Towards Abductive Latent Explanations

Jules Soria[1,*], Zakaria Chihani[1], Julien Girard-Satabin[1], Alban Grastien[1], Romain Xu-Darme[1] and Daniela Cancila[1]

[1]Université Paris-Saclay, CEA List, F-91120, Palaiseau, France

Abstract

Case-based reasoning networks are machine-learning models that make predictions based on similarity between the input and prototypical parts of training samples, called prototypes. Such models are able to explain each decision by pointing to the prototypes that contributed the most to the final outcome. As the explanation is a core part of the prediction, they are often qualified as "interpretable by design". While promising, we show that such explanations are sometimes misleading, which hampers their usefulness in safety-critical contexts. In particular, several instances may lead to different predictions and yet have the same explanation. Drawing inspiration from the field of formal eXplainable AI (formal XAI), we propose Abductive Latent Explanations (ALEs), a formalism to express sufficient conditions on the intermediate (latent) representation of the instance that imply the prediction. Our approach combines the inherent interpretability of case-based reasoning models and the guarantees provided by formal XAI. We propose a solver-free and scalable algorithm for generating ALEs and present the feasibility of our approach on the CUB 200 dataset for the task of fine-grained image classification.

Keywords

explainable artificial intelligence, case-based reasoning, formal XAI, interpretable ML, trustworthy AI

1. Introduction

A widely adopted approach to explain neural network decisions is to analyze the decisions of a model after its training, in a post-hoc fashion [1]. For neural networks in computer vision, a common line of work consists in computing the most relevant pixels by backpropagating gradients on the input space for a given sample [2, 3, 4, 5].

However, such approaches are not without flaws. They have been shown to be sensitive to malign manipulations [6], raising questions on their usefulness in an adverse setting [7]. Moreover, some attribution methods may not correlate to the actual model behavior [8, 9] - raising questions on what they actually aim to explain - or display irrelevant features [10]. Finally, their usefulness on actual scenarios with actual humans has been questioned [11].

To overcome such limitations, the emerging field of *formal explainable AI (FXAI)* [12, 13, 14, 15, 16] provides a rigorous framework to characterize and build explanations. A particular line of work exemplified by [13, 14, 17, 18] builds upon the framework of *abductive reasoning*: explanations are defined as a subset of features that are sufficient to justify the model decision. In particular, it is possible to produce subset-optimal explanations within this framework, such that removing any single feature from such explanations changes the classifier's decision. FXAI provides strong guarantees on the relevance of features in the explanation, thanks to the use of automated provers that directly query the model. As such, FXAI represents a good compromise between compactness and correctness — which are deemed important characteristics of an explanation [19, 20] — and a stepping stone in line with new regulations. Indeed, to reach compliance with the AI Act (article 86 states *"Any affected person subject to a decision which is taken by the deployer on the basis of the output from a high-risk AI [. . .] shall have the*

TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.

[*] Corresponding author.

jules.soria@cea.fr (J. Soria); zakaria.chihani@cea.fr (Z. Chihani); julien.girard2@cea.fr (J. Girard-Satabin); alban.grastien@cea.fr (A. Grastien); romain.xu-darme@cea.fr (R. Xu-Darme); daniela.cancila@cea.fr (D. Cancila)
https://julien.girard-satabin.fr (J. Girard-Satabin); http://www.grastien.net/ban (A. Grastien)
0009-0009-6730-7406 (J. Soria); 0009-0004-8915-4774 (Z. Chihani); 0000-0001-6374-3694 (J. Girard-Satabin); 0000-0001-8466-8777 (A. Grastien); 0000-0002-8630-5635 (R. Xu-Darme); 0000-0002-3483-7947 (D. Cancila)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-16

right to obtain [. . .] clear and meaningful explanations of the role of the AI system [. . .]), the correctness of explanations is of paramount importance.

Although FXAI is a promising approach, it suffers from two main shortcomings:

1. these approaches rely on expensive prover calls, which limits their scalability on realistic computer vision tasks (the associated problems are generally NP-complete [21]);

2. abductive FXAI provides explanations at the *feature-level*, which, for typical computer vision applications, is a pixel. We argue that the pixel-level is not the correct level of abstractions for the human final user of the explanation. Pixel-level explanations rely on the model's perception of the problem, setting a knowledge gap between the human and the machine [20]. Higher-level reasoning like prototypes or concepts allow generalizing facts towards higher-level reasoning [22].

An orthogonal approach involves implementing *case-based reasoning*. Under this setting, the neural network is designed to justify its decision by exposing examples from its dataset that are similar to the new sample. Such approach is exemplified by prototype learning [23, 24, 25, 26, 27, 28] or concept learning [29, 30, 31, 32, 33, 34].

In this paradigm, explanations justify the decision by exposing a certain number of prototypes or concepts to the user. One major drawback is that the presented prototypes are usually not sufficient to entail the decision. Indeed, the number of prototypes in the popular architecture ProtoPNet [35] is fixed as an arbitrary hyperparameter. This produces explanations that may omit relevant prototypes taken for the decision, resulting in misleading explanations.

Table 1 summarizes and qualitatively compares these methods, highlighting how our proposed approach achieves (at least partially) three key properties: interpretability, faithfulness, and sufficiency.

