Robust Hierarchical Symbolic Explanations in Hyperbolic Space for Image Classification

Ainkaran Santhirasekaram[1*], Avinash Kori[1*], Mathias Winkler[2], Andrea Rockall[2],
Francesca Toni[1], and Ben Glocker[1]
[1]Department of Computing, Imperial College London
[2]Department of Surgery and Cancer, Imperial College London
{a.santhirasekaram19, a.kori21, m.winkler, a.rockall, f.toni, b.glocker}@ic.ac.uk

Abstract

Explanations for black-box models help us to understand model decisions as well as provide information on model biases and inconsistencies. Most of the current post-hoc explainability techniques provide a single level of explanation, often in terms of feature importance scores or feature attention maps in the input space. The explanations provided by these methods are also sensitive to perturbations in the input space. Our focus is on explaining deep discriminative models for images at multiple levels of abstraction, from fine-grained to fully abstract explanations. We use the natural properties of hyperbolic geometry to more efficiently model a hierarchical relationship of symbolic features with decreased distortion to generate robust hierarchical explanations. Specifically, we distill the underpinning knowledge in an image classifier by quantising the continuous latent space to form hyperbolic symbols and learn the relations between these symbols in a hierarchical manner to induce a knowledge tree. We traverse the tree to extract hierarchical explanations in terms of chains of symbols and their corresponding visual semantics. Code is available at https://github.com/AinkaranSanthi/HyperbolicReasoning

1. Introduction

Explainable AI (XAI) aims to improve the transparency and trustworthiness of models [9, 20]; XAI can help with identifying biases, which is important for the safe and fair use of prediction models [18]. XAI approaches can be broadly categorized into ante-hoc and post-hoc methods [22]. Ante-hoc explainability focuses on developing inherently transparent models [21, 35]. Post-hoc explanations are the most commonly explored approaches, including explanations via feature-attribution [24, 30, 34], saliency maps [5, 32, 33] counterfactuals [13] or concept extraction [11, 12, 19]. Feature attribution methods [24, 30] focus on assigning importance weighting to features in input space, indicating their contribution towards the classifier's decision. Saliency-based methods [33] generate attention maps in input space indicating the image regions responsible for deriving the classifier's decision. These methods by design provide 'single level' explanations as they do not consider any form of reasoning via feature interaction as perceived by humans [2, 3]. Concept-based explanations [11, 12, 19] go beyond feature-attribution and saliency-based methods by constructing higher-level concepts indicating their influence on the classifier's decision. Hierarchical concept-based reasoning is the most commonly posited learning principle in systems neuroscience [3, 26, 40, 41]. Inspired by this form of reasoning, we provide the first work to develop hierarchical symbolic explanations for deep discriminative models trained on imaging data. We propose to distil the knowledge in an image classifier into a knowledge tree where nodes represents symbols. One can then transverse the tree to derive hierarchical symbolic explanations in the form of a chain of increasingly abstract symbols leading to the class symbol which we denote as a chain rule. In order to efficiently embed the symbols in the knowledge tree without distortion and overlap, we propose to learn symbols as discrete hyperbolic representations.

Our main contributions in this work include:

- **Symbol formation:** A method to discretise the continuous latent space of a given classifier model into a hierarchy of discrete vectors denoted as symbols.
- **Hierarchical Symbolic Relationships:** An effective way of learning hierarchical symbol relations using binary weight layers to form a knowledge tree. This is used to extract a set of robust local image-level and global class-level chain rules as explanations for image classification. .
- **Hyperbolic Symbols:** This is the first work to con-

*Equal contribution

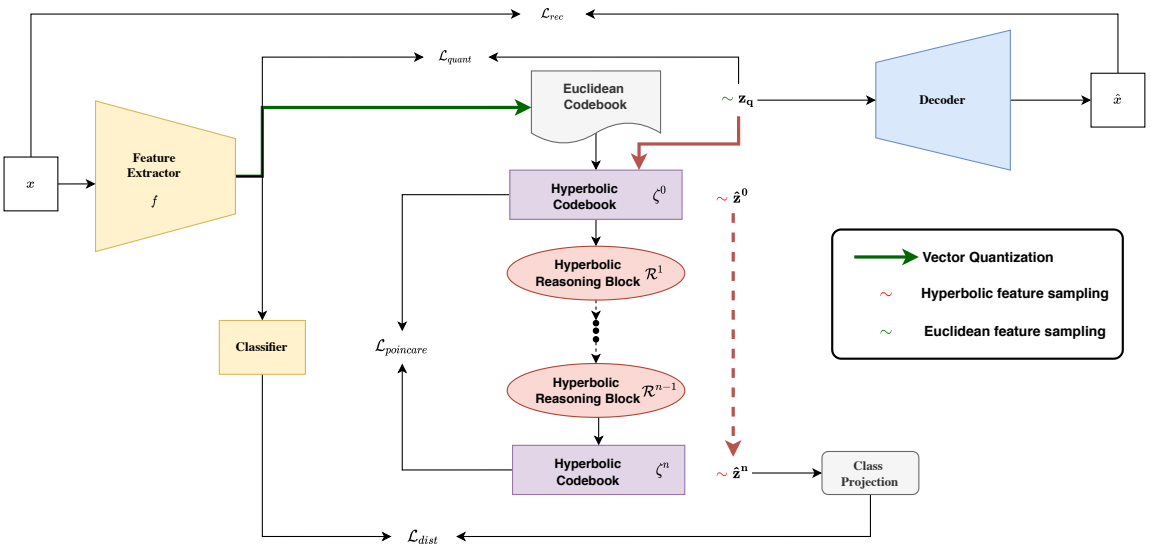

Figure 1. Overview of the proposed framework, in which the feature extractor and classifier describe the trained blocks of the given model. The Euclidean codebook forms a discrete representation of the continuous latent space from the feature extractor, followed by hyperbolic codebooks and reasoning blocks to obtain a knowledge tree, which is later used in extracting explanations. The decoder block is independently trained to obtain visual semantics for the extracted hierarchical symbolic rules.

sider the natural curvature in which to embed symbols to more efficiently model their interactions. We exploit the natural structure of hyperbolic geometry for modelling hierarchical relationships between symbols.

