z, z') = ||c(\hat{\theta}, z) - c(\hat{\theta}, z')||_2 \tag{4}$$

They used three saliency map techniques ([7,105,108]) but nothing prevents one to leverage any other saliency map techniques. However, we should also point out that Fel *et al.* [34] questioned attribution methods' ability to truly capture the internal process of DNN. Additionally in [61], Kenny and Keane proposed to use the Hadamard product of the gradient times the input features as a contribution score in the case of DNN with non-linear outputs.

2.2 Limitations

The current limitations of similarity-based XAI are still significant. Indeed, even though one defines a relevant distance or similarity measure between samples one still has to perform the search in the training dataset to retrieve the closest samples for a given z_{test}. Naively, this would at least require computing the distance between z_{test} with every training data point, which prohibits its computation for large datasets. Fortunately, there are efficient techniques available for searching, as briefly discussed in the paper by Bhatia *et al.* [14]. However, if the training data is sparse in the space in which the distance is computed the retrieved cases might be far from z_{test}, thus questioning their relevance.

Furthermore, **where** the distance is computed does have major implications as mentioned by Hanawa *et al.* [46]. Consequently, authors have suggested different feature spaces or weighting schemes to investigate, but their relevance to reflect the inner workings of a model is as questionable as it is for attributions methods [34]. In addition, it is still unclear in the literature if one approach

prevails over others. Moreover, when a human is faced with examples, he may not be able to understand why they were considered similar. As an example, if two elements are red and round, the human may think the important thing is the red color while the model focuses on the round shape [84].

Finally, the consideration of the position of the retrieved similar examples w.r.t. the decision boundaries of the model, in terms of whether their prediction matches that of z_{test}, is not always accounted for. It is a major issue as providing similar examples to an end-user should comfort it with the model's decision but that becomes confusing if you showcase a factual example for which the model's prediction is different. Thus, taking into account the decision boundaries of a model seems crucial for the explanations' relevance. Such considerations are motivating the field of contrastive explanations, as discussed in section 3.

3 Contrastive explanations

Contrastive explanations are a class of explanation that provides the consequences of another plausible reality, the repercussion of changes in the model's input [18,111]. More simply, they are explanations where we modify the input and observe the reaction of the model's prediction, the modified input is returned as the explanation and its meaning depends on the model's prediction of it. Those methods are mainly *post-hoc* methods applied to classification models. This includes i) counterfactuals (CF): *an imagined alternative to reality about the past, sometimes expressed as "if only ... " or "what if ..."* [18], ii) semi-factuals (SF): *an imagined alternative that results in the same outcome as reality, sometimes expressed as "even if ... "* [18], and iii) adversarial examples (perturbations or attacks) (AP): *inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence* [41]. Examples of those three formats are provided in Fig. 2 from Kenny and Keane [62].

AP and CF are both perturbations with an expected change in the prediction, they only differ in the goal as CF attempt to provide an explanation of the model's decision while AP are mainly used to evaluate robustness. In fact, AP can be considered CF [113], and for robust models, AP methods can generate interpretable CF [103]. Nonetheless, AP are hardly perceptible perturbations designed to fool the model [109], therefore, they are generative and those methods will not be further detailed in this work. Then, we can generalize SF and CF, with a given distance $dist$, and the examples conditioned space $\mathcal{X}_{cond(f,x)} \subset \mathcal{X}$:

$$CF(x_{test}) := \underset{x \in \mathcal{X}_{cond(f,x_{test})} | h(x) \neq h(x_{test})}{\arg\min} dist(x_{test}, x) \tag{5}$$

$$SF(x_{test}) := \underset{x \in \mathcal{X}_{cond(f,x_{test})} | h(x) = h(x_{test})}{\arg\max} dist(x_{test}, x) \tag{6}$$

For natural CF and SF, the input space is conditioned to the training set, $\mathcal{X}_{cond(f,x_{test})} = X_{train}$. While for AP, there is no condition on the input space,

in Eq. (5), $\mathcal{X}_{cond(f,x_{test})} = \mathcal{X}$. The distance and the condition of the input space are the key differences between CF and SF methods.

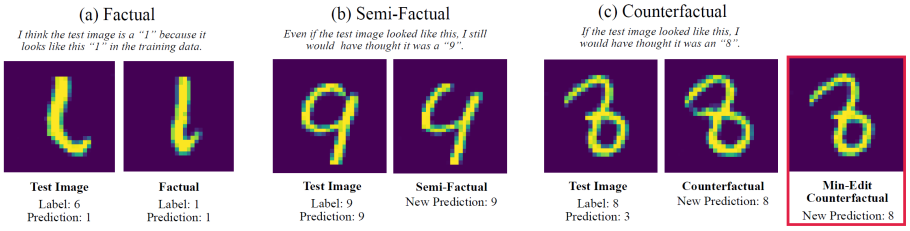

Fig. 2. Illustration of factuals, SF, and CF from Kenny and Keane [62]. The factual makes us understand the misclassification, while SF and CF show us how far or close the decision boundary is. Min-edit represents the AP, as differences are not visible.

3.1 Counterfactuals

The social science grounding of counterfactuals is deep, either in philosophy, or psychology. Indeed, the search for CF's semantic definition goes back a long time [13,44,72], and historically revolves around the notion of cause and effect, sometimes called facts and foils [75,78]. Then, Halpern and Pearl [44] proved that the cause of an event is an answer to the question "Why?" and thus, provides a powerful explanation. Moreover, the philosophical literature argued that CF allow us to communicate and understand the causal relation between facts and foils [72,78]. Psychology also possesses a rich literature regarding CF [18,95], which has continued to evolve in recent years [19,59,79] thanks to the arrival of CF in XAI through Wachter *et al.* [113]. Humans' natural use of counterfactuals in many situations was highlighted by Byrne [18]: *From amusing fantasy to logical support, they explain the past, prepare the future, modulate emotional experience, and support moral judgments*. Furthermore, when people encounter CF they have both the counterfactual and the factual in mind [19,110]. The insights from philosophy and psychology [19,79] have shown the pertinence and potential of CF as well as SF for XAI. To match such promises, CF in XAI need to verify the definitions and properties of CF typically employed by humans.