Table 1
Qualitative Comparison of Explanations

Method	**Interpretability**	**Faithfulness**	**Sufficiency**
Saliency Maps	✓	×	×
ProtoPNet	✓	✓	×
Formal AXp	×	✓	✓
Prototype top-k (Ours)	$\sim$	✓	✓

In this paper, we make the following contributions:

1. We propose a generic framework to describe **Abductive Latent Explanations (ALE)** for prototype-based networks. This formalism can be instantiated for any neural network architecture that relies on identifying concept-rich prototypical parts and reasons on their activations to reach a final decision.

2. We leverage uncertainties on prototype activations to circumvent the need for costly prover calls. This allows us to produce explanations that are guaranteed to secure the model's prediction in a computationally efficient manner.

2. Background

Our approach requires a system with the following components:

1. an **image encoder** whose goal is to map the input image to a latent representation space;

2. a **prototype layer** that measures the distance between the *latent* representation of the image and learned prototypes, and assigns to them an *activation* score;

3. a **decision layer** that gives each class a score and chooses the class with the highest score as its prediction.

In the rest of the paper, we capitalize on the prototype and decision layers as defined in the original ProtoPNet architecture [35] and follow-ups [27] as they represent the state-of-the-art of prototype-based models, and implementations are readily available [36].

To give an intuition on abductive explanations in the latent space, we will provide a running example after introducing the notations used in the paper, further illustrated in Figure 1.

Notations

We briefly summarize here the notations used throughout the paper. The predictor function $\kappa : \mathcal{F} \to \mathcal{C}$ is defined as $\kappa = \mathrm{argmax} \circ h \circ \mathbf{a} \circ f$, where f maps inputs to the latent space, $\mathbf{a}$ computes prototype activations, and h produces class logits. The similarity matrix $\mathbf{sim}(\mathbf{z}, \mathbf{p})$, with $\mathbf{p} = (\mathbf{p}_j)_{j\in\mathbf{P}}$, has entries $\mathrm{sim}(\mathbf{z}_l, \mathbf{p}_j)$ for $l \in \mathbf{L}$, $j \in \mathbf{P}$. Explanations $\mathcal{E} \subseteq \mathbf{P}$, as detailed in Section 3.

Running example

We consider a 2-class classification problem with `emperor_penguin` and `royal_penguin`.

During inference, an image input $\mathbf{v}$ first goes through an **image encoder** and is represented as $\mathbf{z}$, i.e. an object in the latent space $\mathcal{Z}$, which sums up the important concepts *locally* identified. This latent representation is composed of four $(H_1 \times W_1)$ latent vectors, where width $W_1 = 2$, and height $H_1 = 2$ (as shown in Figure 1).

A *prototype* is a latent vector from a training image that the training procedure has identified as prototypical of some class. In this example, we assume that the training procedure computed five prototypes. We further assume that these prototypes can be linked to the following concepts: `emperor_beak`, `emperor_yellow_neck_patch`, `black_back_white_belly`, `royal_beak`, and `royal_yellow_crests` (these labels are solely used to make the example comprehensible; they are neither inputs of the training nor produced by it). We use $\mathbf{p}$ to refer to the vector of prototypes and $\mathbf{p}_i$ for the ith prototype.

Secondly, the **prototype layer** $\mathrm{sim}(\mathbf{z}, \mathbf{p})$ computes a similarity score between each latent vector (row) and the prototypes (column):

$$\mathrm{sim}(\mathbf{z}, \mathbf{p}) = \begin{bmatrix} 1. & 0. & 1. & 0. & 1. \\ 1. & 3. & 0. & 1. & 2. \\ 0. & 1. & 0. & 2. & 1. \\ 0. & 0. & 1. & 8. & 0. \end{bmatrix}$$

where higher value means higher similarity. In this example, the fourth prototype (`royal_beak`, fourth column) has been strongly recognized in the bottom-right part of the image (fourth row), hence a similarity score of 8. Similarly, the second prototype (`emperor_yellow_neck_patch`) is moderately recognized, because of similarity score of 3., while the other prototypes are absent (score 2 or lower). For each prototype, the ProtoPNet architecture is only interested in the top similarity, and computes the column-wise maximum value of $\mathrm{sim}(\mathbf{z}, \mathbf{p})$ called the *activation vector*

$$\mathbf{a}(\mathbf{z}) = \begin{bmatrix} 1. & 3. & 1. & 8. & 2. \end{bmatrix}$$

These values mean that the network detected a clear royal beak and what appears to be an emperor yellow neck patch while no other attribute one would expect from a penguin stands out.

Thirdly, in the **decision layer**, the prototype *activation* vector is fed to a fully connected linear layer with learned parameters W where

$$W = \begin{bmatrix} 10. & 10. & 7. & 0. & 0. \\ 0. & 0. & 5. & 10. & 15. \end{bmatrix}$$