To facilitate future work, all code will be released upon publication.

2. Related Work

Symbolic Reasoning: Concept learning for image classification has been used as an ante-hoc method towards building more interpretable image classifiers. For example, [1] develops an encoder to learn a set of discrete concepts and a parallel neural network to learn the relevance of the concepts to making a classification decision. Instead, we want to learn links between concepts. This is related to a popular wave of AI termed neuro-symbolic learning [15, 25, 39, 43], where a data-driven deep learning method is used to learn sub-symbolic representations to denote concepts while exploiting symbolic methods to capture reasoning. Inductive logic programming (ILP) is a framework often used for symbolic reasoning with learnt relational theories, including using heuristics and physical properties to understand images [27]. The solution we propose is closely related to a family of methods called knowledge graph embeddings [36, 37, 44] which models relations between discrete entities or symbols for knowledge graph reasoning. NeuralLP [42] can learn a relational path from the subject to the object which, in ILP terms, can be formulated as a chain-like first-order rule. This approach is devised for answering knowledge base queries but will form the basis in our work for deriving chain rules from our generated knowledge trees for the purpose of explanation.

Hyperbolic Embeddings: A natural objective for embedding symbolic data in graphs is for the distances between symbols, defined by the space which they reside in, to correlate with their semantic similarity. Yet, to model increasingly complex relations between symbols, one is bounded by the dimensionality of Euclidean embeddings [28]. This is because the number of nodes generally grows exponentially as the graph distance from the centre node increases, while Euclidean space grows polynomially. This leads to distorted embeddings and information loss [31].

Hyperbolic geometry is a form of non-Euclidean geometry with a constant negative Gaussian curvature whose space grows exponentially. One can even informally describe hyperbolic space as the continuous version of trees, making it naturally equipped to deal with tree-like structures. This property has therefore been exploited in the literature for embedding hierarchical data in hyperbolic space without distortion [29] and provides the reasoning for embedding our knowledge tree in hyperbolic space.

Hyperbolic neural networks were proposed to perform all the operations of a neural network in hyperbolic space [10]. This was exploited in the development of hyperbolic graph convolutional neural networks (HGCNN) [4,23] which forms the structure in our method for reasoning in hyperbolic space.

3. Preliminaries

3.1. Hyperbolic Geometry

Here we introduce some important geometric concepts underpinning our method. A d dimensional **Manifold** $\mathcal{M}$ is a topological space embedded in $\mathbb{R}^{d+1}$ that can be locally approximated in Euclidean space $\mathbb{R}^d$, that is each point on the manifold consists of a neighbourhood homeomorphic to an open subset in $\mathbb{R}^n$. The **Tangent Space** $\mathcal{T}_x\mathcal{M}$ of a point x on a differentiable manifold is a vector space that comprises of tangent vectors to all feasible paths on the manifold that pass through x. The **Riemannian Metric** defines the set of inner products $g_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \rightarrow \mathbb{R}$ of every point x on $\mathcal{M}$. **Parallel transport** $P_{x\rightarrow y}$ describes the translation of a vector field V along a differentiable manifold to a new vector field V' such that the covariant derivative always stays at 0.

We now introduce the **Hyperboloid** and **Poincare** models of hyperbolic space ($\mathcal{H}$) with radius K and equipped with constant negative curvature $-1/K, (K > 0)$. In our work, we work only with manifolds of d dimensions with unit radius and hence a fixed negative curvature of -1 which we formally denote as $\mathbb{H}^{d,1}$ and $\mathbb{B}^{d,1}$ for the hyperboloid and Poincare models respectively. We focus on the two sheet unit hyperboloid model equipped with a Riemmanian metric; $g_x^{\mathbb{H},1}$ given by the Minkowski metric tensor $\langle ., . \rangle_S$ whereby $\langle x, x \rangle_S = -1$ and $x \in \mathbb{R}^{d+1}$ [4]. We can use the metric tensor to calculate the geodesic distance defined as the shortest distance between u and v on the hyperboloid; $(u, v) \in \mathbb{H}^{d,1}$. Geodesic distance in the hyperboloid is calculated as follows: $d^{\mathbb{H},1}(u, v) = \text{arccosh}(g_x^{\mathbb{H},1}\langle u, v \rangle)$ [4].

The open Poincare unit ball is formally defined as: $\mathbb{B}^{d,1} = \{x \in \mathbb{R}^d | \|x\| < 1\}$. $\mathbb{B}^{d,1}$ is determined by the following Riemmanian metric tensor: $g_x^{\mathbb{B},1} = \left(\frac{2}{1-\|x\|^2}\right)^2$. Therefore, the derived geodesic distance equation for two points, $(u, v) \in \mathbb{B}^{d,1}$, is given by:

$$d^{\mathbb{B},1}(u, v) = 1 + 2\Big(\frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\Big) \tag{1}$$

We define a mapping between the hyperboloid and Euclidean space (tangential plane) which is useful in our work. Given two points $y \in \mathcal{T}_x\mathbb{H}^{d,1}$ and $v \in \mathbb{H}^{d,1}$, exponential and logarithmic mappings are denoted as $\exp_x^{\mathbb{H},1}(y)$: $\mathcal{T}_x\mathbb{H}^{d,1} \rightarrow \mathbb{H}^{d,1}$ and $\log_x^{\mathbb{H},1}(v)$: $\mathbb{H}^{d,1} \rightarrow \mathcal{T}_x\mathbb{H}^{d,1}$ respectively. In this work, we perform mappings with the tangential space at the origin o. In this case, the mapping between $y \in \mathcal{T}_o\mathbb{H}^{d,1}$ and $v \in \mathbb{H}^{d,1}$ is calculated as follows:

$$\begin{aligned} \exp_o^{\mathbb{H},1}(y) &= \Big(\cosh(\|y_{1:d}\|_2), \sinh(\|y_{1:d}\|_2)\frac{y_{1:d}}{\|y_{1:d}\|_2}\Big) \\ \log_o^{\mathbb{H},1}(v) &= \Big(0, \text{arccosh}(v_0)\frac{v_{1:d}}{\|v_{1:d}\|_2}\Big) \end{aligned} \tag{2}$$

In the case of the Poincare model, we must also define the equations for mapping between $y \in \mathcal{T}_o\mathbb{B}^{d,1}$ and $v \in \mathbb{B}^{d,1}$ as:

$$\begin{aligned} \exp_o^{\mathbb{B},1}(y) &= \Big(\tanh(\|y\|_2)\frac{y}{\|y\|_2}\Big) \\ \log_o^{\mathbb{B},1}(v) &= \Big(0, \text{arctanh}(\|v\|_2)\frac{v}{\|v\|_2}\Big) \end{aligned} \tag{3}$$

We apply projections as described in [4] to constrain points to the manifolds during optimisation.