Expected properties for natural CF can be extrapolated from conclusions and discovered properties in XAI for generated CF even though the literature on natural CF is slim. Such desirable properties for CF, derived from social sciences, could be summarized as follows: i) **plausibility** [58,59,111]: CF should be as realistic as possible; ii) **validity** [83]: if the model's prediction on CF differ from the prediction on the query (see the definition (5)); iii) **sparsity** [58,83,111]: the number of features that were changed between CF and the query should be as

little as possible; iv) **diversity** [54,83]: if several CF are proposed, they should be different from each other; v) **actionability** [58,111]: the method should allow the user to select features, to modify and specify immutable ones; vi) **proximity** [54,58,59,83]: CF should be as close as possible to the query.

Counterfactuals methods: Keane *et al.* [59] argued that nearest unlike neighbors (NUN) [28] is the ancestor of counterfactuals in XAI. NUN are derivative of nearest neighbors [25], which is looking for the nearest element that belongs to a different class, it matches perfectly with the definition of natural counterfactuals. NUN were first used in XAI by Doyle *et al.* [30,88] but not as an explanation, only to find SF. The only method to the best of our knowledge that uses NUN as explanations is KLEOR from Cummins and Bridge [26], which was also called "the nearest miss" and was provided as a complement to SF explanation. Indeed, following the definition, pairs of CF and SF should give a good intuition of the decision boundary. Nonetheless, they highlighted that the decision boundary might be much more complex than what the SF and CF pairs can reveal. Indeed, a line between SF and CF may intersect the decision boundary several times, which can lead to explanations that are not always faithful. Furthermore, Keane *et al.* [59] argued that "good natural counterfactuals are hard to find" as the dataset's low density prevents sparse and proximal natural CF.

Counterfactuals as known in XAI appeared with Wachter *et al.* [113]. While there are numerous methods, as shown through the number of surveys in this field[54,83,111], those are all generative methods. We can distinguish two periods among those papers: a first one with a focus on small and interpretable tabular datasets as described by Verma *et al.* survey [111], and a second on more complex data types such as images [6,62]. While in the first CF period, generating plausible instances was not an issue, it appeared to be a huge drawback toward the generalization of CF to more complex data types [6,62]. Even the most recent methods based on diffusion models [6] failed to consistently generate plausible images. We are surprised that there is so little work that explores natural CF as explanations with their inherent plausibility. Furthermore, in the literature, natural examples were used to ensure plausibility in generated CF [59,111]. Moreover, adversarial perturbations proved that for non-robust DNN, a generated example close to a natural instance is not necessarily plausible. That is to say, we cannot prove that generated instances belong to the manifold without a proper definition of the manifold. To conclude, for high dimensional data, the reader is faced with the choice of simple and plausible natural CF or proximal and sparse generated CF through a model explaining another model.

3.2 Semi-factuals

SF literature is most of the time included in the CF literature be it in philosophy [42], psychology [18], or XAI [26,62]. In fact, SF, "even if ..." are semantically close to CF, "what if ..." [5,13,42], (see Eqs. (5) and (6)). However, psychology has demonstrated that human reactions differ between CF and SF. While CF

strengthen the causal link between two elements, SF reduce it [19], CF increase fault and blame in a moral judgment while SF diminish it.

Expected properties for CF and SF were inspired by social science, hence, because of their close semantic definition, many properties are common between both: SF should also respect their definition in Eq. (6) (**validity**), then to make the comparison possible and relevant they should aim towards **plausibility** [5], **sparsity** [5], **diversity**, and **actionability**. Nonetheless, the psychological impact of CF and SF differ, hence there are also SF properties that contrast with CF properties. The difference between equations (5) and (6) – *i.e.* $\arg\min$ vs $\arg\max$ – suggests that to replace CF's proximity, SF should be the farthest from the studied sample, while not crossing the decision boundary [26]. As such, we propose the **decision boundary closeness** as a necessary property, and a metric to evaluate it could be the distance between SF and SF's NUN. Finally, SF should not go in any direction from the studied sample but aim toward the closest decision boundary. Therefore, it should be aligned with NUN [26,30,88], this property was not named, we suggest calling it **counterfactual alignment**.

Semi-factuals methods were first reviewed in XAI by a recent survey from Aryal and Keane [5]. They divided SF methods and history into four parts. The first three categories consist of one known method that will illustrate them:

- **SF based on feature-utility**, Doyle *et al.* [30] discovered that similar examples may not be the best explanations and suggested giving examples farther from the studied sample. To find the best explanation case, *dist* in Eq. (6) is a utility evaluation based on features difference.
- **NUN-related SF**, Cummins and Bridge [26] proposed KLEOR where Eq. (6)'s *dist* is based on NUN similarity. Then, they penalize this distance to make sure the SF are between the query and nearest unlike neighbors.
- **SF near local-region boundaries**, Nugent *et al.* [88] approximate the decision boundary of the model in the neighborhood of the studied sample through input perturbations (like LIME [94]). Then SF are given by the points that are the closest to the decision boundary.
- **The modern era: *post*-2020 methods**, inspired by CF methods, many generative methods emerged in recent years [55,62].

In conclusion, semi-factuals are a natural evolution of similar examples. Furthermore, their complementarity with counterfactuals was exposed through the literature, first to find and evaluate SF, and then to provide a range to the decision boundary. Even though contrastive explanations bring insights into a model's behavior by answering a "*what if...* or a "*even if...*" statement, it has no impact on the current model situation and what led to this state or how to change it. Contrastively, influential instances (see Section 4) extract the samples with the most influence on the model's training, hence its current state. Thus, removing such samples from the training set will have a huge impact on the resulting model.