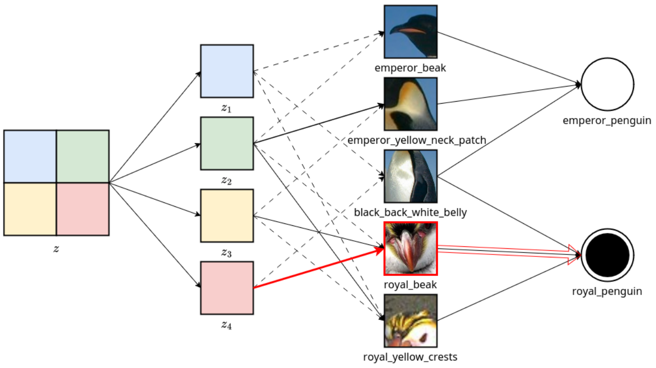

Figure 1: Running example for two classes: `emperor_penguin` and `royal_penguin`

Matrix W indicates that prototypes 1 and 2 are typical of emperor penguins (weights 10. against 0.) and prototypes 4 and 5 of royal penguins (weights 10. and 15. versus 0.), while prototype 3 can be detected in both classes. The decision layer's computation returns the final class scores:

$$h(\mathbf{a}(\mathbf{z})) = W\mathbf{a}(\mathbf{z}) = \begin{bmatrix}47. & 115.\end{bmatrix}$$

and the classifier thus outputs the second class: $\kappa_{\mathcal{Z}}(\mathbf{z}) = \underset{i}{\text{argmax}}\ h_i(\mathbf{a}(\mathbf{z})) = 2$, i.e., `royal_penguin`.

ProtoPNet returns as an explanation the k prototypes with highest activation. In the running example, if $k = 1$, the explanation is $\mathcal{E} = \{4\}$ which involves $\{\mathbf{p}_4 : 8.\}$. We highlight that the above explanation implicitly entails that the score of all other prototypes is 8. or below. At the first glance, such an explanation can appear appropriate: the bird on the image is classified as a `royal_penguin` because a royal beak (typical of royal penguins) has been observed.

However, a closer examination reveals that the explanation is wrong for some instances. Counter-example: let be an image having latent representation $\mathbf{z}'$ such that the activation evaluates to

$$\mathbf{a}(\mathbf{z}') = \begin{bmatrix}6. & 7. & 1. & 8. & 2.\end{bmatrix}$$

The classification is $\kappa(\mathbf{z}) = \arg\max \begin{bmatrix}137. & 115.\end{bmatrix} = 1$ while the previous explanation is still applicable to `emperor_penguin` "because it shows a royal beak". In other words, the explanation is misleading or *optimistic* as discussed in [37]. Hence, the need for formal guarantees on prototypes explanations - abductive latent-based explanations.

3. Abductive explanations in the prototype space

An Abductive explanation (AXp) for $\mathbf{v}$ is traditionally defined as a condition on the input of a classifier satisfied by the current instance such that all instances that satisfy this condition yield the same output [38]. In other words, an AXP defines *preconditions* on the inputs of a classifier, yielding a *postcondition* on the classification of said classifier [39]. Formally, given an input space $\mathcal{F}$, a predictor κ from $\mathcal{F}$ to $\mathcal{C}$, and an input instance $\mathbf{v} \in \mathcal{F}$ with prediction $c = \kappa(\mathbf{v}) \in \mathcal{C}$, a *formal explanation* is a precondition $\phi_{\mathcal{E}}(\mathbf{x}, \mathbf{v})$ over input $\mathbf{x}$ satisfied by $\mathbf{v}$ such that this holds:

$$\forall \mathbf{x} \in \mathcal{F}. \quad \phi_{\mathcal{E}}(\mathbf{x}, \mathbf{v}) \Rightarrow (\kappa(\mathbf{x}) = c).$$

In previous works [13], the explanation is represented as a conjunction $\mathcal{E}$ of input variables (features), and the precondition simply states that the assignments of these variables should match those of $\mathbf{v}$:

$$\phi_{\mathcal{E}}(\mathbf{x}, \mathbf{v}) = \bigwedge_{i \in \mathcal{E}} (\mathbf{x}_i = \mathbf{v}_i),$$

Our main contribution is the extension of the definition of AXps to an *arbitrary latent space*.

In the case of image recognition, an explanation is a subset of the image pixels (if grayscale, else pixel color channels). While this explanation is correct—i.e., any image that includes these specific pixels will be classified as the input image—its interpretability is questionable.

Since explanations are preconditions on the input space of the classifier, in the context of case-based reasoning, we can define such precondition on the *input of the latent classifier*:

Definition 1 (Abductive Latent Explanation (ALE))**.** *Given an input instance* $\mathbf{v}$ *with latent representation* $f(\mathbf{v})$*, an* abductive latent explanation *is a subset of (latent) features* $\mathcal{E}$ *that entails a precondition* $\phi_{\mathcal{E}}(f(\mathbf{x}), f(\mathbf{v}))$ *over* $f(\mathbf{x})$ *satisfied by* $f(\mathbf{v})$*.*

Given an input space $\mathcal{F}$ *and a predictor* $\kappa = \kappa_{\mathcal{Z}} \circ f$ *from* $\mathcal{F}$ *to* $\mathcal{C}$*, the explanation is* correct *if the following holds:*

$$\forall \mathbf{x} \in \mathcal{F}. \quad \phi_{\mathcal{E}}(f(\mathbf{x}), f(\mathbf{v})) \Rightarrow (\kappa(\mathbf{x}) = c).$$

We formalize the ProtoPNet's definition of an explanation:

Definition 2 (ProtoPNet Explanation)**.** *Given a set* $\mathbf{P}$ *of prototype indices, a* ProtoPNet explanation $\mathcal{E} \subseteq \mathbf{P}$ *is a subset of indices that implicitly represents the ALE*

$$\phi_{\mathcal{E}}(\mathbf{z}, \mathbf{y}) = \left(\bigwedge_{i \in \mathcal{E}} \mathbf{a}_i(\mathbf{z}) = \mathbf{a}_i(\mathbf{y}) \right) \wedge \left(\bigwedge_{j \notin \mathcal{E}, i \in \mathcal{E}} \mathbf{a}_j(\mathbf{z}) \leq \mathbf{a}_i(\mathbf{y}) \right).$$

Compared to the pixel-based abductive explanations, we consider that ALEs are more interpretable as they refer to the concepts that humans are able to manipulate. Furthermore, compared to relevance-based explanations, ALEs provide formal guarantees as it is impossible to come up with misleading explanations.