A useful diffeomorphic mapping ψ between a point on the hyperboloid $u \in \mathbb{H}^{d,1}$ and the Poincare unit ball $v \in \mathbb{B}^{d,1}$ is given by:

$$\begin{aligned} \psi_{\mathbb{H}^{d,1}\rightarrow\mathbb{B}^{d,1}}(u_0 \dots u_d) &= \frac{u_1 \dots u_d}{u_0 + 1} \\ \psi_{\mathbb{B}^{d,1}\rightarrow\mathbb{H}^{d,1}}(v_{1d}) &= \frac{(1 + ||v||_2^2, 2v_1 \dots 2v_d)}{1 - ||v||_2^2} \end{aligned} \tag{4}$$

In our work, we apply feature transformations in hyperbolic space, using hyperbolic linear layers [4, 10]. The operations of Mobius addition $\oplus$ and Mobius scalar multiplication $\otimes$ in hyperbolic space can be shown to be analogous to the Euclidean vector space operations of scalar multiplication and addition. [10] proves that Mobius scalar multiplication is equivalent to applying a logarithmic mapping of a point $v \in \mathcal{H}^{d,1}$ to the tangential space at o and multiplying by the scalar r before mapping the scaled point back to hyperbolic space as shown in Eq. (5) below:

$$r \otimes^1 v = \exp_o^{\mathcal{H},1}(r \ \log_o^{\mathcal{H},1}(v)) \tag{5}$$

In a hyperbolic linear layer, we also add a bias term b. [10] derives a simple equivalent solution to Mobius addition ($v \oplus b$) shown in Eq. (6) below. One firstly defines b on $\mathcal{T}_o\mathcal{H}^{1,d}$ which is parallel transported to the tangential space $\mathcal{T}_v\mathcal{H}^{1,d}$ before mapping back to hyperbolic space. (Please refer to the supplementary material for the equations of parallel transport and the general form for $\exp_x^K(y)$, $\log_x^K(v)$.)

$$v \oplus^1 b = \exp_v^{\mathcal{H},1}(P_{o\rightarrow v}^1(b)) \tag{6}$$

We let $W \in \mathbb{R}^{d' \times d}$ and $B \in \mathbb{R}^d$ to define the parameters in a hyperbolic linear feature transformation $h(x)$ for both $\mathbb{H}^{d,1}$ and $\mathbb{B}^{d,1}$ by combining Eq. (5) and Eq. (6) to form:

$$h(x) = (W \otimes^1 x) \oplus^1 B \tag{7}$$

3.2. Knowledge Tree

In this work, we only consider classifiers of the form $\mathcal{C} = f \circ g$, where a feature extractor $f : \mathcal{X} \rightarrow \mathcal{E}$ maps input images to the latent space $\mathcal{E}$ and the feature classifier $g : \mathcal{E} \rightarrow \mathcal{Y}$ maps the latent space to class labels. We distill

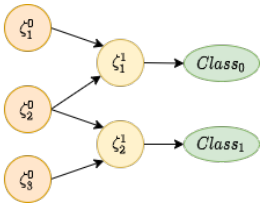

Figure 2. Example knowledge tree. Here, symbols ζ_1^0 and ζ_2^0 are being abstracted to form ζ_1^1 which forms class0. Symbols ζ_2^0 and ζ_3^0 are being abstracted to form ζ_2^1 which forms class1.

the high abstraction knowledge of the continuous feature classifier (g) into a hierarchy of related symbols to form a knowledge tree whereby a symbol is a discrete vector. We assume sufficient training data is available to develop a closed-world knowledge tree.

We denote ζ_j^i to be the j^{th} symbol in the i^{th} level of the knowledge tree. The i^{th} level in the tree is a dictionary of hyperbolic symbols represented as a hyperbolic codebook $\zeta^i \in \mathcal{H}^{M_i \times d'}$; these levels provide different levels of abstraction as shown in Fig. 1. The total number of codebooks n and embedding dimensionalities d are hyper-parameters that are selected based on the use case. $M_0, M_1, \ldots, M_n$ are hyper-parameters corresponding to the total number of symbols in codebooks $\zeta^0, \zeta^1 \ldots, \zeta^n$ respectively. n should be greater than 2 to develop a hierarchy and the greater n, the more levels of abstraction and the 'deeper' the explanations. However, we do not want n to be inappropriately large to prevent the exponential increase in chain rules and reduce interpretability. We strive for an Occam's Razor approach in terms of the numbers of symbols to generate explanations and carry out ablations to find the smallest number of symbols, M_n required in each codebook before knowledge distillation accuracy from a pre-trained classifier diminishes.

Formally, a knowledge tree is developed by learning the function $\mathcal{K}$ which collapses the Euclidean continuous latent space $\mathcal{E}$ into ideally $\lfloor \log_2 N + 1 \rfloor$ symbols in ζ^n as this represents the minimum number of positive symbols to encode N classes. We can decompose $\mathcal{K}$ such that $\mathcal{K} = \mathcal{R} \circ \mathcal{VQ}$, where $\mathcal{VQ}$ denotes discretisation ($\mathcal{VQ}$) of $\mathcal{E}$ using vector quantisation [38]. $\mathcal{R}$ expresses the hyperbolic symbol abstraction module to produce increasingly abstract hyperbolic symbols. $\mathcal{R}$ can be decomposed into $\mathcal{R}^n \circ \mathcal{R}^{n-1} \circ \ldots \circ \mathcal{R}^1$ with the output of each $\mathcal{R}^l$ producing hyperbolic symbols of different level of abstraction in the form of ζ^i (see again Fig. 1). We train $\mathcal{K}$ by sampling z from $\mathcal{E}$ and sequentially mapping and discretising z to increasing levels of abstraction in the knowledge tree to produce z_q^i with the final level of abstraction used to classify z.