4 Influential Examples

Influential instances could be defined as instances more likely to change a model's outcome if they were not in the training dataset. Furthermore, such measures of influence provide one with information on "in which direction" the model decision would have been affected if that point was removed. Being able to trace back to the most influential training samples for a given test sample z_{test} has been a topic of interest mainly for example-based XAI.

4.1 Influence functions

Influence functions originated from robust statistics in the early 70s. In essence, they evaluate the change of a model's parameters as we up-weight a training sample by an infinitesimal amount: [45] $\hat{\theta}_{\epsilon,z_j} := \arg\min_\theta \mathcal{L}(\theta) + \epsilon l(z_j, \theta)$. One way to estimate the change in a model's parameters of a single training sample would be to perform *Leave-One-Out* (LOO) retraining, that is, to train the model again with the sample of interest being held out of the training dataset. However, repeatedly re-training the model to exactly retrieve the parameters' changes could be computationally prohibitive, especially when the dataset size and/or the number of parameters grows. As removing a sample z_j can be linearly approximated by up-weighting it by $\epsilon = -\frac{1}{n}$, computing influence helps to estimate the change of a model's parameters if a specific training point was removed. Thus, by making the assumption that the empirical risk $\mathcal{L}$ is twice-differentiable and strictly convex w.r.t. the model's parameters θ making the Hessian $H_{\hat{\theta}} := \frac{1}{n} \sum_{z_i \in \mathcal{D}_{train}} \nabla^2_\theta l(z_i, \hat{\theta})$ positive definite, Cook and Weisberg [24] proposed to compute the influence of z_j on the parameters $\hat{\theta}$ as:

$$\mathcal{I}(z_j) := -H^{-1}_{\hat{\theta}} \nabla_\theta l(z_j, \hat{\theta}) \tag{7}$$

Later, Koh and Liang [68] popularized influence functions in the machine learning community as they took advantage of auto-differentiation frameworks to efficiently compute the hessian for DNN and derived Eq. (7) to formulate the influence of up-weighting a training sample z_j on the loss at a test point z_{test}:

$$\text{IF}(z_j, z_{test}) := -\nabla_\theta l(z_{test}, \hat{\theta})^T H^{-1}_{\hat{\theta}} \nabla_\theta l(z_j, \hat{\theta}) \tag{8}$$

This formulation opens its way into example-based XAI as it compares to the study of finding the nearest neighbors of z_{test} in the training dataset – *i.e.* the most similar examples (Section 2) – with two major differences though: i) points with high training loss are given more influence *revealing that outliers can dominate the model parameters* [68], and ii) $H^{-1}_{\hat{\theta}}$ measures what Koh and Liang called: *the resistance of the other training points to the removal of* z_j [68]. However, it should be noted that hessian computation remains a significant challenge, that could be alleviated with common techniques [3,77,99]. By normalizing Eq. (8), Barshan *et al.* [10] further added stability to the formulation.

Oftentimes, we are not only interested in individual instance influence but in the influence of a group of training samples (*e.g.* mini-batch effect, multi-source

data, etc..). Koh *et al.* [69] suggested that using the sum of individual influences as the influence of the group constitutes a good proxy to rank those groups in terms of influence. Basu *et al.* [12] on their side suggested using a second-order approximation to capture possible cross-correlations but they specified it is most likely impracticable for DNN. In a later work, Basu *et al.* [11] concluded that influence function estimates for DNN are fragile as the assumptions on which they rely, being near optimality and convexity, do not hold in general for DNN.

LOO approximation is one of the previously mentioned motivations behind influence estimates as it avoids the prohibitive LOO retraining required for every sample in the training data. Thus, some authors proposed approaches that optimize the number of LOO retraining necessary to get a grasp on a sample's influence such as Feldman and Zhang [36]. Although this significantly reduces the number of retraining compared to naive LOO retraining, it still requires a significant amount of them. Recently, a new approach that relates to influence functions and involves training many models, was introduced with data models [52,97] which we do not review here.

As Basu *et al.* [11] pointed out, there is a discrepancy between LOO approximation and influence function estimates, especially for DNN. However, Bae *et al.* [9] claimed that this discrepancy is due to influence functions approaching what they call the proximal Bregman response function (PBRF), rather than approximating the LOO retraining, which does not interfere with their ability to perform the task they were thought for, especially XAI. Thus, they suggested evaluating the quality of influence estimates by comparing them to the PBRF rather than LOO retraining as it was done until now.

4.2 Other techniques

Influence computation that relies on kernels is another paradigm to find the training examples that are the most responsible for a given set of predictions. For instance, Khanna *et al.* [63] proposed an approach that relies on Fisher's kernels and they related it to the one from Koh and Liang [68] as a generalization of the latter under certain assumptions. Yeh *et al.* [115] also suggested an approach that leverages kernels but this time they relied on the representer theorem [101]. That allows them to focus on explaining only the *pre-activation prediction layer* of a DNN for classification tasks. In addition, their influence scores, called representer values, provide supplementary information, with positive representer values being excitatory and negative values being inhibitory. However, this approach requires introducing an $L2$ regularizer during optimization, which can prevent *post-hoc* analysis if not responsible for training. Additionally, Sui *et al.* [107] argued that this approach provides more of a *class-level* explanation rather than an *instance-level* explanation. To address this issue and the $L2$ regularizer problem, they proposed a method that involves hessian computation on the classification layer, with only the associated computational cost. However, the ability to retrieve relevant samples when investigating only the final

prediction layer was questioned by Feldmann and Zhang [36], who found that memorization does not occur in the last layer.