We note however that ProtoPNet explanations (from Definition 2), as will also be the case for the explanations introduced in the next section, implicitly include the clause "under the assumption that function f is a proper encoding of the input image into a latent space". Indeed, some important steps of the procedure for computing the similarity scores are hidden from the user. Nonetheless, we consider that these explanations are an important improvement over the existing work.

4. Building abductive explanations in the prototype space

The explanations proposed in the previous section assume that the human agrees with the fact that the activation scores in the returned explanation are the top-k elements. However, the explanation does not 'prove' that this fact holds. In this section, we propose an extension of the definition of an explanation that relies on the bounds of the final class logits.

Here, explanations will be used to prove bounds over activations. Thus, an explanation will implicitly entail a condition of the form

$$\phi_{\mathcal{E}}(\mathbf{z}, \mathbf{y}) = \left(\bigwedge_{j=1}^{m} \mathbf{a}_j(\mathbf{z}) \in \left[\underline{\mathbf{a}}_{\mathcal{E},j}, \overline{\mathbf{a}}_{\mathcal{E},j} \right] \right).$$

Top-k explanations This paradigm is the one used implicitly by the original explanation [35] provided by ProtoPNet. It traverses the prototype activation scores in decreasing order, with the added knowledge that, for that prototype, the similarity scores with the other latent space feature vectors are lesser than the activation score (result of the `max` function). In that scenario, we obtain $\mathbf{a}_{\mathcal{E}}$ from $\mathcal{E}$ by:

$$\begin{aligned} \forall j \in \mathcal{E}. &\quad \underline{\mathbf{a}}_{\mathcal{E},j} = \overline{\mathbf{a}}_{\mathcal{E},j} = \mathbf{a}_j(\mathbf{y}) \\ \forall j \notin \mathcal{E}. &\quad \underline{\mathbf{a}}_{\mathcal{E},j} = 0 \quad \text{and} \quad \overline{\mathbf{a}}_{\mathcal{E},j} = \min_{i \in \mathcal{E}} \mathbf{a}_i(\mathbf{y}) \end{aligned}$$

Constructing Abductive Latent Explanations (ALE)

Given a candidate explanation $\mathcal{E} \subseteq \mathbf{P}$, the constrained **prototype activation** space is

$$\mathbf{a}_{\mathcal{E}} = [0, \max_{j} \ \overline{\mathbf{a}}_{\mathcal{E},j}]^m$$

Constructing an abductive latent explanation $\mathcal{E}$ in the prototype activation space involves defining preconditions on a generic activation vector $\mathbf{a} \in \mathbb{R}^m$. These preconditions are derived from $\mathcal{E}$ and $\mathbf{a}(\mathbf{y})$. Crucially, they must guarantee that any activation vector $\mathbf{a}$ satisfying these conditions results in the same predicted class c:

$$\forall \mathbf{a} \in \mathbf{a}_{\mathcal{E}} \subseteq \mathbb{R}^m : \mathbf{a} \models \ \phi_{\mathcal{E}}, \quad \operatorname*{argmax}_{k \in \mathcal{C}} h_k(\mathbf{a}) = c$$

These preconditions effectively define a constrained region - or set of constraints $\mathbf{a}_{\mathcal{E}}$ within the activation space $\mathbb{R}^m$ - such that $\mathbf{a}(\mathbf{y})$ satisfies these constraints, and all vectors within this region yield the prediction c. By making Assumption 1, our proposed explanation method can be applied to **any** prototype-based model that has a Linear layer for its decision head h that links the prototype activations to the class logits in the form of a weighted sum.

Assumption 1 (Linear Logit Difference). *For any two classes k, c, the difference between their logit functions is linear in the prototype activation vector $\mathbf{a}$:*

$$h_k(\mathbf{a}) - h_c(\mathbf{a}) = \sum_{j=1}^{m} (w_{jk} - w_{jc}) \ \mathbf{a}_j + (b_k - b_c)$$

for some weights w_{jk}, w_{jc} and biases b_k, b_c.