As a simple illustration, in Fig. 2, we have two levels in the knowledge tree corresponding to codebooks ζ^0 and ζ^1 each with 3 and 2 symbols, which are getting abstracted to form $Class_0$ and $Class_1$. For example, it can be seen that symbols $\zeta_2^0$and ζ_3^0 are getting merged to form an abstract concept ζ_2^1 which is further getting merged with ζ_3^1 to form higher level abstraction of concept $Class_1$. This forms a class-level induced tree which indicates a subtree corresponding to a specific class.

3.3. Symbolic Relations

We assume the task of single label image classification only requires reasoning in a hierarchical manner. Therefore, in our generated knowledge tree, paths which link symbols at the bottom of the tree to the class label at the top of the tree, represent chain rules which implies a class is true for a given image. Specifically, the first level of the tree indicates whether a symbol exists (*ex* in short) in an image, x. Thereafter, the symbols are being merged to form a more abstract symbol in the next level and therefore an arrow/relation between symbol ζ_i^{n-1} and ζ_i^{n-2} indicates that ζ_j^{n-2} is part of (*pf* in short) symbols ζ_i^{n-1}; $pf(\zeta_i^{n-1}, \zeta_i^{n-2})$. For example, a cat's ear is part of the cat's face. $pf(y, \zeta_i^n)$ indicates symbol ζ_i^n is part of class y. We formally aim to learn the relations (*ex*, *pf*) in the form of the chain rule shown in Eq. (8) which implies that image x belongs to class y if $pf(y, \zeta_i^n),\ pf(\zeta_i^{n-1}, \zeta_i^{n-2}) \ldots pf(\zeta_i^1, \zeta_i^0),\ ex(x, \zeta_i^0)$ are all true. The learnt set of chain rules are used to construct a knowledge tree as shown in Fig. 2. This is achieved in our works using a binary neural network [14].

$$class(y, x) \leftarrow pf(y, \zeta_i^n) \ \wedge \ pf(\zeta_i^{n-1}, \zeta_i^{n-2}) \ \wedge \ldots \\ pf(\zeta_i^1, \zeta_i^0) \ \wedge \ ex(x, \zeta_i^0) \tag{8}$$

4. Methods

4.1. Symbol formation

The first step in our framework, symbol formation, is performed by learning discrete symbols in the form of d' dimensional vectors using vector quantisation ($\mathcal{VQ}$) to form a fixed sized Euclidean codebook $\mathbb{C} \in \mathbb{R}^{M_0 \times d'}$ (Fig. 1). We do not quantise in hyperbolic space as we found this significantly less stable.

Given $(\hat{z}, z) \in \mathbb{R}^{K \times d'}$ and $k \in K$, we define a deterministic process which maps each embedding vector $z_k \in z$ to the nearest Euclidean codebook vector to form $\hat{z}_k \in \hat{z}$ shown below:

$$\hat{z}_k = \underset{j}{\operatorname{argmin}} \|z_k - \mathbb{C}_j\|_2, \forall \quad k \in K \tag{9}$$

Eq. (9) defines a sampling process which is non-differentiable but in order to update/learn the symbols which form $\mathbb{C}$ based on this sampling method, we apply straight through gradient approximation. This then allows

our discrete surrogate model to be trained end to end with the following Quantisation loss [38]: $\mathcal{L}_{quant} = \|sg(z) - \hat{z}\|_2 + \beta\|z - sg(\hat{z})\|_2$. We apply stop gradients (sg) to constrain updates to the appropriate operands [38]. This process of sampling the Euclidean codebook is equivalent to learning the *exists* predicate in the first step of the chain rule. Next, we reduce the dimensionality using a linear projection layer to the desired embedding dimensionality d in the hyperbolic codebooks ζ^i before applying an exponential mapping with Eq. (3) to Poincare space. This leads to the first hyperbolic codebook ($\zeta^0 \in \mathbb{B}^{M_0 \times d,1}$). We choose Poincare space in this work due to the enhanced visual interpretability of 2D embeddings on the Poincare disc [29].

4.2. Hyperbolic Symbol Abstraction

The goal of the hyperbolic symbol abstraction module ($\mathcal{R}^l$) is produce increasingly abstract hyperbolic symbols at each level of the tree by merging symbols with edges (*pf* relation) from the previous level, i.e. the hyperbolic symbols for cat eyes and nose are merged together to form the symbol, cat face in the next level. ($\mathcal{R}^l$) is similar to a single layer HGCNN [4] but here we learn the edges of the graph between ζ^i and ζ^{i+1} using a binary function ($1 = edge, 0 = no\ edge$). This constructs a knowledge graph structure equivalent to a tree which is used for reasoning about Euclidean external representations of the visual world. Similar to [4], the first stage of hyperbolic symbol abstraction is a hyperbolic feature transformation shown in Eq. (7), which is performed in the unit hyperboloid where we found training to be more stable compared to within the Poincare unit disc/ball. Therefore, we map a codebook ζ^i from Poincare space to the hyperboloid using Eq. (4) before applying a hyperboloid linear layer ($h(x)^{\mathbb{H},1}$) to each codebook vector using Eq. (7). This is followed by a logarithmic mapping (Eq. (2)) to the tangent space ($\mathcal{T}_o\mathbb{H}^{1,d}$). The second stage of hyperbolic abstraction is the aggregation/merging of symbols in $\mathcal{T}_o\mathbb{H}^{1,d}$ as proposed in [4] to form the next codebook ζ^{i+1}. This is achieved by first learning the edges or equivalently the relations/*partOf* predicate with a binary layer [14] between symbols in consecutive codebooks with weights: $w_l \in \mathbb{R}^{M_i \times M_{i+1}} \in \{0, 1\}$.