Tracing the training process has been another research field to compute influence scores. It relies on the possibility to replay the training process by saving some checkpoints of our model parameters, or states, and reloading them in a post-hoc fashion [23,47,93]. In contrast to the previous approaches, they rely neither on being near optimality nor being strongly convex, which is more realistic when we consider the reality of DNN. However, they require handling the training procedure to save the different checkpoints, potentially numerous, hence they are intrinsic methods, which in practice is not always feasible.

4.3 In a nutshell

Influential techniques can provide both global and local explanations to enhance model performance. Global explanations allow for the identification of training samples that significantly shape decision boundaries or outliers (see Fig. 1), aiding in data curation. On the other hand, local explanations offer guidance for altering the model in a desired way (see Fig. 3). Although they have been compared to similar examples and have been shown to be more relevant to the model [46], they are more challenging to interpret and their effectiveness for trustworthiness is unclear. Further research, particularly user studies, is necessary to determine their ability to take advantage of human cognitive processes.

Fig. 3. Figure taken from F. Liu [93]: A tracing process for estimating influence, TracIn, applied on ImageNet. The first column is composed of the test sample, the next three columns display the training examples that have the most positive value of influence score while the last three columns point out the training examples with the most negative values of influence score. (fr-bulldog: french-bulldog)

5 Prototypes

Prototypes are a set of representative data instances from the dataset, while criticisms are data instances that are not well represented by those prototypes [64].

Fig. 4. Figure taken from [64]: Learned prototypes and criticisms from Imagenet dataset (two types of dog breeds)

Fig. 4 shows examples of prototypes and criticisms from Imagenet dataset. Prototypes and criticism can be used to add data-centric interpretability, *post-hoc* interpretability, and to build an interpretable model [82]. The data-centric approaches will be very briefly introduced.

5.1 Prototypes for data-centric interpretability

Clustering algorithms that return actual data points as cluster centers such as k-medoids methods [56,86] could be used to better understand the data distribution. In fact, the cluster centers can be considered as prototypes.

The abundance of large datasets has renewed the interest in the data summarization methods [8,73,74,81,106], also known as set cover methods, which consist of finding a small subset of data points that covers a large dataset. The subset elements can be considered prototypes. Additionally, we found data summarization methods based on the Maximum Mean Discrepancy (MMD), such as MMD-critic [64] and Protodash [43], that learn both prototypes and criticisms.

5.2 Prototypes for *post-hoc* interpretability

Prototypes and criticisms can be used to add *post-hoc* interpretability [82]. This can be achieved by predicting the outputs for the selected prototypes and criticisms with the black-box model, and then using these predictions to find the weaknesses of the model. We can also explain the model by applying clustering and data summarization methods to select prototypes in its latent space.

Filho *et al.* [37] proposed M-PEER (Multiobjective Prototype-based Explanation for Regression) method that finds the prototypes using both the training data and the model output. It optimizes both the error of the explainable model and the fidelity and interpretability metrics. The selected prototypes are then used to provide global and local post-hoc explanations for regression problems.

5.3 Prototype-based models interpretable by design

After data-centric and *post-hoc* methods, there are methods that construct prototype-based models. Those models are interpretable by design because they provide a

set of prototypes that make sense for the model, those methods are mainly designed for classification. Given a training set of points $X_c := \{(x, y) \in \mathcal{D}_{train} | y = c\}$ for each class c, an interpretable classifier learns a set of prototypes $P_c \subseteq X_c$ for each class c. Each P_c is designed to capture the full variability of the class c while avoiding confusion with other classes. The learned prototypes are then used by the model to classify the input. We identified three types of prototype-based classifiers: those that resolve set cover problems, those that use Bayesian models for explanation, and those that are based on neural networks.

Prototype-based classifiers resolving set cover problems select convex sets that cover each class with prototypes to represent it. Various types of convex sets such as boxes, balls, convex hulls, and ellipsoids can be used. Class Cover Catch Digraphs (CCCD) [76] and ProtoSelect [16] used balls where the centers were considered prototypes. Then, the nearest-prototype rule is used to classify the data points. CCCD finds, for each class c, one ball that covers all points of the class c and no points of other classes. Its radius is chosen as large as possible. However, even within large classes, there can still be a lot of interesting within-class variability that should be taken into account when selecting the prototypes. To overcome this limitation, ProtoSelect used a fixed radius across all points, to allow the selection of multiple prototypes for large classes, and they also allow wrongly covered and non-covered points. They simultaneously minimize three elements: i) the number of prototypes; ii) the number of uncovered points; iii) the number of wrongly covered points.

Prototype-based classifiers using Bayesian models for explanation: Kim *et al.* [65] proposed the Bayesian Case Model (BCM) that extends Latent Dirichlet Allocation [17]. In BCM, the idea is to divide the data into s clusters. For each cluster, a prototype is defined as the sample that maximizes the subspace indicators that characterize the cluster. When a sample is given to BCM, this last one yield a vector of probability to belong to each of the s clusters which can be used for classification. Thus, the classifier uses as an input a vector of dimension s, which allows the use of simpler models due to dimensionality reduction. In addition, the prototype of the most likely cluster can then be used as an explanation.