Definition 3 (Maximally Class-Favoring Element within $\mathbf{a}_{\mathcal{E}}$). *For a given explanation $\mathcal{E}$, predicted class c, and alternative class $k \neq c$, the* maximally (k/c)-favoring **prototype activation** vector within $\mathbf{a}_{\mathcal{E}}$ *is denoted by $\mathbf{a}^*_{\mathcal{E}}(k, c)$ and satisfies:*

$$\forall \mathbf{a} \in \mathbf{a}_{\mathcal{E}}. \quad h_k(\mathbf{a}^*_{\mathcal{E}}(k, c)) - h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{a}) - h_c(\mathbf{a})$$

Under Assumption 1, its components $(\mathbf{a}^*_{\mathcal{E}}(k, c))_j$ are constructed as:

$$(\mathbf{a}^*_{\mathcal{E}}(k, c))_j = \begin{cases} (\overline{\mathbf{a}}_{\mathcal{E}})_j & \text{if } w_{jk} \geq w_{jc} \\ (\underline{\mathbf{a}}_{\mathcal{E}})_j & \text{if } w_{jk} < w_{jc} \end{cases}$$

One way to intuitively understand this item is to view it as the element that *satisfies* the condition expressed by the explanation, and that causes the most the classifier from reaching the initial prediction (in favor of a specific different class).

Definition 4 (Class-wise Prediction Domination within $\mathbf{a}_{\mathcal{E}}$). *For explanation $\mathcal{E}$, predicted class c, and alternative class $k \neq c$, we say c* dominates k *within $S_{\mathcal{E}}$, denoted $\psi_{\mathcal{E}}(k, c)$, if the logit of c is greater than or equal to the logit of k even for the maximally (k/c)-favoring vector:*

$$\psi_{\mathcal{E}}(k, c) \iff h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{a}^*_{\mathcal{E}}(k, c))$$

Definition 5 (Total Prediction Domination within $\mathbf{a}_{\mathcal{E}}$ (Explanation Verification)). *A candidate explanation $\mathcal{E}$ is considered* verified *if class c dominates all other classes $k \neq c$ within the constrained space $S_{\mathcal{E}}$:*

$$\textit{Verify}(\mathcal{E}) \iff \forall k \in \{1, \ldots, C\} \setminus \{c\}, \quad \psi_{\mathcal{E}}(k, c)$$

In Definition 4 we say that a class (different from the predicted class) is *verified* if its *Maximally Class-Favoring Element* does not entail a different classification. In Definition 5 we say that the explanation *verifies* the prediction if **all** classes are *verified.*

According to Theorem 1, if an explanation $\mathcal{E}$ is verified according to Definition 5, it satisfies the original abductive explanation definition (Definition 1).

Theorem 1 (Verified Explanation is Sufficient). *Let $\mathcal{E}$ be a candidate explanation for the decision $(\mathbf{v}, c)$. If $\mathcal{E}$ is verified according to Definition 5 (i.e., Verify($\mathcal{E}$) is true), then $\mathcal{E}$ is a valid abductive explanation according to Definition 1.*

Proof. Found in the Appendix. □

Algorithm 1 Generating ALE

```
function GenerateALE($\mathbf{v}, c$)
    $\mathcal{E} = \emptyset$
    UnverifiedClasses $= \mathcal{C} \setminus \{c\}$
    $\mathcal{A} \leftarrow$ Sort($\mathbf{a}(\mathbf{v})$)
    while UnverifiedClasses $\neq \emptyset$ do
        $j =$ NextPrototype($\mathcal{E}, \mathcal{A}$)
        $\mathcal{E} \leftarrow \mathcal{E} \cup j$
        for $k \in$ UnverifiedClasses do
            CEx $\leftarrow \mathbf{a}^*_{\mathcal{E}}(k, c)$
            if $h_c$(CEx) $> h_k$(CEx) then
                UnverifiedClasses $\leftarrow$ UnverifiedClasses $\setminus\{k\}$
            end if
        end for
    end while
    return $\mathcal{E}$
end function
```

We present the pseudo-code for Algorithm 1 used to generate an ALE in Section 4. The function `NextPrototype` we use returns the prototype with the highest activation score among the ones not included already in the explanation $\mathcal{E}$. Following the reasoning used to derive Theorem 1 and the associated definitions 3, 4, 5, we iteratively add the highest remaining activation score until it becomes impossible to create a sample that satisfies the explanation and has a different predicted class.

5. Experimental Study

Computational Resources

Our experiments were conducted on a SLURM-managed computing cluster, primarily utilizing GPU-equipped nodes for computationally intensive tasks, with each node featuring 48 cores at 2.6GHz, 187GB of RAM, four NVIDIA V100 32GB GPUs, and a 1.9TB local SSD.

Methodology

Our study uses the Case-Based Reasoning Network (CaBRNet) framework [36] to train and manipulate the model. CaBRNet integrates concepts from case-based reasoning into a deep learning framework, leveraging learned prototypes for classification.

We use a model trained by CaBRNet, with a VGG-19 backbone pretrained on ImageNet [40], and 10 prototypes per class (as is the standard for case-based reasoning training). For reference, it obtains 68% accuracy on the test set.

Dataset

The Caltech-UCSD Birds-200-2011 dataset (CUB200) [41] contains 11,788 images spanning 200 species of North American birds (thus, the total number of prototypes was $m = 10 \times 200 = 2000$). It is a widely used benchmark for fine-grained image classification, featuring detailed annotations including bounding boxes and part locations. We used the standard train/test splits provided by the dataset creators and followed common preprocessing procedures, such as resizing and cropping and data augmentation techniques, described in [35].

Results

Table 2 summarizes: the average sizes of explanations on CUB200 broken down by correct and incorrect classifications; and the average time to compute an explanation.