We learn a weighted aggregation denoted a_l of the symbols with edges from ζ^i to ζ^{i+1}, before mapping back to Poincare space (Eq. (3)). A single pass through $\mathcal{R}^l$ is summarised in Eq. (10) below.

$$\zeta^{i+1} = \exp_o^{\mathbb{B},1}(a_l^\intercal \ \log_o^{\mathbb{H},1}(h^{\mathbb{H},1}(\psi_{\mathbb{B}^{d,1} \to \mathbb{H}^{d,1}}(\zeta^i)))) \quad (10)$$

In order for the hyperbolic symbol abstraction module to update its weights to form an accurate knowledge tree, it needs to map each feature in $\hat{z}^0$ to the correct class. Firstly, we apply an exponential mapping of $\hat{z}^0$ to the Poincare unit ball defined in Eq. (3) in order to map every feature in $\hat{z}^0$ to the nearest codebook vector by Poincare distance in ζ^0 using Eq. (1) to then form $\hat{z}^1$. This process is repeated sequentially for every hyperbolic codebook until the last codebook where z_q^n is formed by sampling ζ^n. The abstraction process is completed by mapping z_q^n into Euclidean space and then applying a linear class projection layer to map to the class prediction. As this process is done for each of the k features in $\hat{z}^1$, we have k chain rules. However, there multiple chain rules which are repeated and therefore are removed to form a unique set of chain rules for each image.

The process of sequentially sampling each hyperbolic codebook by Poincare distance is equivalent to learning to extract the best chain rules represented as a set of discrete vectors transversing the hyperbolic knowledge tree in order to classify the image. The notion of sampling rules based on distance also allows to rank the best rules as well as ascertain uncertainty over the rules.

The knowledge distillation loss is defined as the cross-entropy loss between the classifiers prediction (y) and the hyperbolic discrete surrogate model's prediction ($\hat{y}$): $\mathcal{L}_{dist}(\hat{y}, y)$. We determine that the Poincare distances between codebooks correspond to graph distances in the tree. Therefore, a Poincare codebook loss $\mathcal{L}_{Poincare}$ is calculated such that symbols with an edge are closer together and those without an edge are pushed apart in hyperbolic space. First, let u to be any symbol in the set of all ζ^i while v and v' are defined as symbols with and without an edge with u respectively; then two sets $\mathcal{P}$ and $\mathcal{W}$ are created such that: $u, v \in \mathcal{P}$ and $u, v' \in \mathcal{W}$. Given this, we can calculate the Poincare codebook loss shown in Eq. (11).

$$\mathcal{L}_{Poincare} = \frac{\sum_{u,v \in \mathcal{P}} e^{d^{\mathbb{B},1}(u,v)}}{\sum_{u,v' \in \mathcal{W}} e^{d^{\mathbb{B},1}(u,v')}}. \quad (11)$$

We now define our total training loss as:
$\mathcal{L}_{Total} = \mathcal{L}_{dist} + \mathcal{L}_{quant} + \mathcal{L}_{Poincare}$.

Continuous weights in our framework are updated with Adam optimisation while the binary weights in $\mathcal{R}^l$ are updated using the *Bop* algorithm proposed by [14]. In our case, we map our weights to {0, 1} rather than {-1, 1}. Please refer to supplementary material for further training details of the binary weights.

4.3. Explanations and visual semantics

We derive a unique set of chain rules for each class to form a class tree. An explanation for the classification of an image is derived in the form a unique set chain rules sampled from the knowledge tree and the visual semantics for symbols. These chain rules can be combined to form an image level tree. For example, in Fig. 2, the subtree for $Class_0$ consists of the chain rules: $class(y, x) \leftarrow pf(y, \zeta_1^1) \ \wedge \ pf(\zeta_1^1, \zeta_1^0) \ \wedge \ ex(x, \zeta_1^0)$ and $class(y, x) \leftarrow pf(y, \zeta_1^1) \wedge pf(\zeta_1^1\ \zeta_2^0) \wedge ex(x, \zeta_2^0)$. An image-level tree is input dependent, and if correctly classified forms a subtree

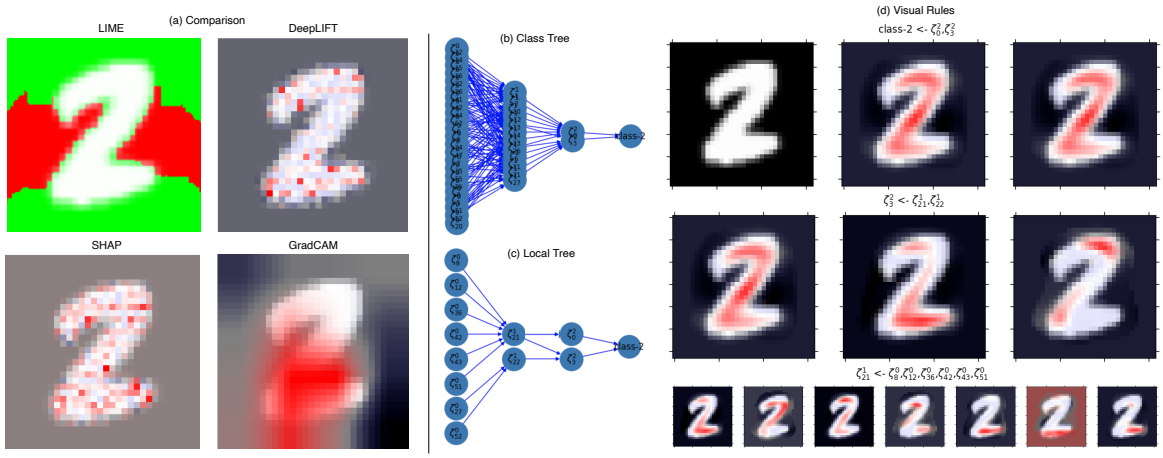

Figure 3. Explanations obtained using the proposed framework for a MNIST classifier. (a) Shows the 4 post-hoc XAI methods we compared with. (b) Demonstrates the obtained class-level tree, which is a complete set of FOL chain rules responsible for 'class 2', (c) indicates an image-level tree formed from the sampled chain rules responsible for making a decision for a given image, and (d) demonstrates a subset of the visual $PartOf$ relations obtained from a image-level tree.