Prototype-based neural network classifiers learn to select prototypes defined in the latent space, which are used for the classification. This lead to a model that is more interpretable than a standard neural network since the reasoning process behind each prediction is "transparent". Learning Vector Quantization (LVQ) [70] is widely used for generating prototypes as weights in a neural network. However, the use of generated prototypes reduces their interpretability. ProtoPNet [22] also stocks prototypes as weights and trains them, but projects them to training samples patches representation during training. Given an input image, its patches are compared to each prototype, the resulting similarity

scores are then multiplied by the learned class connections of each prototype. ProtoPNet has been extended to time series data using ProSeNet [80], or with a more interpretable structure with ProtoTree [85] and HPNet [48]. Instead of using linear bag-of-prototypes, ProtoTree and HPNet used hierarchically organized prototypes to classify images. ProtoTree improves upon ProtoPNet by using a decision tree which provides an easy-to-interpret global explanation and can be used to locally explain a single prediction. Each node in this tree contains a prototype (as defined by ProtoPNet). The similarity scores between image patches and the prototypes are used to determine the routing through the tree. Decision-making is therefore similar to human reasoning [85]. Nauta *et al.* [84] proposed a method called "This Looks Like That, Because" to complete the "This Looks Like That" reasoning used in ProtoPnet. This method allows checking why the model considered two examples as similar. For instance, it is possible that a human thinks that the common point between two examples is their color, while the model uses their shape. The method modifies some characteristics of the input image, such as hue, or shape, to observe how the similarity score changes. This allows us to measure the importance of each of these characteristics.

5.4 In conclusion

Prototypes can either be: i) selected from the training data to explain the data distribution. These prototypes can also be used to find weaknesses of a black-box model by analyzing the output prediction of these prototypes with this model. ii) selected using both the training data and the model output or in the latent space of the model. This allows for *post-hoc* explanations on the model. iii) integrated and selected by the model itself during training and then used for prediction. This allows the model to be interpretable by design.

6 Concept-based XAI

Prototype-based models compare prototypical parts, *e.g.* patches, and the studied sample to make the classification. The idea of parts is not new to the literature, the part-based explanation field, developed for fine-grained classification, is able to detect semantically significant parts of images. The first part-based model required labeled parts for training and can be considered object detection with a semantic link between the detected objects. Afterward, unsupervised methods such as OPAM [92] or Particul [114] emerged, those methods still learned classification in a supervised fashion, but no labels were necessary for part identification. In fact, the explanation provided by this kind of method can be assimilated into concept-based explanations. A concept is an abstraction of common elements between samples, as an example Fig. 5 shows the visualization of six different concepts that the CRAFT method [35] associated with the given image. To understand parts or concepts, the method uses examples and supposes that with a few examples, humans are able to identify the concept.

Fig. 5. Illustration from Fel *et al.* [35]. Natural examples in the colored boxes define a concept. **Purple box**: could define the concept of "**chainsaw**". **Blue box**: could define the concept of "**saw's motor**". **Red box**: could define the concept of "**jeans**".

Like in part-based XAI, the first concept-based method used labeled concepts. Kim et al. [66] introduced concept activation vectors (CAV) to represent concepts using a model latent space representation of images. Then, they design a post-hoc method, TCAV [66] based on CAV to evaluate an image correspondence to a given concept. Even though it seems promising, this method requires prior knowledge of the relevant concepts, along with a labeled dataset of the associated concepts, which is costly and prone to human biases.

Fortunately, recent works have been conducted to automate the concept discovery in the training dataset without humans in the loop. For instance, ACE, proposed by Ghobarni et al. [39], employs a semantic segmentation technique on images belonging to a specific class of interest and use an Inception-V3 neural network to compute activations of an intermediate model layer for these segments. The resulting activations are then clustered to form a set of prototypes, which they refer to as "concepts". However, the presence of background segments in these concepts requires a post-processing clean-up step to remove irrelevant and outlier concepts. Zhang et al. [116] propose an alternative approach to solve the unsupervised concept discovery problem through matrix factorizations [71] in the networks' latent spaces. However, such methods operate at the convolutional kernel level, which may lead to concepts based on shape and/or ignore more abstract concepts.

As an answer, Fel et al. [35] propose CRAFT, which uses Non-Negative Matrix Factorization [71] for concept discovery. In addition to filling in the blank of previous approaches, their method provides an explicit link between the concepts' global and local explanations (Fig. 5). While their approach successfully alleviates the previously mentioned issues, the retrieved concepts are unfortunately not always interpretable. Nonetheless, their user study proved the pertinence of the method.

To conclude, concept-based explanations allow *post-hoc* global and local explanations, by understanding the general concepts associated with a given class

and the concepts used for a decision. We draw attention to methods that do not require expert knowledge to find out relevant concepts as it is prone to confirmation bias. Even though automated concept discovery is making tremendous progress, the interpretation of such concepts and their ability to gain users' trust stay questionable as very few user studies have been conducted on the subject.

7 Conclusions

This paper explored explainability literature about natural example-based explainability and provided a general social science justification for example-based XAI. We described each kind of explanation possible through samples. For each possibility, we reviewed what explanation do they bring, classified and presented the major methods. We summarize all explored described methods in table 1. We saw that all those methods are based on a notion of similarity. As such, for them to explain the model, the similarity between instances should take into account the model. There are two ways of doing it: project the instances in a meaningful space for the model and/or weight instances.

Among the formats, similar examples and influential instances are natural examples by definition. However, contrastive explanations, prototypes, and concept examples can be generated, which brings competition to non-generative methods. We argue that while a "good" natural example may not exist for a given case, at least, natural examples are realistic in the sense that they belong to the data distribution. While generative methods may be able to create such "good" examples, they cannot prove that the generated samples belong to the data manifold. Furthermore, such methods require a model to explain another model, which in turn should be investigated and might involve extensive tuning.

We have illustrated that the different example-based formats bring different kinds of explanations, and each one has its own advantages, Fig. 1 shows their diversity and complementarity. To summarize those advantages non-exhaustively: i) Factuals give confidence in the decisions of the model and are pertinent in AI-assisted decisions. ii) For classification, contrastive explanations give insight into the decision boundary in the locality of the studied sample. iii) Influential instances explain how samples influenced the model training. iv) Prototypes and concepts give information on a global scale, on the whole, model behavior, but may also be used to explain decisions. Nonetheless, like all explanations, we cannot be sure that humans will have a correct understanding of the model or the decision. Furthermore, there is a non-consensus on how to ensure a given method indeed explain the decisions or inner working of the model. Moreover, for example-based explainability, the data is used as an explanation, hence, without profound knowledge of the dataset, humans will not be able to draw conclusions through such explanations. Therefore, the evaluation of example-based methods should always include a user study, which is lacking in this field and in XAI in general. Finally, we hope our work will motivate, facilitate and help researchers to keep on developing the field of XAI and in particular, natural example-based XAI and to address the identified challenges.