	Mean Explanation Size	Correct Samples	Incorrect Samples	Computation Time (s)
Top_k	427	306	673	0.6

Table 2
Summary of Mean Explanation Sizes

Discussion

We show that the 10 most activated prototypes are never enough to guarantee the decision. Indeed, in practice 427 similarity scores are required on average to generate an ALE. This result is significant as it entails that (for the model used) **all** previous ProtoPNet explanations were misleading, or optimistic ; i.e., there exists samples that match the provided explanations and had a distinct classification.

Furthermore, Table 2 shows that, for samples that are incorrectly classified, the average explanation size is much higher. This would mean that information about a sample and the data distribution can be extracted from the computation of ALEs.

However, it would be dishonest to say that an explanation that contains hundreds of prototypes and associated activation scores is *interpretable*, especially to a human receiver. Indeed, the cardinality of an explanation influences its interpretability and usefulness ; a larger explanation will be harder to follow and understand for the user [20].

We respond to this issue in two ways: first, our proposed method primarily focuses on generating explanations that are *sufficient* to guarantee the decision, as this is the key property we want in an explanation. This means that, if there is a trade-off between explanation size and sufficiency, we will always prioritize sufficiency. Then, large sufficient explanations can be due to the model itself. Although one of the ProtoPNet's training objective is to have latent representation vectors of training images far away from other classes' prototypes [35], it can have in the end a very “disorganized” latent space, where the separation between prototypes of different classes is unclear. When that is the case, generating small sufficient explanations will not be possible, which questions the self-explainable nature of ProtoPNet.

6. Comparing our methodology and tool with AI Act

Our work can be tied with article 86 of the AI Act [42, 43], that grants the right to “obtain from the deployer clear and meaningful explanations of the role of the AI system in the decision-making procedure [...]". As such, assessing the correctness of methodologies providing explanations is an important endeavour. It has been shown in other fields of applications that formal-based approaches can be used to provide safety, according to norms such as the ISO/IEC 24029 in the avionics sector. Furthermore, the use of a prototype architecture has been shown to be human-interpretable [44]. However, since the prototypes are extracted from (a subset of) the training set, any private data included is susceptible to be accessed by the model during inference. This family of models, coupled with our formal explanations, should not be used as of now when concerns about data privacy arise.

7. Conclusion and Future Works

In this work, we have introduced Abductive Latent Explanations (ALE), a novel framework that formalizes explanations for prototype-based networks as rigorous abductive inferences within the latent space. We proposed a solver-free algorithm for generating ALEs that drastically reduces the computation time. Our analysis, enabled by ALEs, reveals that common prototype-based explanations can be misleading, raising concerns about their reliability in high-stakes decisions.

Furthermore, we investigated the relationship between prediction correctness and ALE size, with findings showing that larger ALEs often correlate with incorrect predictions, suggesting ALE size as a potential proxy for model uncertainty.

While our current investigation focused on a specific class of prototypes, a significant direction for future work involves exploring the generalizability of ALEs across the diverse spectrum of prototype definitions and modalities found in the literature.

A primary challenge, and thus a key area for future research, is the considerable size of currently generated ALEs, which can impede human interpretability. Developing methods to compact these formal explanations is crucial. Success in this area would also contribute to more resource-efficient and sustainable XAI.

Enhancing the semantic quality of ALEs hinges on the interpretability of the underlying prototypes. Future work could explore how the formal properties or structures of ALEs might, in turn, inform or guide the learning of more human-aligned and explanatorily effective prototypes, potentially leading to more intrinsically compact ALEs. Exploring other modalities of prototypical parts, such as *concept activation vectors* lying in a different latent space, could result in more interpretable yet formal explanations.

The formal structure of ALEs provides a strong foundation for generating robust and diverse latent-space counterfactuals, offering a promising avenue for future exploration.

Finally, ALEs could potentially identify non-salient or provably irrelevant latent components. Future work could develop techniques to leverage this information for targeted pruning or deactivation of such components, leading to more focused latent representations and, consequently, more concise ALEs.

Overall, this work bridges formal abductive reasoning with prototype-based networks, advancing XAI by offering rigorous yet interpretable explanations. ALE not only addresses limitations in existing prototype explanations but also lays a foundation for future innovations in efficient, human-centric AI interpretability.

Acknowledgments

This publication was made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-De-France Regional Council. This work was supported by the SAIF project, funded by the "France 2030" government investment plan managed by the French National Research Agency, under the reference ANR-23-PEIA-0006.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini and ChatGPT in order to: Grammar and spelling check. Further, the authors used the same tools in order to: Improve writing style.

References

[1] C. Molnar, Interpretable Machine Learning, 3 ed., 2025. URL: https://christophm.github.io/interpretable-ml-book.

[2] D. Smilkov, N. Thorat, B. Kim, F. B. Viégas, M. Wattenberg, Smoothgrad: removing noise by adding noise, ArXiv abs/1706.03825 (2017).

[3] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: D. Precup, Y. W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.

[4] C. L. Choi, A. Duplessis, S. Belongie, Unlearning-based neural interpretations, 2025. URL: https://arxiv.org/abs/2410.08069. arXiv:2410.08069.

[5] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, K.-R. Müller, Layer-wise relevance propagation: An overview, in: Explainable AI, 2019.