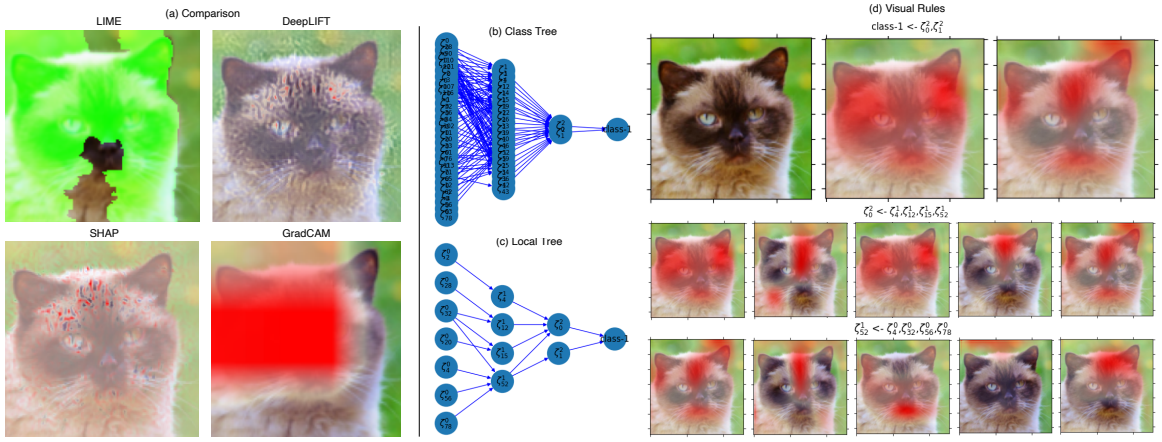

Figure 4. Explanations obtained for an AFHQ classifier deciding upon the 'class cat'. The subset of visual $PartOf$ relation is shown.

of the class-level tree for the ground truth class or in other words a subset of the unique chain rules for the class label.

The extraction of visual semantic for symbols, first requires us to train a decoder $\mathbb{D}$ to reconstruct images (x) in Euclidean space as perceived by the classifier, with a reconstruction loss defined as: $\mathcal{L}_{recon} = ||\mathbb{D}(z_q) - x||_2^2$. During training, we make sure that the gradients from the decoder block do not affect the weights of the discrete surrogate model, to maintain faithfulness of the discretisation process. We visualise the effect of a symbol, $\hat{z}^i \in \zeta^i$ in the reconstructions by finding the symbols in the first abstracted features layer, $\hat{z}^0 \in \zeta^0$ connected via edges to the interested sampled symbol. We then reconstruct the image with these symbolic features set to 0 to visualise the semantic corresponding to the symbol, $\hat{z}^i \in \zeta^i$.

5. Experiments

We use our framework to explain a model pre-trained on the MNIST dataset [8] achieving 98% accuracy as a proof of concept study. In order to indicate the scalability and generalisability of our proposed method we explain models pre-trained on AFHQ [6], STL10 [7] and the MIMIC chest x-ray dataset [16, 17], which achieved 98%, 97% and 81% accuracy respectively. We choose a 3 level hierarchy. Please refer to the supplementary material for details on the classifier model details and codebook ablations.

5.1. Qualitative Experiments

We compare the explanations obtained from our framework against standard post-hoc explainability frameworks: LIME [30], SHAP [24], deepLIFT [34], and gradCAM

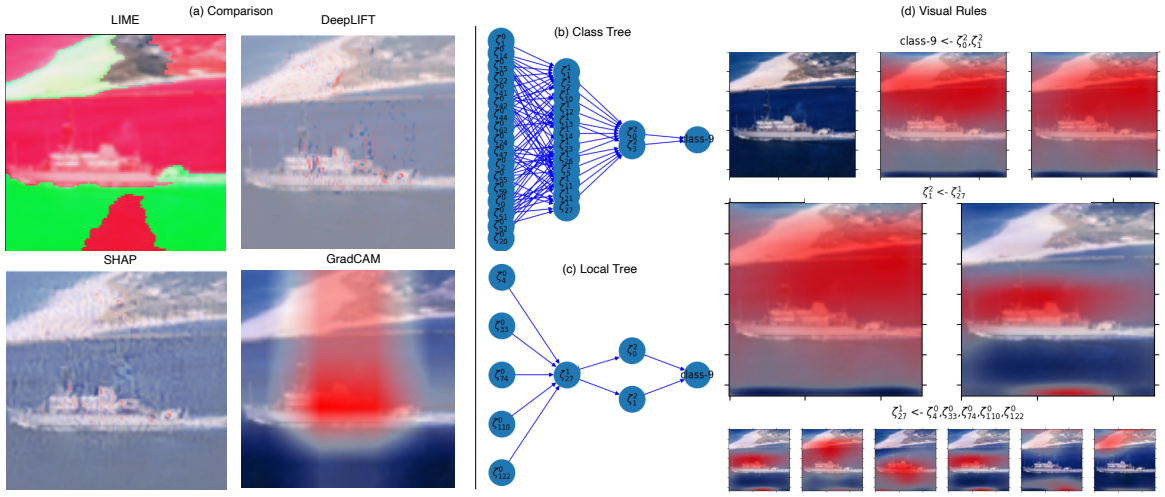

Figure 5. Explanations obtained using the proposed method for a STL10 classifier deciding upon 'class 9 (Boat)'.

[33]. We note the far richer and more expressive explanations by our method shown on the right of Fig. 3, 4 and 5 where we show hierarchical visual explanations.