SIMILAR EXAMPLES	Year	Global / Local	Post-hoc / Intrinsic	Model or data -type specificity	Distance	Weighting
Caruana et al. [21]	1999	Local	Post-hoc	DNN	Euclidean	None
Shin et al. [104]	2000	Local	Post-hoc	DNN	Euclidean	Global
Park et al. [91]	2004	Local	Intrinsic	DNN	Euclidean	Local
Nugent et al. [87]	2005	Local	Post-hoc	None	Euclidean	Local
Sani et al. [96]	2017	Local	Post-hoc	Deep CNN	Cosine similarity	Local
Papernot and McDaniel [90]	2018	Local	Post-hoc	Deep CNN	Cosine similarity	Local
Cole [60] [61]	2019	Local	Post-hoc	None	Euclidean	Local with attributions

CONTRASTIVE EXPLANATIONS	Year	Global / Local	Post-hoc / Intrinsic	Model or data -type specificity	Semi-factual group of method
Doyle et al. [30,31]	2004	Local	Post-hoc	None	SF based on feature-utility
NUN [26,28,30]	2006	Local	Post-hoc	None	Natural CF
KLEOR [26]	2006	Local	Post-hoc	None	NUN-related SF
Nugent et al. [88]	2009	Local	Post-hoc	None	Local-region boundaries

INFLUENTIAL INSTANCES	Year	Global / Local	Post-hoc / Intrinsic	Model or data -type specificity	Requires model's gradients
Koh and Liang [68]	2017	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ	Yes
Khanna and al. [63]	2018	Local	Post-hoc	Requires an access to the function and gradient-oracles	Yes
Yeh and al. [115]	2018	Local	Intrinsic	Work for classification neural networks with regularization	Yes
Hara and al. [47]	2019	Local	Intrinsic	Models trained with SGD, saving intermediate checkpoints	Yes
Koh and Liang [69]	2019	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ	Yes
Basu and al. [12]	2019	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ	Yes
Barshan and al. [10]	2020	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ	Yes
Feldman and Zhang [36]	2020	Global	Intrinsic	Requires to train numerous models on subsampled datasets	No
Pruthi and al. [93]	2020	Local	Intrinsic	Requires saving intermediate checkpoints	Yes
Sui and al. [107]	2021	Local	Post-hoc	Work for classification neural networks	Yes
Chan and al. [23]	2021	Both	Intrinsic	Requires saving intermediate checkpoints	Yes

PROTOTYPES	Year	Global / Local	Post-hoc / Intrinsic	Model or data-type specificity	Task	Other
CCCD [76]	2003	Both	NA	by-design	Classification	Set cover
ProtoSelect [16]	2011	Both	NA	by-design	Classification	Set cover
Kim et al. [65]	2019	Both	NA	by-design, tabular	Classification	Bayesian-based
ProtoPNet [22]	2019	Both	NA	by-design, FGCV	Classification	Neural network
ProSeNet [80]	2019	Both	NA	by-design, sequences	Classification	Neural network
ProtoTree [85]	2021	Both	NA	by-design, FGCV	Classification	Neural network
M-PEER [37]	2023	Both	Post-hoc	No	Regression	NA

CONCEPTS	Year	Global / Local	Post-hoc / Intrinsic	Model or data -type specificity	Need labeled concepts	Concepts format
OPAM [92]	2017	Global	NA	By-design, FGCV	Yes	part-based
TCAV [66]	2018	Global	Post-hoc	Neural network	Yes	same as input
ACE [39]	2019	Global	Post-hoc	Neural network	No	segmented parts
Zhang et al. [116]	2021	Global	Post-hoc	Neural network	No	segmented parts
CRAFT [35]	2022	Global	Post-hoc	Neural network	No	crops
Particul [114]	2017	Global	NA	By-design, FGCV	Yes	part-based

Table 1. Comparison table between the different natural example-based formats and methods. NA: Not applicable, FGCV: Fine-grained computer vision

8 Acknowledgments

This work has been supported by the French government under the "France 2030" program as part of the SystemX Technological Research Institute. This work was conducted as part of the Confiance.AI program, which aims to develop innovative solutions for enhancing the reliability and trustworthiness of AI-based systems. Additional funding was provided by ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANR-19-PI3A-0004).

This work has benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French "Investing for the Future – PIA3" program under the Grant agreement ANR-19-P3IA-0004. The authors gratefully acknowledge the support of the DEEL project.[3]

[3] `https://www.deel.ai/`

We are also thankful to the DEEL's core team for their expertise and feedback. A.M. Picard, D. Vigouroux, C. Friedrich, V. Mussot, and Y. Prudent.

Finally, we are thankful to the authors who accepted our use of their figures. E.M Kenny and M.T. Keane [61,62], F. Liu [93], B. Kim [64], and T. Fel [35].