[6] A.-K. Dombrowski, M. Alber, C. Anders, M. Ackermann, K.-R. Müller, P. Kessel, Explanations can be manipulated and geometry is to blame, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[7] S. Bordt, M. Finck, E. Raidl, U. von Luxburg, Post-hoc explanations fail to achieve their purpose in adversarial contexts, in: 2022 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’22, ACM, 2022, p. 891–905. URL: http://dx.doi.org/10.1145/3531146.3533153. doi:10.1145/3531146.3533153.

[8] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity Checks for Saliency Maps, in: Advances in Neural Information Processing Systems 32, 2018, p. 11.

[9] A. Hedström, L. Weber, S. Lapuschkin, M. Höhne, A fresh look at sanity checks for saliency maps, 2024. URL: https://arxiv.org/abs/2405.02383. arXiv:2405.02383.

[10] J. Marques-Silva, X. Huang, Explainability is not a game, Communications of the ACM 67 (2024) 66–75. doi:10.1145/3635301.

[11] J. Colin, T. Fel, R. Cadene, T. Serre, What I Cannot Predict, I Do Not Understand: A Human-Centered Evaluation Framework for Explainability Methods, Technical Report, 2023. URL: http://arxiv.org/abs/2112.04417. doi:10.48550/arXiv.2112.04417, zSCC: NoCitationData[s0] arXiv:2112.04417 [cs] type: article.

[12] G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, P. Marquis, On the explanatory power of decision trees, 2021. URL: https://arxiv.org/abs/2108.05266. arXiv:2108.05266.

[13] J. Marques-Silva, A. Ignatiev, Delivering trustworthy ai through formal xai, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 12342–12350.

[14] S. Bassan, G. Katz, Towards formal xai: Formally approximate minimal explanations of neural networks, arXiv preprint arXiv:2210.13915 (2022).

[15] W. Shi, A. Shih, A. Darwiche, A. Choi, On tractable representations of binary neural networks, 2020. URL: https://arxiv.org/abs/2004.02082. arXiv:2004.02082.

[16] L. Wolf, T. Galanti, T. Hazan, A formal approach to explainability, in: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 255–261.

[17] A. De Palma, S. Durand, Z. Chihani, F. Terrier, C. Urban, On using certified training towards empirical robustness, 2024. doi:10.48550/ARXIV.2410.01617.

[18] M. Wu, H. Wu, C. Barrett, Verix: Towards verified explainability of deep neural networks, 2023. arXiv:2212.01051.

[19] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. Van Keulen, C. Seifert, From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai, ACM Computing Surveys 55 (2023) 1–42.

[20] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38.

[21] G. Katz, C. Barrett, D. L. Dill, K. Julian, M. J. Kochenderfer, Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, Springer International Publishing, 2017, pp. 97–117. doi:10.1007/978-3-319-63387-9_5.

[22] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman, Building machines that learn and think like people, 2016. URL: https://arxiv.org/abs/1604.00289. arXiv:1604.00289.

[23] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, C. Rudin, *This* looks like *That*: Deep learning for interpretable image recognition, Proceedings of the 33rd International Conference on Neural Information Processing Systems (2019) 8930–8941.

[24] A. V. Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, ArXiv abs/1907.02584 (2019).

[25] M. Nauta, R. van Bree, C. Seifert, Neural prototype trees for interpretable fine-grained image recognition, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 14928–14938. doi:10.1109/cvpr46437.2021.01469.

[26] D. Rymarczyk, Ł. Struski, M. Górszczak, K. Lewandowska, J. Tabor, B. Zieliński, Interpretable image classification with differentiable prototypes assignment, in: European Conference on Computer Vision, Springer, 2022, pp. 351–368.

[27] F. Willard, L. Moffett, E. Mokel, J. Donnelly, S. Guo, J. Yang, G. Kim, A. J. Barnett, C. Rudin, This looks better than that: Better interpretable models with protopnext, arXiv preprint arXiv:2406.14675 (2024).

[28] M. Sacha, B. Jura, D. Rymarczyk, Ł. Struski, J. Tabor, B. Zieliński, Interpretability benchmark for evaluating spatial misalignment of prototypical parts explanations, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21563–21573.

[29] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, R. Sayres, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), 2018. URL: https://arxiv.org/abs/1711.11279. arXiv:1711.11279.

[30] T. Fel, A. Picard, L. Bethune, T. Boissin, D. Vigouroux, J. Colin, R. Cadène, T. Serre, CRAFT: Concept Recursive Activation FacTorization for Explainability, Technical Report, 2023. URL: http://arxiv.org/abs/2211.10154. doi:10.48550/arXiv.2211.10154, zSCC: 0000057 arXiv:2211.10154 [cs] type: article.

[31] F. De Santis, G. Ciravegna, P. Bich, D. Giordano, T. Cerquitelli, V-cem: Bridging performance and intervenability in concept-based models, arXiv preprint arXiv:2504.03978 (2025).

[32] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, P. Liang, Concept bottleneck models, in: International conference on machine learning, PMLR, 2020, pp. 5338–5348.

[33] M. Espinosa Zarlenga, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, Z. Shams, F. Precioso, S. Melacci, A. Weller, et al., Concept embedding models: Beyond the accuracy-explainability trade-off, Advances in Neural Information Processing Systems 35 (2022) 21400–21413.

[34] A. Helbling, T. H. S. Meral, B. Hoover, P. Yanardag, D. H. Chau, Conceptattention: Diffusion transformers learn highly interpretable features, arXiv preprint arXiv:2502.04320 (2025).