Fig. 3 demonstrates our explanations on a pre-trained classifier for MNIST. Fig. 3(b) shows the class-level *global-tree* representing all possible chain rules for a particular class. To explain a given image we construct a image-level *local-tree* indicating the sampled chain rules which is shown in Fig. 3(c). Fig. 3(d) provides a visual hierarchical description of an image-level tree. In this example, the visual semantics for the symbolic relations obtained for a given image x corresponding to $pf(Class2,\ \zeta_0^2)$ and $pf(Class2,\ \zeta_3^2)$ is shown in the first row of Fig. 3(d). The second row of Fig. 3(d) visualises the obtained symbolic relations corresponding to $pf(\zeta_3^2,\ \zeta_{21}^1)$ and $pf(\zeta_3^2,\ \zeta_{22}^1)$. The last row visualises sampled symbols from ζ^0 which are part of ζ_1^{21}. As we move down the level of the hierarchy or, in other terms, as we move closer to the boundary of Poincare space, visually the symbols start to move from a complete digit heatmap to a more focused region in a digit, demonstrating a visual hierarchical explanation.

Similar explanations and visual semantic behaviour can be observed for models trained on the AFHQ and STL10 datasets, as shown in Fig. 4 and Fig. 5 respectively, where we again observe symbols getting localized as we go deeper into the hierarchy. We note most of the existing explanation methods shown on the left of Fig. 4 and 5 focus on either pixel importance or gradient-based attention to provide only single level explanations. These explanations do not yield any form of reasoning between the features and the class.

Our method goes beyond feature attribution by allowing the user to decide on the level of abstraction (length of chain rule) upon which to provide the symbolic and corresponding visual semantic relationships. Please see the supplementary material for more examples.

5.2. Robustness Experiments

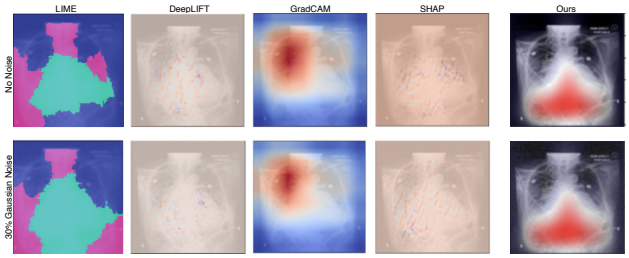

Figure 6. Robustness comparison of considered methods with our proposed method. In our method we show the visual semantic of a single symbol sampled for the last codebook; ζ_0^2. The top shows the heatmaps with no noise added. The second row shows heatmaps with 30% Gaussian noise added

We evaluate the robustness of our method compared to LIME [30], DeepSHAP [24], deepLIFT [34], and grad-CAM [33] by measuring the change in explanations under noise perturbations in the input space. Specifically, we measure the average variance in the heatmaps (visual semantics) generated by our method at the last level of abstraction compared to popular post-hoc methods under 1%, 5%, 10% and 30% Gaussian, Salt and Pepper (s&p) and Poisson noise. We show in Tab. 1 that there is significantly less variance in the explanations generated by our method with the addition of Gaussian noise, highlighting the robustness of our explanations. A visual example to demonstrate our finding is shown in Fig. 6 where under Gaussian noise addition the heatmaps generated by our method show no change compared to the other XAI methods. We note similar results for s&p and Poisson noise (the results tables and examples for

Poisson and s&p noise are in the supplementary material).

	MNIST	STL10	MIMIC	AFHQ
LIME [30]	0.236	1.341	0.785	1.109
SHAP [24]	0.923	1.514	1.143	1.381
deepLIFT [34]	0.535	0.483	0.253	0.585
gradCAM [33]	1.477	1.664	1.966	1.644
Ours	$\mathbf{1e^{-6}}$	$\mathbf{2e^{-5}}$	$\mathbf{4e^{-5}}$	$\mathbf{1e^{-5}}$

Table 1. Average variance in the heatmaps generated by various post-hoc explainability methods under Gaussian addition.

5.3. Hyperbolic vs Euclidean Experiments

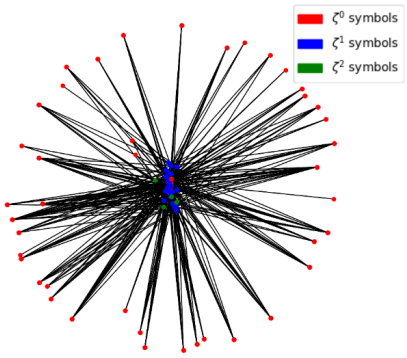

Figure 7. 2D Poincare embedding of symbols obtained for MNIST. Red, blue, and green nodes indicate symbols from $\zeta^0, \zeta^1, \zeta^2$ respectively.

We hypothesised that hyperbolic embeddings will better embed a knowledge tree without distortion and hence allow to reduce the dimensionality d of ζ^i such that knowledge distillation would not be affected. We support this hypothesis by achieving better knowledge distillation accuracy with Poincare embeddings highlighted in Tab. 2. We note an increasingly wider margin in performance between Poincare and Euclidean embeddings as we reduce the dimensionality of ζ^i down to 2 in all 4 datasets. Fig. 7 shows the 2 dimensional embeddings on the Poincare disk for the MNIST dataset maintaining a robust hierarchy. Furthermore, we are also simultaneously showing in Tab. 2 that hyperbolic symbolic abstraction via a higher knowledge distillation accuracy leads to a more accurate explanation of the classifier.

Our hypothesis also implies, that reasoning hierarchically in hyperbolic space should lead to less overcrowding of concepts at the first level of reasoning. One can show this by measuring the overlap using the dice score between all the visual semantics in the first abstraction layer corresponding to all symbols sampled from ζ^0 for each image. Tab. 3 demonstrate less overlap (lower dice score) between concepts in hyperbolic space which becomes more apparent as one reduces the dimensionality of embeddings compared to Euclidean space.

Emb. dim → **Dataset ↓**	**Poincare**			**Euclidean**		
	2	**4**	**16**	**2**	**4**	**16**
MNIST	**0.90**	**0.96**	**0.99**	0.81	0.92	0.95
AFHQ	**0.90**	**0.95**	**0.98**	0.80	0.90	0.97
STL10	**0.84**	**0.87**	**0.88**	0.72	0.82	0.86
MIMIC	**0.78**	**0.83**	**0.84**	0.71	0.80	0.80

Table 2. Knowledge distillation accuracy of different dimensional Euclidean and Poincare embeddings.