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. (1994)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (xai). IEEE access (2018)
3. Agarwal, N., Bullins, B., Hazan, E.: Second-order stochastic optimization for machine learning in linear time. Journal of Machine Learning Research (2017)
4. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion (2020)
5. Aryal, S., Keane, M.T.: Even if explanations: Prior work, desiderata & benchmarks for semi-factual xai. arXiv preprint arXiv:2301.11970 (2023)
6. Augustin, M., Boreiko, V., Croce, F., Hein, M.: Diffusion visual counterfactual explanations. In: NeurIPS (2022)
7. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one (2015)
8. Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., Krause, A.: Streaming submodular maximization: Massive data summarization on the fly. In: KDD (2014)
9. Bae, J., Ng, N., Lo, A., Ghassemi, M., Grosse, R.B.: If influence functions are the answer, then what is the question? NeurIPS (2022)
10. Barshan, E., Brunet, M.E., Dziugaite, G.K.: Relatif: Identifying explanatory training samples via relative influence. In: AISTATS (2020)
11. Basu, S., Pope, P., Feizi, S.: Influence functions in deep learning are fragile. In: ICLR (2021)
12. Basu, S., You, X., Feizi, S.: On second-order group influence functions for black-box predictions. In: ICML (2020)
13. Bennett, J.: A philosophical guide to conditionals. Clarendon Press (2003)
14. Bhatia, N., et al.: Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085 (2010)
15. Bichindaritz, I., Marling, C.: Case-based reasoning in the health sciences: What's next? Artificial intelligence in medicine (2006)
16. Bien, J., Tibshirani, R.: Prototype selection for interpretable classification. The Annals of Applied Statistics (2011)
17. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR (2003)
18. Byrne, R.M.: Counterfactual thought. Annual review of psychology (2016)
19. Byrne, R.M.: Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In: IJCAI (2019)
20. Cambria, E., Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N.: A survey on xai and natural language explanations. IPM (2023)
21. Caruana, R., Kangarloo, H., Dionisio, J., Sinha, U., Johnson, D.: Case-based explanation of non-case-based learning methods. In: AMIA Symposium (1999)
22. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: deep learning for interpretable image recognition. NeurIPS (2019)
23. Chen, Y., Li, B., Yu, H., Wu, P., Miao, C.: Hydra: Hypergradient data relevance analysis for interpreting deep neural networks. In: AAAI (2021)
24. Cook, R.D., Weisberg, S.: Residuals and influence in regression. New York: Chapman and Hall (1982)

25. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE TIT (1967)
26. Cummins, L., Bridge, D.: Kleor: A knowledge lite approach to explanation oriented retrieval. CAI (2006)
27. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: ICCBR (2003)
28. Dasarathy, B.V.: Nearest unlike neighbor (nun): an aid to decision confidence estimation. Optical Engineering (1995)
29. De Mantaras, R.L., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., et al.: Retrieval, reuse, revision and retention in case-based reasoning. KER (2005)
30. Doyle, D., Cunningham, P., Bridge, D., Rahman, Y.: Explanation oriented retrieval. In: ECCBR (2004)
31. Doyle, D., Cunningham, P., Walsh, P.: An evaluation of the usefulness of explanation in a case-based reasoning system for decision support in bronchiolitis treatment. Computational Intelligence (2006)
32. Dudani, S.A.: The distance-weighted k-nearest-neighbor rule. IEEE TSMC (1976)
33. Fel, T., Cadène, R., Chalvidal, M., Cord, M., Vigouroux, D., Serre, T.: Look at the variance! efficient black-box explanations with sobol-based sensitivity analysis. NeurIPS (2021)
34. Fel, T., Ducoffe, M., Vigouroux, D., Cadène, R., Capelle, M., Nicodème, C., Serre, T.: Don't lie to me! robust and efficient explainability with verified perturbation analysis. In: CVPR (2022)
35. Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., Serre, T.: Craft: Concept recursive activation factorization for explainability. In: CVPR (2022)
36. Feldman, V., Zhang, C.: What neural networks memorize and why: Discovering the long tail via influence estimation. NeurIPSs (2020)
37. Filho, R.M., Lacerda, A.M., Pappa, G.L.: Explainable regression via prototypes. ACM TELO (2023)
38. Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive science (1983)
39. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. NeurIPS (2019)
40. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. JCGS (2015)
41. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: ICLR (2015)
42. Goodman, N.: The problem of counterfactual conditionals. The journal of philosophy (1947)
43. Gurumoorthy, K.S., Dhurandhar, A., Cecchi, G., Aggarwal, C.: Efficient data representation by selecting prototypes with importance weights. In: ICDM (2019)
44. Halpern, J.Y., Pearl, J.: Causes and explanations: A structural-model approach. part ii: Explanations. BJPS (2005)
45. Hampel, F.R.: The influence curve and its role in robust estimation. JASA (1974)
46. Hanawa, K., Yokoi, S., Hara, S., Inui, K.: Evaluation of similarity-based explanations. In: ICLR (2021)
47. Hara, S., Nitanda, A., Maehara, T.: Data cleansing for models trained with sgd. NeurIPS (2019)
48. Hase, P., Chen, C., Li, O., Rudin, C.: Interpretable image recognition with hierarchical prototypes. In: HCOMP (2019)