[35] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, J. K. Su, This looks like that: deep learning for interpretable image recognition, Advances in neural information processing systems 32 (2019).

[36] R. Xu-Darme, A. Varasse, A. Grastien, J. Girard, Z. Chihani, CaBRNet, an open-source library for developing and evaluating Case-Based Reasoning Models, in: Joint Proceedings of the xAI-2024 Late-breaking Work, Demos and Doctoral Consortium co-located with the 2nd World Conference on eXplainable Artificial Intelligence (xAI-2024), Valletta, Malta, 2024, pp. 265–272. URL: https://cea.hal.science/cea-04688217.

[37] A. Ignatiev, N. Narodytska, J. Marques-Silva, On validating, repairing and refining heuristic ml explanations, arXiv preprint arXiv:1907.02509 (2019).

[38] A. Ignatiev, N. Narodytska, J. Marques-Silva, Abduction-based explanations for machine learning models, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 1511–1519.

[39] E. Dijkstra, A discipline of programming, Prentice-Hall series in automatic computation, Prentice-Hall, 1976.

[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[41] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200-2011 dataset (2011).

[42] European Commission. Shaping Europe's digital future, AI Act, ???? URL: https://digital-strategy.

ec.europa.eu/en/policies/regulatory-framework-ai, last visit in 2025, July 22.

[43] European Union, EUR-Lex Acess to European Uninion Law. AI Act, ???? URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj, last visit in 2025, July 22.

[44] O. Davoodi, S. Mohammadizadehsamakosh, M. Komeili, On the interpretability of part-prototype based classifiers: a human centric analysis, Scientific Reports 13 (2023) 23088.

A. Proofs

Proof. Assume the explanation $\mathcal{E}$ is verified, meaning Verify($\mathcal{E}$) holds. By Definition 5, this implies that for all classes $k \neq c$, the condition $\psi_{\mathcal{E}}(k, c)$ holds. By Definition 4, this means:

$$\forall k \neq c : \quad h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{a}^*_{\mathcal{E}}(k, c)) \tag{1}$$

where $\mathbf{a}^*_{\mathcal{E}}(k, c) = \text{argmax}_{\mathbf{s}' \in S_{\mathcal{E}}}(h_k(\mathbf{s}') - h_c(\mathbf{s}'))$.

We want to show that $\mathcal{E}$ satisfies Definition 1. This requires showing that for any latent representation $\mathbf{z}$ (corresponding to a similarity vector $\mathbf{s}$) such that its components satisfy the constraints imposed by $\mathcal{E}$ (i.e., $\mathbf{s} \in S_{\mathcal{E}}$), the classification is c. That is, we need to show:

$$\forall \mathbf{s} \in S_{\mathcal{E}} : \quad \kappa_{\mathcal{Z}}(\mathbf{z}) = c$$

Since $\kappa_{\mathcal{Z}}(\mathbf{z}) = \text{argmax}_{k'} h_{k'}(\mathbf{s})$, this is equivalent to showing:

$$\forall \mathbf{s} \in S_{\mathcal{E}}, \quad \forall k \neq c : \quad h_c(\mathbf{s}) \geq h_k(\mathbf{s})$$

Let $\mathbf{s}$ be an arbitrary similarity vector in the constrained space $S_{\mathcal{E}}$. Let $k \neq c$ be an arbitrary alternative class.

By the definition of $\mathbf{a}^*_{\mathcal{E}}(k, c)$ as the maximizer of $(h_k - h_c)$ within $S_{\mathcal{E}}$, we know that for our chosen $\mathbf{s} \in S_{\mathcal{E}}$:

$$h_k(\mathbf{a}^*_{\mathcal{E}}(k, c)) - h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{s}) - h_c(\mathbf{s})$$

From our initial assumption that $\mathcal{E}$ is verified, we know from (1) that:

$$h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{a}^*_{\mathcal{E}}(k, c))$$

This can be rewritten as:

$$0 \geq h_k(\mathbf{a}^*_{\mathcal{E}}(k, c)) - h_c(\mathbf{a}^*_{\mathcal{E}}(k, c))$$

Combining these two inequalities, we have:

$$0 \geq h_k(\mathbf{a}^*_{\mathcal{E}}(k, c)) - h_c(\mathbf{a}^*_{\mathcal{E}}(k, c)) \geq h_k(\mathbf{s}) - h_c(\mathbf{s})$$

This directly implies:

$$0 \geq h_k(\mathbf{s}) - h_c(\mathbf{s})$$

Which rearranges to:

$$h_c(\mathbf{s}) \geq h_k(\mathbf{s})$$

Since $\mathbf{s} \in S_{\mathcal{E}}$ was arbitrary and $k \neq c$ was arbitrary, we have shown that $h_c(\mathbf{s}) \geq h_k(\mathbf{s})$ for all $\mathbf{s} \in S_{\mathcal{E}}$ and for all $k \neq c$. Therefore, for any $\mathbf{s} \in S_{\mathcal{E}}$, $c = \text{argmax}_{k'} h_{k'}(\mathbf{s})$, which means $\kappa_{\mathcal{Z}}(\mathbf{z}) = c$.

This fulfills the condition required by Definition 1. Thus, if $\mathcal{E}$ is verified via Total Prediction Domination within $S_{\mathcal{E}}$, it is a sufficient abductive explanation. □