Emb. dim → **Dataset ↓**	**Poincare**			**Euclidean**		
	2	**4**	**16**	**2**	**4**	**16**
MNIST	**0.23**	**0.19**	**0.20**	0.33	0.24	0.21
AFHQ	**0.46**	**0.21**	**0.17**	0.72	0.31	0.28
STL10	**0.41**	**0.17**	**0.16**	0.60	0.23	0.19
MIMIC	**0.49**	**0.30**	**0.32**	0.55	0.31	**0.32**

Table 3. Average Dice score overlap between the visual semantics of symbols sampled from ζ^0 for different dimensional Euclidean and Poincare embeddings

6. Conclusion

This work provides novel hierarchical explanations for deep discriminative models, demonstrated on several datasets. The proposed framework discretises the continuous latent space of classifiers into discrete features, followed by multiple layers of symbolic abstraction in hyperbolic space to form a knowledge tree which provides hierarchical chain rules as explanations. We demonstrate that hyperbolic geometry allows to embed our knowledge tree with minimal distortion and hence prevent the overcrowding of concepts compared to the Euclidean counterpart. The results show the existence of a consistent and robust set of chain rules for each class, visualised by generating attention regions in an image which are more robust compared to traditional post-hoc methods.

For future work, our framework can be developed into a stand-alone interpretable deep discriminative neuro-symbolic model which improves generalisability. We plan to extend this method with domain experts and assign human-interpretable meaning to symbols.

References

[1] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018. 2

[2] Sharon Lee Armstrong, Lila R Gleitman, and Henry Gleitman. What some concepts might not be. *Cognition*, 13(3):263–308, 1983. 1

[3] Daniel C Burnston and Philipp Haueis. Evolving concepts of "hierarchy" in systems neuroscience. In *Neural Mechanisms*, pages 113–141. Springer, 2021. 1

[4] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019. 2, 3, 5

[5] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 1

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020. 6

[7] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[8] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012. 6

[9] Finale Doshi-Velez, Ryan Budish, and Mason Kortz. The role of explanation in algorithmic trust. Technical report, Technical report, Artificial Intelligence and Interpretability Working Group . . . , 2017. 1

[10] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018. 2, 3

[11] Asma Ghandeharioun, Been Kim, Chun-Liang Li, Brendan Jou, Brian Eoff, and Rosalind W Picard. Dissect: Disentangled simultaneous explanations via concept traversals. *arXiv preprint arXiv:2105.15164*, 2021. 1

[12] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. *arXiv preprint arXiv:1902.03129*, 2019. 1

[13] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019. 1

[14] Koen Helwegen, James Widdicombe, Lukas Geiger, Zechun Liu, Kwang-Ting Cheng, and Roeland Nusselder. Latent weights do not exist: Rethinking binarized neural network optimization. *Advances in neural information processing systems*, 32, 2019. 4, 5

[15] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[16] AEW Johnson, TJ Pollard, SJ Berkowitz, R Mark, and S Horng. Mimic-cxr database (version 2.0. 0). physionet, 2019. 6

[17] Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019. 6

[18] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018. 1

[19] Avinash Kori, Parth Natekar, Balaji Srinivasan, and Ganapathy Krishnamurthi. Interpreting deep neural networks for medical imaging using concept graphs. In *International Workshop on Health Intelligence*, pages 201–216. Springer, 2021. 1

[20] Joshua Alexander Kroll. *Accountable algorithms*. PhD thesis, Princeton University, 2015. 1

[21] Andrew K Lampinen, Nicholas A Roy, Ishita Dasgupta, Stephanie CY Chan, Allison C Tam, James L McClelland, Chen Yan, Adam Santoro, Neil C Rabinowitz, Jane X Wang, et al. Tell me why!–explanations support learning of relational and causal structure. *arXiv preprint arXiv:2112.03753*, 2021. 1

[22] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. 1

[23] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 1, 6, 7, 8

[25] Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*, 2019. 2

[26] David Meunier, Renaud Lambiotte, Alex Fornito, Karen Ersche, and Edward T Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3:37, 2009. 1

[27] Stephen Muggleton, Wang-Zhou Dai, Claude Sammut, Alireza Tamaddoni-Nezhad, Jing Wen, and Zhi-Hua Zhou. Meta-interpretive learning from noisy images. *Machine Learning*, 107(7):1097–1118, 2018. 2

[28] Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[29] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017. 2, 5

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016. 1, 6, 7, 8

[31] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR, 2018. 2

[32] Sam Sattarzadeh, Mahesh Sudhakar, Konstantinos N Plataniotis, Jongseong Jang, Yeonjeong Jeong, and Hyunwoo Kim. Integrated grad-cam: Sensitivity-aware visual explanation of deep convolutional networks via integrated gradient-based scoring. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1775–1779. IEEE, 2021. 1

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 1, 7, 8

[34] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017. 1, 6, 7, 8

[35] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3619–3629, 2021. 1

[36] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019. 2

[37] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016. 2

[38] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4, 5

[39] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Probabilistic neural symbolic models for interpretable visual question answering. In *International Conference on Machine Learning*, pages 6428–6437. PMLR, 2019. 2

[40] Petra Vetter and Albert Newen. Varieties of cognitive penetration in visual perception. *Consciousness and cognition*, 27:62–75, 2014. 1

[41] CM Wessinger, J VanMeter, Biao Tian, J Van Lare, James Pekar, and Josef P Rauschecker. Hierarchical organization of the human auditory cortex revealed by functional magnetic resonance imaging. *Journal of cognitive neuroscience*, 13(1):1–7, 2001. 1

[42] Fan Yang, Zhilin Yang, and William W Cohen. Differentiable learning of logical rules for knowledge base reasoning. *Advances in neural information processing systems*, 30, 2017. 2

[43] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Josh Tenenbaum. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *Advances in neural information processing systems*, 31, 2018. 2

[44] Yuyu Zhang, Xinshi Chen, Yuan Yang, Arun Ramamurthy, Bo Li, Yuan Qi, and Le Song. Efficient probabilistic logic reasoning with graph neural networks. *arXiv preprint arXiv:2001.11850*, 2020. 2