49. Hastie, T.: The elements of statistical learning: data mining, inference, and prediction. Springer (2009)
50. Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W.: Explainable ai methods-a brief overview. In: xxAI-Beyond Explainable AI (2022)
51. Humer, C., Hinterreiter, A., Leichtmann, B., Mara, M., Streit, M.: Comparing Effects of Attribution-based, Example-based, and Feature-based Explanation Methods on AI-Assisted Decision-Making. preprint, Open Science Framework (2022)
52. Ilyas, A., Park, S.M., Engstrom, L., Leclerc, G., Madry, A.: Datamodels: Predicting predictions from training data. In: ICML (2022)
53. Jeyakumar, J.V., Noor, J., Cheng, Y.H., Garcia, L., Srivastava, M.: How can i explain this to you? an empirical study of deep neural network explanation methods. NeurIPS (2020)
54. Karimi, A.H., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: AISTATS (2020)
55. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: CVPR (2020)
56. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley & Sons (2009)
57. Keane, M.T., Kenny, E.M.: The twin-system approach as one generic solution for xai: An overview of ann-cbr twins for explaining deep learning. In: IJCAI Workshop on XAI (2019)
58. Keane, M.T., Kenny, E.M., Delaney, E., Smyth, B.: If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. arXiv preprint arXiv:2103.01035 (2021)
59. Keane, M.T., Smyth, B.: Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable ai (xai). In: ICCBR (2020)
60. Kenny, E.M., Keane, M.T.: Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ann-cbr twins for xai. In: IJCAI (2019)
61. Kenny, E.M., Keane, M.T.: Explaining deep learning using examples: Optimal feature weighting methods for twin systems using post-hoc, explanation-by-example in xai. KBS (2021)
62. Kenny, E.M., Keane, M.T.: On generating plausible counterfactual and semi-factual explanations for deep learning. In: AAAI (2021)
63. Khanna, R., Kim, B., Ghosh, J., Koyejo, S.: Interpreting black box predictions using fisher kernels. In: AISTATS (2019)
64. Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! Criticism for Interpretability. In: NeurIPS (2016)
65. Kim, B., Rudin, C., Shah, J.A.: The bayesian case model: A generative approach for case-based reasoning and prototype classification. NeurIPS (2014)
66. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: ICML (2018)
67. Kim, S., Jeong, M., Ko, B.C.: Lightweight surrogate random forest support for model simplification and feature relevance. Applied Intelligence (2022)
68. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: NeurIPS (2017)
69. Koh, P.W.W., Ang, K.S., Teo, H., Liang, P.S.: On the accuracy of influence functions for measuring group effects. NeurIPS (2019)

70. Kohonen, T.: The self-organizing map. IEEE (1990)
71. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature (1999)
72. Lewis, D.: Counterfactuals. John Wiley & Sons (1973)
73. Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: NAACL (2010)
74. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: ACL HLT (2011)
75. Lipton, P.: Contrastive explanation. Royal Institute of Philosophy Supplements (1990)
76. Marchette, C.E.P.D.J., Socolinsky, J.G.D.D.A.: Classification using class cover catch digraphs. Journal of Classification (2003)
77. Martens, J.: Deep learning via hessian-free optimization. In: ICML (2010)
78. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence (2019)
79. Miller, T.: Contrastive explanation: A structural-model approach. KER (2021)
80. Ming, Y., Xu, P., Qu, H., Ren, L.: Interpretable and steerable sequence learning via prototypes. In: KDD (2019)
81. Mirzasoleiman, B., Karbasi, A., Badanidiyuru, A., Krause, A.: Distributed submodular cover: Succinctly summarizing massive data. NeurIPS (2015)
82. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
83. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: ACM FAccT (2020)
84. Nauta, M., Jutte, A., Provoost, J., Seifert, C.: This looks like that, because... explaining prototypes for interpretable image recognition. In: PKDD workshop (2022)
85. Nauta, M., Van Bree, R., Seifert, C.: Neural prototype trees for interpretable fine-grained image recognition. In: CVPR (2021)
86. Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB (1994)
87. Nugent, C., Cunningham, P.: A case-based explanation system for black-box systems. The Artificial Intelligence Review (2005)
88. Nugent, C., Doyle, D., Cunningham, P.: Gaining insight through case-based explanation. JIIS (2009)
89. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017)
90. Papernot, N., McDaniel, P.: Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.04765 (2018)
91. Park, J.H., Im, K.H., Shin, C.K., Park, S.C.: Mbnr: case-based reasoning with local feature weighting by neural network. Applied Intelligence (2004)
92. Peng, Y., He, X., Zhao, J.: Object-part attention model for fine-grained image classification. IEEE TIP (2017)
93. Pruthi, G., Liu, F., Kale, S., Sundararajan, M.: Estimating training data influence by tracing gradient descent. NeurIPS (2020)
94. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. In: KDD (2016)
95. Roese, N.J., Olson, J.M.: Counterfactual thinking: A critical overview. What might have been: The social psychology of counterfactual thinking (1995)
96. Sani, S., Wiratunga, N., Massie, S.: Learning deep features for knn-based human activity recognition. CEUR Workshop (2017)
97. Saunshi, N., Gupta, A., Braverman, M., Arora, S.: Understanding influence functions and datamodels via harmonic analysis. ICLR (2023)

98. Schank, R.C.: Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge University Press (1983)
99. Schioppa, A., Zablotskaia, P., Vilar, D., Sokolov, A.: Scaling up influence functions. In: AAAI (2022)
100. Schoenborn, J.M., Weber, R.O., Aha, D.W., Cassens, J., Althoff, K.D.: Explainable case-based reasoning: a survey. In: AAAI Workshop (2021)
101. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: EuroCOLT (2001)
102. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
103. Serrurier, M., Mamalet, F., Fel, T., Béthune, L., Boissin, T.: When adversarial attacks become interpretable counterfactual explanations. arXiv preprint arXiv:2206.06854 (2022)
104. Shin, C.K., Yun, U.T., Kim, H.K., Park, S.C.: A hybrid approach of neural network and memory-based learning to data mining. IEEE TNN (2000)
105. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML (2017)
106. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
107. Sui, Y., Wu, G., Sanner, S.: Representer point selection via local jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models. In: NeurIPS (2021)
108. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML (2017)
109. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
110. Thompson, V.A., Byrne, R.M.: Reasoning counterfactually: making inferences about things that didn't happen. JEP LMC (2002)
111. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: A review. arXiv preprint arXiv:2010.10596 (2020)
112. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating xai: A comparison of rule-based and example-based explanations. Artificial Intelligence (2021)
113. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. Harvard JOLT (2017)
114. Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, M.C.: Particul: Part identification with confidence measure using unsupervised learning. arXiv preprint arXiv:2206.13304 (2022)
115. Yeh, C.K., Kim, J., Yen, I.E.H., Ravikumar, P.K.: Representer point selection for explaining deep neural networks. NeurIPS (2018)
116. Zhang, R., Madumal, P., Miller, T., Ehinger, K.A., Rubinstein, B.I.: Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In: AAAI (2021)